Oracle Communications Digital BSS Reference Solution Implementation Guide

October 2018

Copyright © 2017, 2018, Oracle and/or its affiliates. All rights reserved.

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software, without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons of whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Contents

1. Introduction

The goal of this document is to introduce the concept of solution level extensibility to its users. This document will cover a sample business use case in detail and the extensions required for each application to enable an end to end flow.

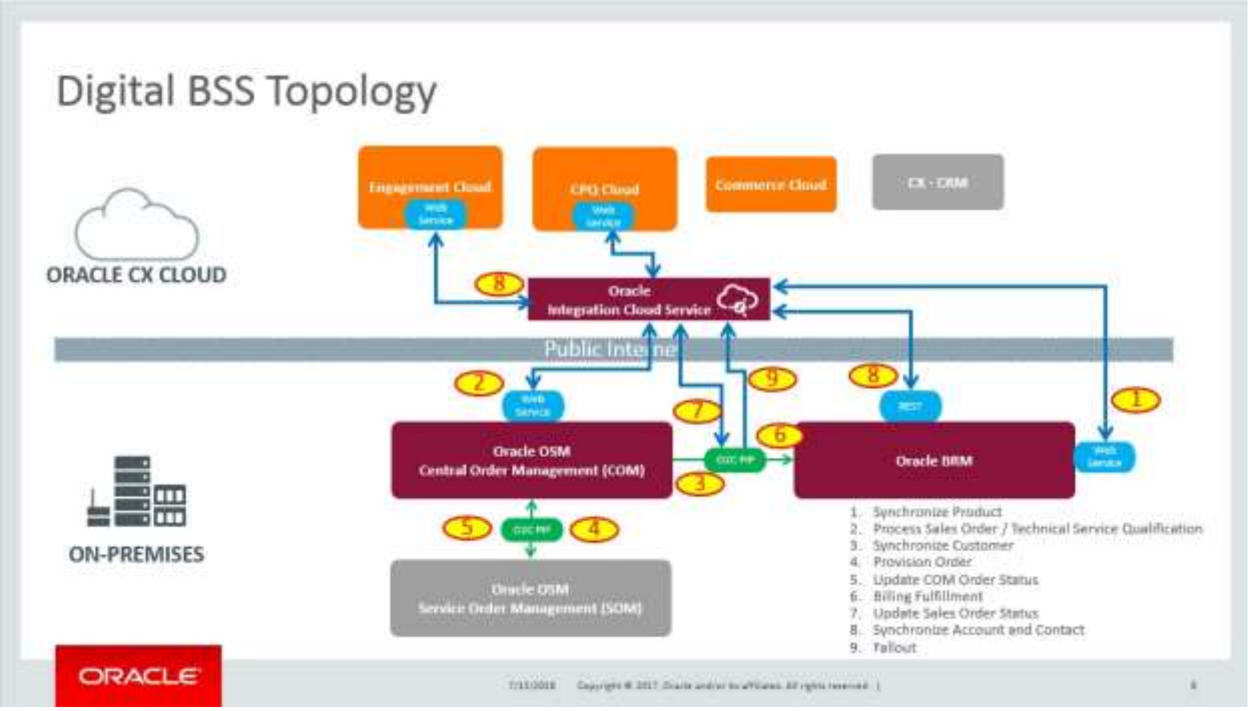
1.1. Executive Summary

The Oracle Communications Digital BSS Reference Solution is a set of Concept-to-Cash reference and sample materials, which enables service providers to rapidly monetize their customer engagements with a modern customer buying, order delivery and personalized usage experience. Key capabilities are: Offer Design, Order Capture, Order Delivery, and Customer & Billing Care.

1.2. What's new in Digital BSS release?

With this release, RODOD has taken the first step towards the cloud enablement by providing a pre-built integration with CPQ and Oracle Engagement Cloud using ICS.

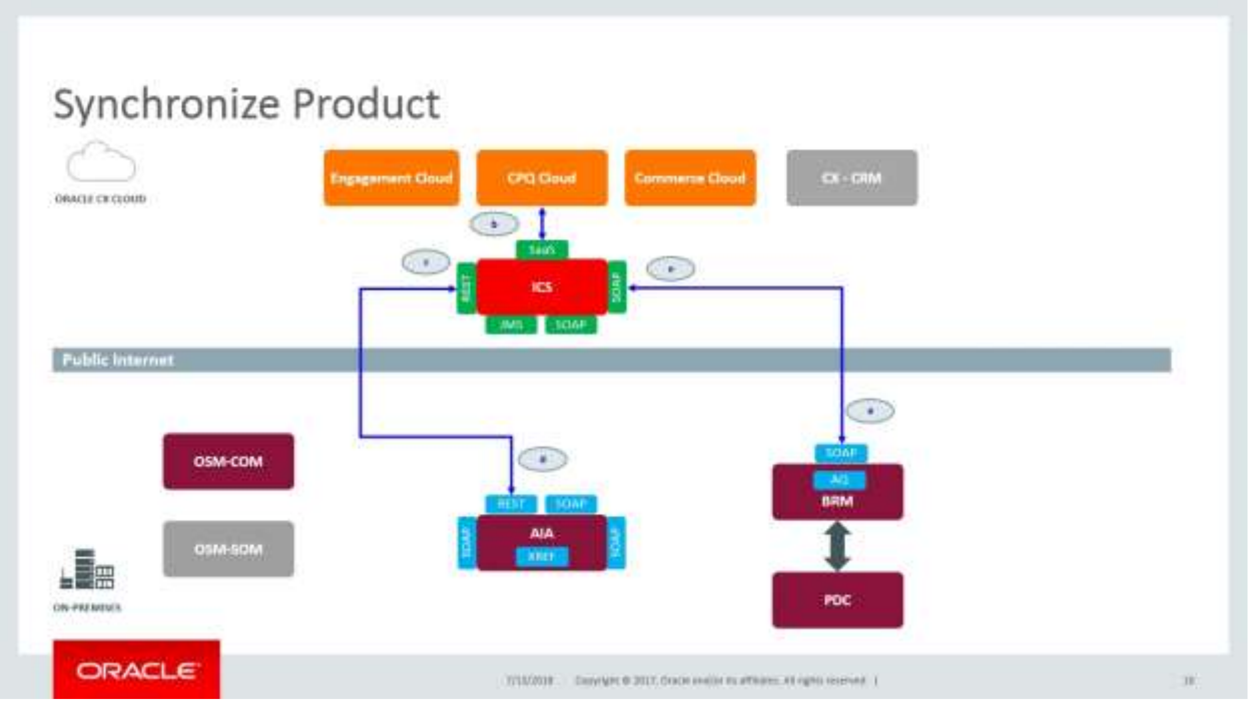
The reference solution enables order to cash business process integration between CPQ, OSM, & BRM/PDC. The integration has capabilities across offer design, order delivery and order fulfillment. The diagram below shows the new topology.



Digital BSS Topology

1. Synchronize Product
2. Process Sales Order / Technical Service Qualification
3. Synchronize Customer
4. Provision Order
5. Update COM Order Status
6. Billing Fulfillment
7. Update Sales Order Status
8. Synchronize Account and Contact
9. Fallout

1.2.1. Synchronize Product



Synchronize Product

Copyright © 2017, Oracle and/or its affiliates. All rights reserved. |

Synchronize product integration flow details:

a) Create charge/discount offers in PDC. Various offers are created in a changeset and when submitted, this data is synced automatically from PDC into BRM, and then made available in BRM AQ.

  - The current AQ adapter publishes the data in the AQ_SYNC queue table in a CLOB containing the ProductInfoChange/DiscountInfoChange XML message.
  - The CLOB is transformed into an XML document using PIN_EVENT_TY_To_ProductInfoChange.xsl/PIN_EVENT_TY_To_DiscountInfoChange.xsl.
  - Once product sync is triggered in PDC, product sync monitor table will be updated with status 1, and updated to 2 once sync is successful. A status of 3 indicate an error, and error message is shown.

b) ICS receives the Product/Discount Sync message and sends it to target CRM application

  - ICS integration, CommsC3_BRMToCXProductSync is triggered when product sync request is triggered from PDC, and CommC3_BRMToCXDiscountSync is triggered when discount sync is triggered from PDC.
  - The target CRM application for the Product/Discount Sync message can be one or many applications - for example CPQ and/or Commerce Cloud. The reference solution is pre-integrated with CPQ. CommsC3_BRMToCPQProductSync is triggered by CommsC3_BRMToCXProductSync and CommC3_BRMToCXDiscountSync to send product/discount sync message to CPQ.
  - Specific enrichment and orchestration process is defined in ICS as a reference solution for SIs to determine a pattern for mapping the Product/Discount Sync message.
  - After Product/Discount synchronization is done, the CRM application sends a response back to ICS.

c) ICS continue orchestration and eventually call AIA XREF
  - This flow starts after receiving success response from CRM application for the Product/Discount Sync message.
  - ICS has orchestration logic which will use a lookup table to determine if AIA is configured for the overall solution.
  - AIA XREF REST adapter is used to update XREF_ITEM_ITEMID, XREF_PRICELINETYPE_ID and XREF_PRICELINE_ID table

d) Sending Product/Discount sync response back to BRM application
  - ICS responds with success or failure to BRM WS
  - BRM WS updates the response status to product sync monitor table

e) When the data are sync over to CPQ via ICS integrations, parts and several tables are updated in CPQ
  a. Parts Model – When a product or discount is synced from PDC/BRM, a corresponding part is either created or updated. To create or update a part is determined by the value of product POID of product/discount stored in _part_custom_field2. Table below lists data mapping between BRM/PDC and CPQ.

Name	**Part Attribute**	**BRM Field**	**Values**	**Description**
Part Number	partNumber	ProductInfoChange:ProductInfo:Name	[string]	Text field for identifying the parts in CPQ, matching the Charge Offer or Discount Offer name in PDC. Update Supported = Yes
Description	description:en	ProductInfoChange:ProductInfo:Description		Update Supported = Yes
Product Master ID	_part_custom_field2	ProductInfoChange:ProductInfo:ProductPoid (last number of the POID is always truncated to 0)	[string]	text field matching the POID in BRM Update Supported = No
Billing Flag	_part_custom_field10	Y	[string]	Set to Y for billable products and services in BRM Set to N for Promotions and Commercial Bundles Update Supported = No

Billing Service Type	_part_custom_field11	ProductInfoChange:ProductInfo:Type, and ICS lookup data table CommsC3_BillingCode	[string]	Mapped from "Applicable to" field in PDC
Billing Type	_part_custom_field12		Subscription [SUBSCRIPTION]	For subscription services, set Billing Type to Subscription [SUBSCRIPTION] For account purchases, set Billing Type to Item [ITEM] For service bundles, set Billing Type to Service [SERVICE BUNDLE] For Discount Offers, set Billing Type to Discount [DISCOUNT] Sponsorship [SPONSORSHIP] is not used Special Rating [SPECIAL RATING] is not used
Pricing Commit Type	_part_custom_field13	DYNAMIC	Committed [COMMITTED], Dynamic [DYNAMIC]	Default to DYNAMIC Update Supported = No
Offer Type	_part_custom_field14	PRODUCT	Promotion [OFFER] Bundle [BUNDLE] Simple [PRODUCT] Promotion Group [PROMOTION GROUP]	For Promotions, set Offer Type to Promotion [OFFER] For Commercial Bundle, set Offer Type to Bundle [BUNDLE] For Product and Service Bundle, set offer Type to Simple [PRODUCT] Promotion Group [PROMOTION

				GROUP] is not used Update Supported = No
Service instance Flag	_part_custom_field15	false	True [true], False [false]	Set to true for Simple Bundle, default to false Update Supported = No

b. OC_PRODUCT data table, insert a row to this table when a product is synced from PDC/BRM. If part (product name) already exists in CPQ, no new row is added

OC_PRODUCT (Column Name)	BRM Field	Description
productName	ProductInfoChange:ProductInfo:Name	Update Supported = No
startDate	ProductInfoChange:ProductInfo:ProductStartTime or current	Update Supported = No
endDate	ProductInfoChange:ProductInfo:ProductEndTime or null	Update Supported = No
createdDate	Current date	Update Supported = No

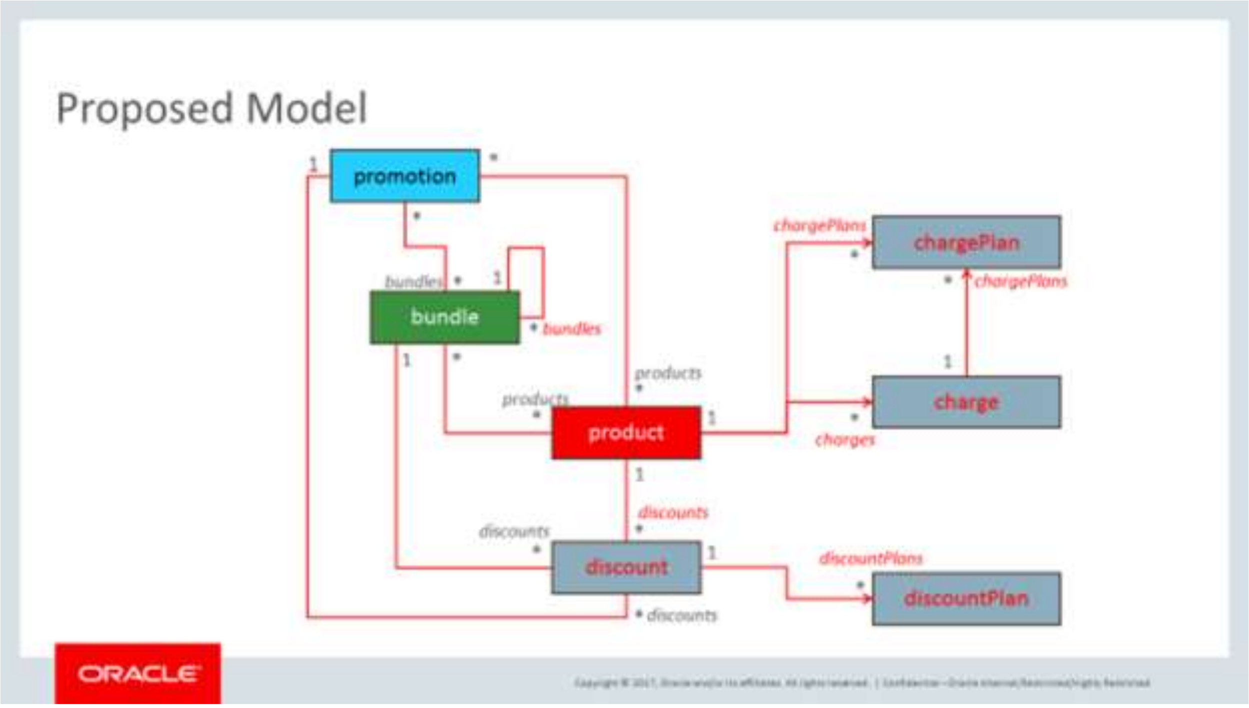
c. OC_PRICELIST data table. A given charge offer may have more than one charge event (MRC/NRC). When a product is synced from PDC/BRM, one or more rows are inserted/updated to this table. Changes to this table takes place when ProductInfoChange:ProductInfo:UsageMap:UsageEventType:RateObj value found a match in ICS lookup table CommsC3_UsageEventType. The insert or update action is dictated by ProductInfoChange:ProductInfo:UsageMap:UsageEventType:RateObj. Table below contains mapping from PDC/BRM fields to OC_PRICELIST table.

OC_PRICELIST (Column Name)	BRM Field	ICS Lookup	Description
rateTierId (Key)	ProductInfoChange:ProductInfo:UsageMap:UsageEventType:RateObj:RateDetails:RateObj (Last number on RateObj is always truncated to 0)		Row Primary Key Update Supported = No
priceListName	ProductInfoChange:ProductInfo:UsageMap:UsageEventType:RatePlanSelector:ValueRangesRatePlanName OR DEFAULT	CommsC3_PriceList	Price List Name Update Supported = Yes

productName	ProductInfoChange:ProductInfo:UsageMap:Name		The name reference to Part Number
priceChargeType	ProductInfoChange:ProductInfo:UsageMap:UsageEventType	CommsC3_UsageEventType	ONE-TIME or RECURRING Update Supported = No
currency	ProductInfoChange:ProductInfo:UsageMap:BRMCurrencyCode	CommsC3_CurrencyCode	Currency code for this Price List Update Supported = No
listPrice	ProductInfoChange:ProductInfo:UsageMap:UsageEventType:RateObj:RateDetails:QuantityTiers:BalImpacts:ScaledAmount		The list price of the given part for this Price List. Update Supported = Yes
costPrice			The cost price of the given part for this Price List (not used)
startDate	ProductInfoChange:ProductInfo:ProductStartTime or current date		Date when the current Price List is started (not used) Update Supported = No
endDate	ProductInfoChange:ProductInfo:ProductEndTime or null		Date when the current Price List is ended (not used)
attribtueId			Not Used
createdDate	current date		Date and time when the record is created from Synchronize Product flow. Update Supported = Yes
updateDate	current date		Update Supported = Yes

createdBy			Not Used.

1.2.2. Catalog Operations (Enhanced Offer Design)



Proposed Model

Catalog operations exposes a REST web-service from ICS integration. The operations can be used to

a. Create all new objects
    a. Create a whole new hierarchy of promotion which includes new promotion, new bundle in the promotion, new bundle within bundle, new products and discounts within promotion and/or bundle.
    b. Create new bundle which includes new bundle, new bundle within bundle, new products and/or discounts within bundle.
    c. Create new set of products.
    d. Create new set of discounts.
b. Create new objects and associate to existing objects
    a. Create a new bundle and associate it to existing promotion.
    b. Create a new bundle and associate it to existing bundle.
    c. Create a new bundle with existing products and/or discounts.
    d. Create a new promotion with existing bundle(s)/product(s)/discount(s).
    e. Create a new product and associate it to existing bundle(s)/promotion(s).
    f. Create a new discount and associate it to existing bundle(s)/promotion(s).
c. Create association among existing objects to existing objects.
    a. Associate existing bundle(s) to existing promotion.
    b. Associate existing bundle(s) to existing bundle.

    c. Associate existing product(s) to existing promotion.
    d. Associate existing discount(s) to existing promotion.
    e. Associate existing product(s) to existing bundle.
    f. Associate existing discount(s) to existing bundle.

Catalog Operations can also be used to

a. Capture attributes of promotion(s) and/or bundle(s) and/or product(s) and/or bundle(s) of multiple data type during creation.
b. Capture new attributes of promotion(s) and/or bundle(s) and/or product(s) and/or bundle(s) of multiple data type during update.
c. Change data type of the attributes during update.

External User can capture a product hierarchy including promotions, bundles, products, discounts etc. in various edge applications using this REST service.

This REST service accepts json input data which includes elements enablePDCChangesetReview, promotions, bundles, products, charges, chargePlans, discountPlans and discounts.

Refer to “IBCMarketOfer.json” in CX Reference Solution Bunder for sample payload.

Information is captured in CPQ and PDC in following way:

Promotion

- Create/update 2 CPQ parts
    1. One with promotionName and description as provided and OfferType as **PROMOTION**.
    2. Another with promotionName-Root
- Create a BOM Item with variable name as promotionName-Root, part as promotionName-Root and root as variable name promotionName-Root. This will act as root BOM item
- Create another BOM item with variable name as promotionName, part as promotionName, parent as variable name of promotionName-Root and root as variable name promotionName-Root. This is 1st level child of Root.
- If child bundles' information is provided.
    1. Capture/update information in OC_BUNDLE data table.
    2. Create/update BOM Item for BUNDLE with part as BUNDLE part name, parent as promotionName variable name and root as promotionName-Root variable name.
    3. Query for bundle hierarchy in OC_BUNDLE and create/update BOM items for BUNDLE hierarchy.
    4. Query for attributes of bundle and hierarchy in OC_PROD_ATTRIBUTES and create/update attributes' information in BOM attributes.
- If child products' information is provided.
    1. Capture/update information in OC_BUNDLE data table.
    2. Create/Update BOM item for PRODUCT with part as PRODUCT part name, parent as promotionName variable name and root as promotion-Root variable name.

- If child discounts' information is provided.
    1. Capture/update information in OC_BUNDLE data table.
    2. Create/Update BOM item for DISCOUNT with part as DISCOUNT part name, parent as promotionName variable name and root as promotion-Root variable name.
- If attributes' information is provided.
    1. Capture/update information is OC_PROD_ATTRIBUTES.
    2. Create/update attribute information in BOM attributes.

Bundle

- Create/update a CPQ part with name and description and offerType as **BUNDLE.**
- If Bundle's type is SERVICE, billingType should be set to **SERVICE BUNDLE.**
- If Bundle's applicableTo is provided, billingServiceType should be set to /service/<applicableTo>
- If child bundles' information is provided.
    1. Capture/update information in OC_BUNDLE data table.
- If child products' information is provided.
    1. Capture/update information in OC_BUNDLE data table.
- If child discounts' information is provided.
    1. Capture/update information in OC_BUNDLE data table.
- If attributes' information is provided.
    1. Capture/update information is OC_PROD_ATTRIBUTES.
- Query BOM items for corresponding BOM item, if present
    1. Synchronize the complete bundle hierarchy as BOM item hierarchy by querying OC_BUNDLE.
    2. Query for attributes of bundle and hierarchy in OC_PROD_ATTRIBUTES and create/update attributes' information in BOM attributes.

Product

- Create/modify charge offer in PDC.
- If attributes' information is provided.
    1. Capture/update information in OC_PROD_ATTRIBUTES.
- Query BOM items for corresponding BOM item, if present create/update Bom Item attributes.

Charge

- Create/modify charge rate plan selector in PDC.

Charge Plan

- Create/modify charge rate plan in PDC.

Discount

- Create/modify discount offer in PDC.
- If attributes' information is provided.
    1. Capture/update information is OC_PROD_ATTRIBUTES.
- Query BOM items for corresponding BOM item, if present create/update Bom Item attributes.

Discount Plan

- Create/modify discount rate plan in PDC.

PDC changeset can be controlled using following methods

- User will be able to control behaviors of PDC changeset from input payload, by providing “true” or “false” in “enablePDCChangesetReview” element. (i.e. whether to just create changeset in PDC to review the elements or let it submit)

- User will be able to control behaviors of PDC changeset from System configuration lookup in ICS.
- Default value of submitting changeset will be provided.
- Priority of these variables in changeset control
  - User Input > System Configuration > Default

Catalog Operations integration flow details:

CommsC3_Catalog

- ***This is entry point to Digital BSS catalog operations***.
- This is used to orchestrate syncing of catalog objects to various edge systems.
- It calls CommsC3_PDCCatalog and CommsC3_CPQCatalog respectively.

CommsC3_PDCCatalog

- This integration is used to create/update PDC catalog for Digital BSS.
- This generates an xml structure which is understood by PDC.
- It invokes COMMS_XREF_REST to query xref details of the objects provided from CHARGEPLAN_ID.
- Based on the results retrieved first it creates a payload for new entities and calls CommsC3_PDCOperations with action CREATE, and then creates a payload for updated entities and calls CommsC3_PDCOperations with action MODIFY.
- Once response is received from CommsC3_PDCOperations it updates the CHARGEPLAN_ID.
- ***This integration can be invoked directly by End User which will interact with configured PDC***.

CommsC3_PDCOperations

- This integration is used to interact with PDC Webservices
- Input to this integration is an xml schema which adheres to **PDCOffers.wsdl** found in CommsC3_PDCOperations Connection. Sample examples can be found by locating referenceDiscountOffers.xml and referenceChargeOffers.xml in CX Reference Solution Bundle.

CommsC3_CPQCatalog

- This is used to orchestrate information to be captured in CPQ.
- It calls CommsC3_CPQBomHierAndAttributes and CommsC3_UpdateCPQProducts.
- For parts and discounts only CommsC3_CPQBomHierAndAttributes is called with type as PRODUCT and DISCOUNT respectively.

- For bundles CommsC3_UpdateCPQProducts is called to create a part and then CommsC3_CPQBomHierAndAttributes is called with type as BUNDLE.
- For promotions CommsC3_UpdateCPQProducts is called to create 2 parts and then CommsC3_CPQBomHierAndAttributes is called with type as PROMOTION.
- Elements are processed in order Products > Discounts > Bundles > Promotions.
- ***This integration can be invoked directly by End User which will interact with configured CPQ to create Catalog Object Hierarchy***.

CommsC3_CPQBomHierAndAttributes

- This integration is used to capture the Catalog Objects hierarchy in OC_BUNDLE and then populating/enriching BOM Item hierarchy also to take care of Catalog Object attributes.
- First it queries and captures the children information in OC_BUNDLE.
- If it has received PROMOTION type object, it creates/updates 2 BOM Items for promotion using CommsC3_CPQUpdateBomHierarchy.
- It captures the catalog object attributes using CommsC3_CPQCatalogAttr.
- Then it queries OC_BUNDLE for child and OC-BOMItemDef to create/update BOM Item hierarchy and attributes.
- It is implemented to process 4 levels of children considering Promotion as Level 0. i.e. it can process Promotion -> Bundle -> Bundle -> Bundle -> Product

CommsC3_CPQCatalogAttributes

- This integration is used to capture attributes in OC_PROD_ATTRIBUTES, hard attributes (eg specification) in OC_PART_ATTRIBUTES and populate/update Bom Item Attributes.
- First queries and captures attributes in OC_PROD_ATTRIBUTES.
- It captures specification information in OC_PART_ATTRIBUTES.
- It queries BOM Item for parts and updates BOM item attributes.

CommsC3_CPQUpdateProducts

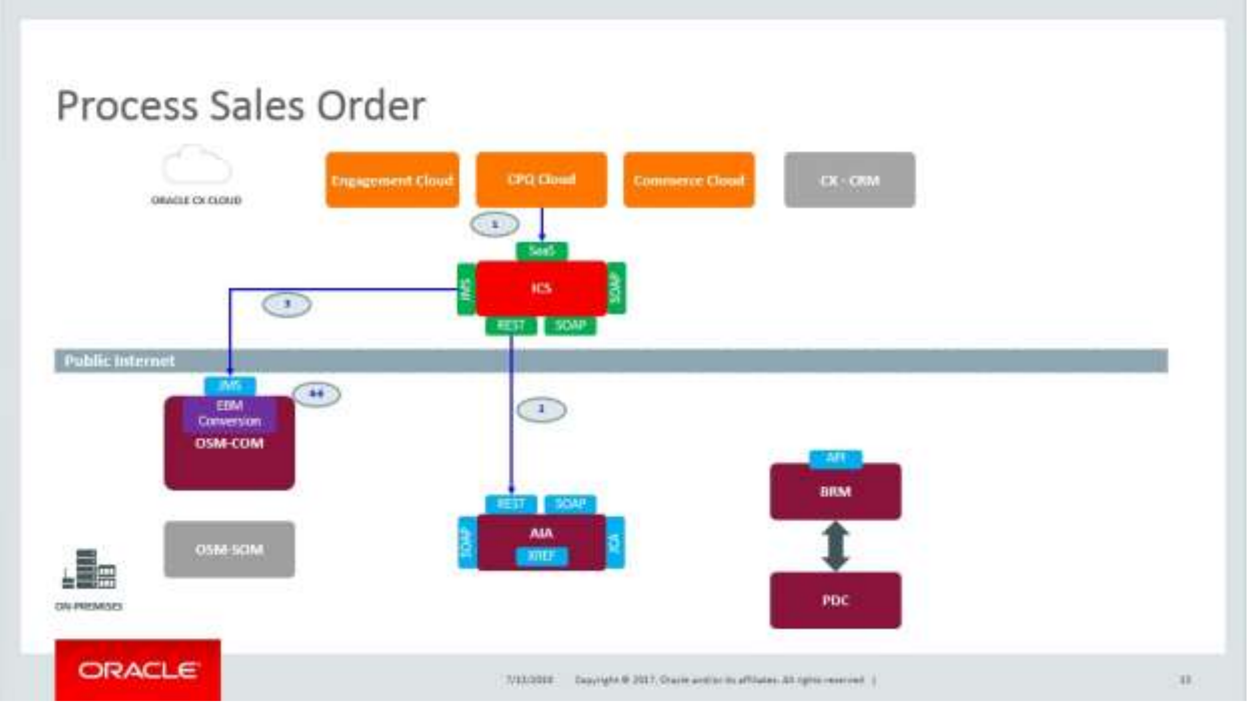
- This integration is used to create/update CPQ parts and corresponding BOM data.
- ***This integration can be invoked directly by End User which will interact with configured CPQ to create part and BOM structure.***

COMMSC3_CPQUPDATEBOM

- This integration is used to update BOM hierarchy, mapping and attributes.

This integration can be invoked directly by End User which will interact with configured CPQ to create BOM hierarchy, capture attributes, item and attributes mapping

1.2.3. Process Sales Order



Process Sales Order

1. The CRM application generates an Order message and sends to the ICS for orchestration.
    a. The reference solution provides a design pattern on how the Order message for a given application (e.g. CPQ) can be transformed to the desired target order message for the OSM.
    b. When user select “Submit” action button in Commerce/CPQ, it triggers ICS integration CommsC3_CPQCreateSalesOrder.
    c. CommsC3_CPQCreateSalesOrder integration is very specific to CPQ and it transforms CPQ transactional XML message to Optimized Sales order message.
    d. The optimized sales order is sent to CommsC3_CreateSalesOrder for further enrichment to match the required order message format in downstream system.

2. The ICS enriches the Order message with data from other CRM applications and send to target application.
    a. CommsC3_CreateSalesOrder receives Optimized Sales Order message.
    b. Order message enrichment requires additional integration in ICS to get data for the customer account hierarchy, product/pricing catalogs and AIA XREF data.

3. Order orchestrator intercepts the Order message from ICS to enrich and transform the message.
    a. The orchestration logic in reference solution transforms the CPQ transaction request XML message into the Optimized sales order XML message.
    b. In this transformation process, it populates and enriches the optimized sales order with data from XREF tables. The data returned from each XREF table uniquely identify a resource within the XREF table.
    c. Table below contains the list of XREF table involved in this transformation process and the corresponding changes made to XREF table by the order orchestrator:

XREF Table Name (XREF FILE Name)	Description
XREF_SALESORDER_ID (SALESORDER_ID.xref)	Add a new record. Insert CPQ transaction identifier in CPQ_01 column and auto-generated value in COMMON column. Value from COMMON column is included in optimized sales order.
XREF_SALESORDER_LINEID (SALESORDER_LINEID.xref)	Add a new record for each CPQ transaction line. The transaction line identifier and document number for the transaction line are concatenated, and inserted in CPQ_01 column. Auto-generated value insert in COMMON column. Value from COMMON column included in optimized sales order.
XREF_ITEM_ITEMID (ITEM_ITEMID.xref)	Lookup the COMMON value by matching the part number from transaction line with value in CPQ_01 column in this XREF table.
XREF_INSTALLEDPRODUCT_ID (INSTALLEDPRODUCT_ID.xref)	Add a new record for the asset in each transaction line. Insert asset identifier in CPQ_01 column, and auto-generated value in COMMON column. Value from COMMON column included in optimized sales order. BRM_01 column is updated by AIA with BRM POID when product is purchased in BRM.
XREF_CUSTOMERPARTY_A_23 (CUSTOMERPARTY_ACCOUNTID.xref)	Add a new record for each unique account identifier. Insert owner, ship to, service, billing account identifier in CPQ_01 column. Auto-generated value in COMMON column. Value from COMMON column included in optimized sales order. BRM_01 column is updated by AIA with BRM POID when account is created in BRM.
XREF_CUSTOMERPARTY_A_31 (CUSTOMERPARTY_ADDRESSID.xref)	Add a new record for each ship to address identifier and service address identifier. Insert address identifier in CPQ_01 column, and auto-generated value in COMMON column. Value from COMMON column included in optimized sales order. BRM_01 column is updated by AIA with BRM POID when address is configured in BRM.
XREF_CUSTOMERPARTY_C_33 (CUSTOMERPARTY_CONTACTID.xref)	Add a new record for each owner contact identifier and ship to contact identifier. Insert contact identifier in CPQ_01 column, and auto-generated value in COMMON column. Value from COMMON column included in optimized sales order BRM_01 column is updated by AIA with BRM POID

	when contact is configured in BRM.
XREF_CUSTOMERPARTY_B_32 (CUSTOMERPARTY_BILLPROFILEID.xref)	Add a new record. Insert billing profile identifier in CPQ_01 column, and auto-generated value in COMMON column. Value from COMMON column included in optimized sales order. BRM_01 column is updated by AIA with BRM POID when billing information is configured in BRM.
XREF_CUSTOMERPARTY_P_45 (CUSTOMERPARTY_PAYPROFILEID.xref)	Add a new record. Insert billing profile identifier in CPQ_01 column, and auto-generated value in COMMON column. Value from COMMON column included in optimized sales order. BRM_01 column is updated by AIA with BRM POID when payment profile is configured in BRM.
XREF_CUSTOMERPARTY_D_39 (CUSTOMERPARTY_DEFAULTBALANCEGROUPID.xref)	There are 2 types of record added to this table: Add a new record for each unique account inserted into CUSTOMERPARTY_ACCOUNTID.xref. Insert COMMON value generated for CUSTOMERPARTY_ACCOUNTID.xref in COMMON column, and account identifier into CPQ_01 column. Add new record for each product inserted into INSTALLEDPRODUCT_ID.xref. Insert COMMON value generated for INSTALLEDPRODUCT_ID.xref in COMMON column and asset identifier in CPQ_01 column. BRM_01 column is updated by AIA with BRM POID when balance group is configured in BRM.

d. Once the translation and enrichment is done the ICS orchestration flow will publish the Optimized Sales Order into the Custom queue defined in the OSM

4. The Soap Forwarder MDB listening on the custom queue consumes the Optimized Sales Order and wraps it in a SOAP envelope before transferring to the incoming OSM queue.
5. An OSM recognition rule transforms the optimized message into an EBM for the O2A cartridge
6. The O2A cartridge extensions deployed in the OSM application will further process the sales order according to the orchestration plan.

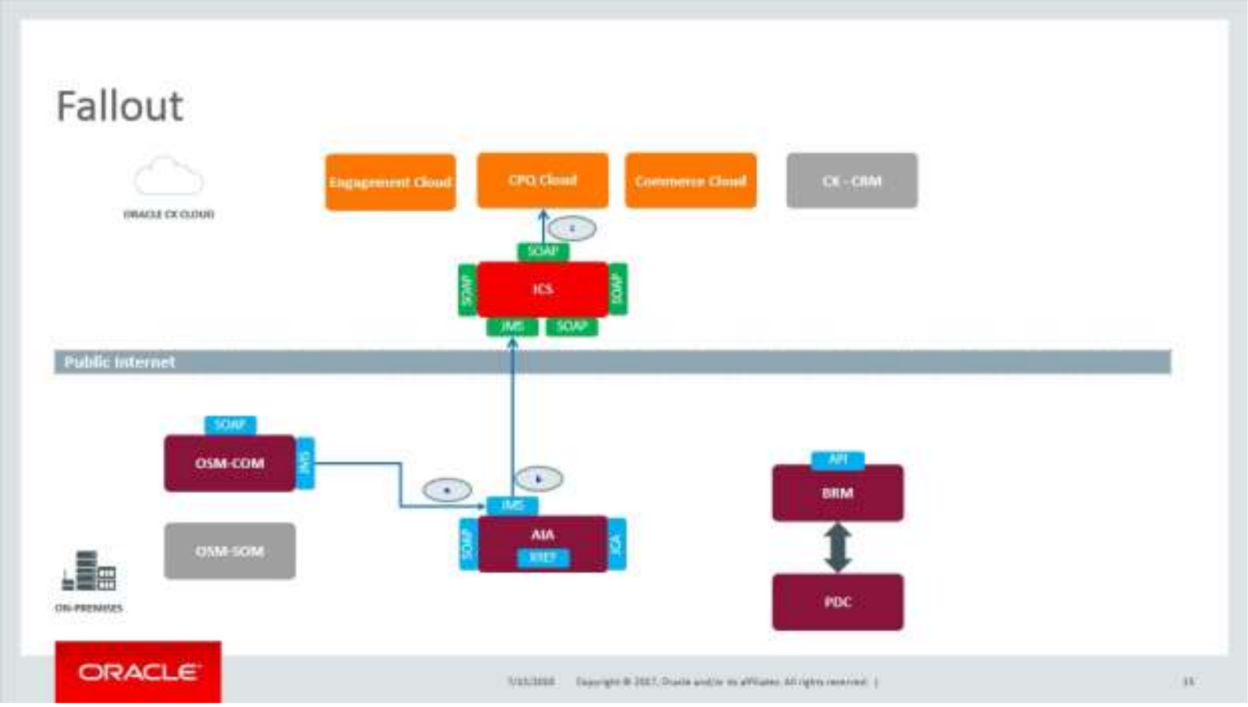
1.2.4. Technical Service Qualification

TSQ (Technical Service Qualification) flow execute the same Process Sales Order flow. Existing ICS integration CommsC3_CreateSalesOrder is updated to support TSQ. When TSQ request is submitted from CRM, process sales order flow is triggered to send TSQ message to OSM. OSM validates the Technical Service Qualification and sends TSQ status back to CRM.

The following DVM changes have been done in AIA for TSQ.

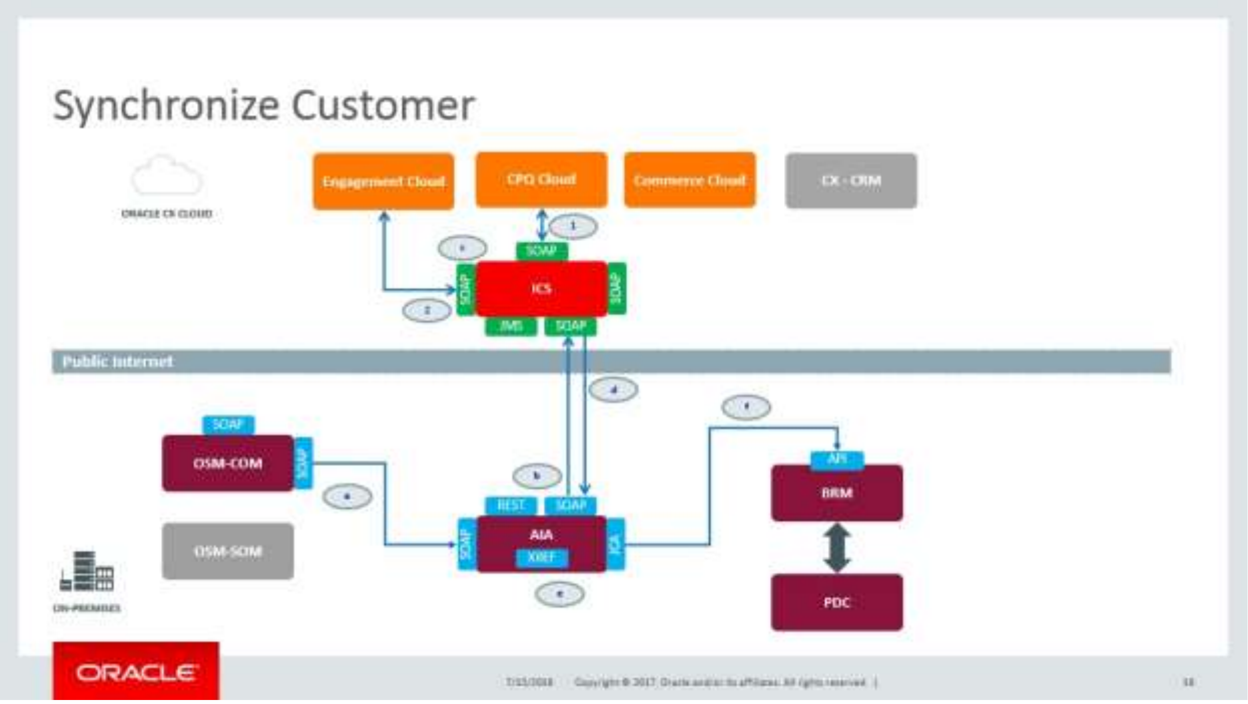
SALESORDER_STATUS.dvm	CPQ_01 column has been added to support Passed TSQ,Failed TSQ , Pending TSQ status.
SALESORDER_LINESTATUS.dvm	CPQ_01 column has been added to support Passed TSQ, Failed TSQ, and Pending TSQ status.

1.2.5. Fallout



If error occurs in the order flow, OSM sends Trouble Ticket message to AIA to raise Trouble Ticket. The ICS integration COMMSC3_AIAGETTTDETAILS consume Trouble Ticket message from AIA and send Fallout message to CRM/CPQ.

1.2.6. Synchronize Customer


Synchronize Customer

CPQ query account details from Oracle Engagement Cloud:

1. Provide an OEC opportunity name in CPQ "Opportunity Name" field and select "Assign Customer Account"
    a. CPQ invokes ICS integration CommsC3_CPQRetrieveAccDetails
2. The integration retrieves the account details from Oracle Engagement Cloud
    a. Gets OEC opportunity using opportunity name
    b. Gets billing account using the target party ID from the opportunity
    c. Gets the owner account, which is the parent of billing account
    d. Gets all billing profiles configured on billing account
    e. Finds all the service accounts, which will all have the billing account as their parent
    f. Copies the owner account, billing account and service accounts hierarchy and details, and billing profiles identifier into ocFullAccountList attribute in JSON format, and return results to CPQ.

Synchronize customer integration flow details:

a. OSM sends synchronize customer request to AIA.
b. AIA consume QueryCustomerPartyListEBM message from message queue, transform EBM message to optimized message structure, and issue query account request to ICS
    - QueryCustomerPartyListICSJMSConsumer consume QueryCustomerPartyListEBM from AIA_QRYACCT_GENERIC_JMSQ message queue and forward EBM message to QueryCustomerPartyListICSProvABCSImpl

- QueryCustomerPartyListICSProvABCSImpl extract customer account ID and/or billing profile ID from EBM message, and compose optimized XML message defined in AccountEntity.xml file
- QueryCustomerPartyListICSProvABCSImpl sends optimized XML message containing account ID and/or billing profile ID to ICS integration CommsC3_AIARetrieveAcctDetails

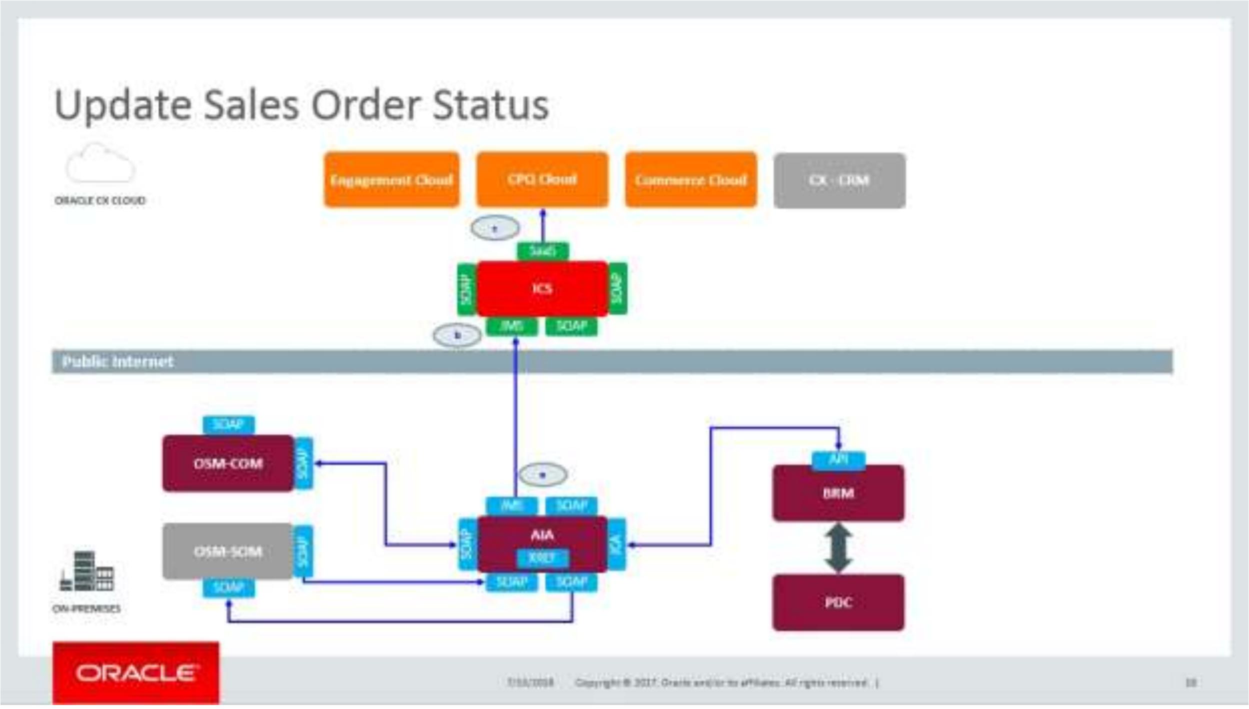
c. ICS project CommsC3_AIARetrieveAcctDetails receives query account message from AIA, and invoke Oracle Engagement Cloud to retrieve account information.
    - For each account ID, the ICS orchestration invoke OEC SOAP API to retrieve following information:
        1. Account contact
        2. Account address
        3. Account configuration detail
    - If billing profile ID is provided, ICS orchestration invoke another ICS project CommsC3_OSCRetrieveBillingProfile to retrieve billing profile details.
    - The CommsC3_OSCRetrieveBillingProfile orchestration invoke OEC SOAP API to retrieve the following information:
        1. Billing profile contact
        2. Billing profile address
        3. Billing profile details (e.g bill currency, billing type)

d. ICS sends queried customer account and billing profile information to AIA.
    - The ICS integration maps customer account information retrieved from Oracle Engagement Cloud to optimized message defined in CustomerAccount.wsdl
    - If there is a billing profile ID, the ICS integration maps billing profile information retrieved from Oracle Engagement Cloud to optimized message defined in CustomerAccount.wsdl
    - Invoke AIA QueryCustomerPartyListICSProvABCSImpl with customer data queried from Oracle Engagement Cloud.

e. QueryCustomerPartyListICSProvABCSImpl receives query accounts response from ICS
    - Transform optimized response message received from CommsC3_AIARetrieveAcctDetails to QueryCustomerPartyListResponseEBM message structure
    - Invoke CommsProcessBillingAccountListEBF composite to continue synchronize customer process flow

f. The synchronize customer process flow continues execution, and create accounts in BRM

1.2.7. Update Sales Order Status



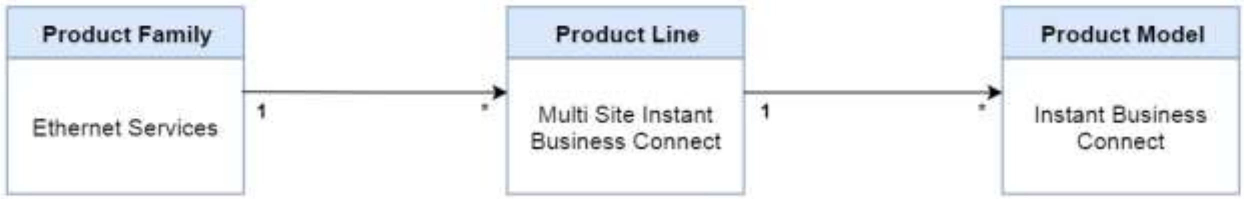
Update Sales Order Status

1. OSM-COM uses existing AIA O2CPIP to request downstream systems such as OSM-SOM and BRM to process the order. For instance:
    a. OSM-COM sends an Initiate Billing request to BRM
    b. OSM-COM sends a Provision Order request to OSM-SOM
    c. OSM-COM sends a Fulfill Billing Message to BRM
2. All these existing integration flows utilizes AIA JMS queues to produce and consume EBM that are already implemented as a complete integrated solution for existing RODOD deployments. The integration architecture remains intact and reusable, and any existing dependencies to XREF data is also satisfied. There is no need to make any changes to the integrations already deployed in customer production environments, including no impacts to change any field extensions.
3. AIA receives the message and sends to the intended "Target" application as specified in the EBM header. However, if the intended target recipient of the messages is not configured or registered with AIA, then the messages would be persisted in a generic JMS queue for further processing. For instance, when an UpdateSalesOrder message is sent to AIA, it does not know the intended target application CPQ_01. Therefore, the **UpdateSalesOrderEBM** messages are persisted into the **AIA_UPDSO_OUT_GENERIC_JMSQ** queue for future processing.
4. The **UpdateSalesOrderEBM** messages that are persisted in the generic **AIA_UPDSO_OUT_GENERIC_JMSQ** queue can then be consumed by new specialized SOA composites which are designed for status updates back to the CPQ system using ICS.
    a. **UpdateSalesOrderICSBatchCommsJMSConsumer** - this composite is responsible for consuming individual messages from the AIA_UPDSO_OUT_GENERIC_JMSQ so that the EBM message can be re-formatted to a simplified message format (ABM) that is intended for ICS. In addition this composite also consumes one or more UpdateSalesOrderEBM, and batch them into temporary files containing multiple

    messages on the local filesystem. Local filesystem is used here merely as an example reference implementation. Additional SOA Suite capabilities can be utilized to implement better approaches for storing transient batch files for further processing.
   b. **UpdateSalesOrderICSBatchCommsFileProducer** - this composite is responsible for producing batched messages to send to ICS using a service provider. The content of the message files containing batched messages are read by the producer, validated and transformed before forwarding on to the service provider for sending to ICS.
   c. **UpdateSalesOrderICSBatchCommsProvABCSImpl** - this composite is responsible for forwarding the transformed UpdateSalesOrderEBM messages as ABMs to the provider ABCS interface in ICS. The implementation of this ABCS is done by calling an ICS interface exposed via a SOAP adapter. Specific URL and user credentials are configured in the Provider ABCS to successfully invoke an intended target integration in ICS (**CommsC3_AIAUpdateSalesOrderBatch**).
5. As messages are successfully consumed by the JMS Consumer off the generic JMS queue, these messages would be batched according to the desired configuration. Then as the batched message files are made available and visible to the File Producer, these message files would be forwarded to the Provider ABCS for forwarding on to ICS.
6. In ICS, the batched message is received, logged for tracking and then transformed further if necessary before calling the reusable Reference Integration CommsC3_CPQUpdateSalesOrder.
7. The Reference Integration **CommsC3_CPQUpdateSalesOrder** is responsible for looping and processing each transformed UpdateSalesOrder messages to map to the appropriate CPQ status update messages.
   a. This integration has additional validation logic to ensure that status update messages that are received out of sequence would not unintentionally update status that are already processed in earlier messages.
   b. This integration also has cross-validation to ensure that status updates at the CPQ transaction and transaction line level would remain coherent.
   c. This integration also has added responsibility to fail the transaction if such errors are received from on-prem stack, or to complete the transaction and create/update CPQ assets when the order is completed successfully downstream.
   d. This integration is using CPQ REST APIs to update the CPQ transaction and transaction lines. This is because the CPQ REST APIs allow out of sequence transaction line updates, whereas the SOAP interfaces do not allow that

1.2.8. CPQ

Product modeling in CPQ is implemented using three key features - Parts Model, Data Table Model and BOM Model. These models and designs are combined to provide a comprehensive set of capabilities to support individual configurable product and the flexibility to reuse the same product in one or more context in a BOM application. The Parts Model is the building block for creating a hierarchical structure of BOM Model Items. In addition to these CPQ product modeling capabilities, CPQ also supports product segmentation by defining Product Family, Product Lines and Product Models.



Product Family

This is the highest level of product segmentation in CPQ. In the reference solution, a new Ethernet Services Product Family is added to delineate this from other possible product families such as Mobile Services or Media Services. This Product Family is the root parent for one or more Product Lines which falls into this category of services. Note that configurable products at this segmentation level and below are not sharable or visible across different Product Family. Typically a few common configurable attributes that cut across all Product Families are placed at the "All Products" level in CPQ.

Product Line

A new Multi Site Instant Business Connect Product Line is added as a child of the Ethernet Services Product Family. Other child Product Lines can be added as extensions by using the features already provided by CPQ. Again custom configurable attributes at this level are only visible to the child Product Models level and not across to other Product Family and Product Lines.

Product Model

The primary Product Line implemented for the RI is the Instant Business Connect (IBC). This is the level where many of the configurable attributes are defined so that the user can select and provide the appropriate values to drive CPQ Configure process in selecting the correct Parts and BOM hierarchy. This is also the level where several Configuration rules such as Recommendation, Constraints and BOM Mapping rules are implemented to support the overall business requirements to generate the correct set of configurable products before the system generates a quote or transaction in the CPQ Commerce process.

Parts Model

CPQ Parts are created

- Manually to support product bundling, or
- Synchronizing Product via PDC and ICS into CPQ directly. See Synchronize Product chapter for more details. OR
- By Catalog Operations. See Catalog Operations chapter for more details.

Oracle Comms Data Table Model

The following CPQ data tables are implemented to support Digital BSS User scenarios.

- OC_PRICELIST

Table below list of columns supported for this table

ColumnName	Description
rateTierId (key)	Rate Tier Id
priceListName	Name of price list

productName	Product name
priceChargeType	Price charge type
Currency	Currency
listPrice	List price
costPrice	Cost price
startDate	Effective start date
endDate	Effective end date
attributeId	Atttribute Id key.
createdDate	Created date.
updatedDate	Updated date.
createdBy	User that created the record.

Table below lists sample data present in the OC_PRICELIST table once product sync has happened. Showing only relevant columns

rateTierId	priceList Name	productName	priceChargeType	Currency	listPrice
0.0.0.1 /rate 193898 0	DEFAULT	EBC 1GB Bandwidth	RECURRING	USD	750
0.0.0.1 /rate 190118 0	Business Pricelist	EBC 1GB Bandwidth	RECURRING	USD	700
0.0.0.1 /rate 228493 0	DEFAULT	Basic UNI Access	ONE-TIME	USD	400

- OC_PRODUCT

Table below lists the columns supported for this table

ColumnName	Description
productName	Product name
startDate	Effective start date
endDate	Effective end date
createdDate	Created date.
updatedDate	Updated date.
createdBy	User that created the record.

Table below lists sample data present in the OC_PRODUCT table once product sync has happened. Showing only the relevant columns.

productName
EBC 1GB Bandwidth
Basic UNI Access

- **OC_BUNDLE**

Table below lists the columns present in the OC_BUNDLE table.

Column Name	Description
productName	Product name
parent	Parent product name

sequence	Indicates the sequence in the model where the product should appear
common	Indicates if the product is common across the model. Typically used for Network model where a product is common across all sites.
createdDate	Created date.
updatedDate	Updated date.
createdBy	User that created the record.

Table below lists sample data present in the OC_BUNDLE table once product sync has happened. Showing only the relevant columns.

productName	**parent**
`Instant Business Connect`	`IBC Market Offer`
`IBC Endpoint (BO)`	`Instant Business Connect`
`EBC 1GB Bandwidth`	`IBC Endpoint (BO)`

- OC_PROD_ATTRIBUTES

Table below lists the columns present in the OC_PROD_ATTRIBUTES table.

Column Name	Description
VariableName (key)	Key column of the table
ProductName	Name of the Product
AttributeName	Name of the Attribute
DataType	Data type of the attribute

Table below lists sample data present in the OC_PROD_ATTRIBUTES table once product sync has happened. Showing only the relevant columns.

VariableName	**ProductName**	**AttributeName**	**DataType**
`IBC Endpoint (BO)-bandwidth-12343213`	`IBC Endpoint (BO)`	`bandwidth`	`String`
`IBC Endpoint (BO)-site-2384878238`	`IBC Endpoint (BO)`	`site`	`String`

- OC_PART_ATTRIBUTES

Table below lists the columns present in the OC_PART_ATTRIBUTES table.

Column Name	**Description**
PartName	Name of the Part
ProductSpecification	Product Specification of the part

Table below lists sample data present in the OC_PART_ATTRIBUTES table once product sync has happened. Showing only the relevant columns.

PartName	**ProductSpecification**
`EBC 1GB Bandwidth`	`CE CoS Bandwidth PS`
`EBC 2GB Bandwdith`	`CE CoS Bandwidth PS`

See Synchronize Product and Catalog Operations chapter for more details.

BOM Data Tables

The following BOM data tables are defined in CPQ to create the BOM hierarchy. Additional columns (*) are added to extend the solution for BOM hierarchy.

1. **Oracle_BomItemDef**: defines the individual BOM items which are used to construct the BOM item hierarchy. This table contains part number references and parent-child relationships for BOM items. It uses Part Number value for BOM Item row key, ID and name. Table below list all columns supported in the table

Column Name	Description
VariableName (Key)	Row Primary Key
SequenceNum	Default sequence Number
ItemId	Unique Item Id
Name	Item Name
ItemType	Not Used – default to Standard Item
PartNumber	Item mapping reference to Part Number
DefaultQuantity	Not Used – default to 1
Optional	Not Used – default to N
SalesItem	Not Used – default to Y
ParentVariableName	Parent BOM Item Row Key. Null for root BOM Item.
RootVariableName	Root BOM Item Row Key
EffectiveFrom	Not Used
EffectiveTo	Not Used
ManufacturingItem	Not Used – default to N

Table below list some sample data in Oracle_BomItemDef table. (Only columns used in reference solution is listed)

VariableName	**Sequence Num**	**ItemId**	**Name**	**PartNumber**	**ParentVariable Name**	**RootVariable Name**
Instant Business Connect-Root	0	Instant Business Connect-Root	Instant Business Connect-Root	Instant Business Connect-Root		Instant Business Connect-Root
IBC Market Offer	1	IBC Market Offer	IBC Market Offer	IBC Market Offer	Instant Business Connect-Root	Instant Business Connect-Root
Instant Business Connect	13	Instant Business Connect	Instant Business Connect	Instant Business Connect	IBC Market Offer	Instant Business Connect-Root
IBC Endpoint (BO)	131	IBC Endpoint (BO)	IBC Endpoint (BO)	IBC Endpoint (BO)	Instant Business Connect	Instant Business Connect-Root

EBC 100MB Bandwidth	1311	EBC 100MB Bandwidth	EBC 100MB Bandwidth	EBC 100MB Bandwidth	IBC Endpoint (BO)	Instant Business Connect-Root

2. **Oracle_BomItemMap**: defines the mapping of CPQ Config attributes to BOM Item. It uses Part Number as row primary and foreign key. Table below list all columns supported in the table

Column Name	Description
VariableName (Key)	Row Primary Key
BomItemVarName	BOM Item foreign key to Oracle_BomItemDef
ConfigAttrVarName	Config Attribute Variable Name
ConfigAttrVarValue	Config Attribute Variable Value to match before adding BOM item to hierarchy
ParentBomMapVarName	Unique path notation totable-based IBC BOM Mapping Rule
EffectiveFrom	Not Used
EffectiveTo	Not Used
Comments (*)	Simple comments

Table below list some example data in Oracle_BomItemMap. (Only columns used in reference solution is listed)

VariableName	BomItemValName	ConfAttrVarName	ConfigAttrVarValue	ParentBomMapVarName	Comments
Instant Business Connect-Root	Instant Business Connect-Root	ocProductModel	IBC	ocEthernetServices:ocMultiSiteInstantBusinessConnect:ocInstantBusinessConnect:ocIBCBOMMappingRule	N
IBC Market Offer	IBC Market Offer	ocProductModel	IBC	ocEthernetServices:ocMultiSiteInstantBusinessConnect:ocInstantBusinessConnect:ocIBCBOMMappingRule	N
Instant Business Connect	Instant Business Connect	ocProductModel	IBC	ocEthernetServices:ocMultiSiteInstantBusinessConnect:ocInstantBusinessConnect:ocIBCBOMMappingRule	N
IBC Endpoint (BO)	IBC Endpoint (BO)	ocInternalEndpoint	EP	ocEthernetServices:ocMultiSiteInstantBusinessConnect:ocInstantBusinessConnect:ocIBCBOMMappingRule	N

EBC 100MB Bandwidth	EBC 100MB Bandwidth	ocBandwidth	100 Mbps	ocEthernetServices:ocMultiSiteInstantBusinessConnect:ocInstantBusinessConnect:ocIBCBOMMappingRule	N

3. **Oracle_BomAttrDef**: defines the individual BOM item attributes to create for a given BOM Item. It uses Part Number as row primary and foreign key. This table provides extension to add BOM Item Attribute to Commerce Line Attribute and Extended Attribute. Table below list all columns supported in the table

Column Name	Description
VariableName (Key)	Row Primary Key
Name	BOM Attribute Name
Values	Values for BOM Item Attribute. Not Used – default to null
BomItemVarName	BOM Item Foreign Key to Oracle_BomItemDef
DataType	Default to String
RootBomItemVarName	Root BOM Item Foreign Key to Oracle_BomItemDef
EffectiveFrom	Not Used
EffectiveTo	Not Used
DisplayValues	Display Values for BOM Item Attribute. Not Used – default to null.
ExtendedAttribute (*)	Extended Attribute name to create and map the BOM Item Attribute value to. Optional.
LineAttributeName (*)	Commerce Line attribute name to create and map the BOM Item Attribute value to. Optional.

Table below list some example data in Oracle_BomAttrDef. (Only columns used in reference solution is listed)

VariableName	Name	BomItemVarName	BomItemMapVarName	DataType	RootBomItemVarName	Extended Attribute	LineAttributeName
EBC 100MB Bandwidth	bandwidth	EBC 100MB Bandwidth	EBC 100MB Bandwidth	String	Instant Business Connect-Root	true	ocBandwidth
IBC	site	ocSites	IBC Endpoint	String	Instant	false	ocSiteName

Endpoint (BO) – site			(BO) – site		Business Connect-Root		

4. **Oracle_BomAttrMap** defines the mapping of CPQ Config attributes to BOM Item attributes. It uses Part Number as row primary and foreign key. Table below list all columns supported in the table

Column Name	Description
VariableName (Key)	Row Primary Key
TargetType	Target type for BOM attribute. Not Used – default to BOM_ATTRIBUTE
TargetVariableName	BOM Attribute Foreign Key to Oracle_BomAttrDef name
SourceType	Source value for BOM attribute. Not Used – default to CONFIG_ATTRIBUTE
StaticEntry	Config Attribute value to match. Not Used – default to null
ConfigAttrVarName	Item mapping reference to Part Number
ConfigAttrVarValue	Set by BOM Advance Mapping Rules. Not Used – default to null
BomItemMapVarName	BOM Attribute Foreign Key to Oracle_BomItemMap
RootBomMapVarName	Unique path notation to table-based IBC BOM Mapping Rule
EffectiveFrom	Not Used
EffectiveTo	Not Used

Table below list some example data in Oracle_BomAttrDef. (Only columns used in reference solution is listed)

VariableName	**TargetVariableName**	**ConfAttrVarName**	**BomItemMapVarName**	**ParentBomMapVarName**
EBC 100MB Bandwidth	bandwidth	ocBandwidth	EBC 100MB Bandwidth	ocEthernetServices:ocMultiSiteInstantBusinessConnect:ocInstantBusinessConnect:ocIBCBOMMappingRule
IBC Endpoint (BO) – site	site	ocSites	IBC Endpoint	ocEthernetServices:ocMultiSiteInstantBusinessConnect:ocInstantBusinessConnect:ocIBCBOMMapp

			(BO) – site	ingRule

5. **Oracle_aboPart2Model**, table for setting primary Root BOM Item to use for managing assets in CPQ to support ABO functionality.

Oracle_aboPart2Model (Column Name)	Description
Part Number	Root Part Number for creating asset in CPQ
Segment	Product Family where the root part belongs
ProductLine	Product Line where the root part belongs.
Model	Product Model where the root part belongs.

Table below list example data in Oracle_aboPart2Model

PartNumber	Segment	ProductLine	Model
IBC Market Offer-Root	ocMultiSiteInstantBusinessConnect	ocEthernetServices	ocInstantBusinessConnect

BOM Mapping Rule

The following are the BOM mapping rules implemented to drive the creation of BOM hierarchy using the data captured in the BOM Model data tables. They are executed in sequence. Note that the execution order is important.

- Config to BOM - the execution sequence of the following rules are meant to iteratively and successively build up the BOM hierarchy using the Config Attributes to represent the Multi-Site Ethernet Services
- BOM to Config - only a few of these rules (Attributes and Custom) have logic to reverse the process of converting the BOM attributes back to Config Attributes. This is executed only when the user wants to modify an asset and therefore need to see the correct Config Attribute values again.

	Rule Name	Variable Name	Description
1	IBC BOM Mapping Rule	ocIBCBOMMappingRule	This is a simple table-based BOM mapping rule. It is used for building the BOM hierarchy partially only for products and services that are not site details related. It uses the BOM Item data definitions provided in Oracle_BomItemDef and Oracle_BomItemMap to create the initial BOM hierarchy which includes the following items:

			Instant Business Connect – Root IBC Market Offer IBC SLA IBC SLA 10% Discount Instant Business Connect IBC Endpoint (BO) [2-n] The config attribute ocProductModel has only one default static value "IBC". This config attribute value would trigger this rule to create the top-level BOM Items excluding IBC Endpoint (BO). The config attribute ocInternalEndpoint has only one default static value "EP", but this attribute is additionally configured as an attribute within the ocSitesArray array set. Therefore this config attribute value in the array would trigger this rule to create one IBC Endpoint (BO) item per site to use as the anchor BOM Item for the remaining BOM Advanced Mapping Rules to construct the complex hierarch
2	IBC BOM Advanced Mapping Rule - Hierarchy	ocIBCBOMAdvancedMappingRuleHierarchy	This is an advanced BML-based BOM Mapping rule. It is used for building the remaining BOM hierarchy using the aggregated config attribute values for a given site. For example, Site 1 is for Washington DC, 50 Mbps and Site 2 is for San Francisco 100 Mbps. These config attributes are defined within an array set and therefore can only be used for building the BOM hierarchy using BML-based advanced mapping rules.
3	IBC BOM Advanced Mapping Rule – Attributes	ocIBCBOMAdvancedMappingRuleAttributes	This is an advanced BML-based BOM Mapping rule. It is used for setting the BOM Item Attributes, Commerce Line Attributes and Extended Attributes using the BOM Attribute data definitions provided in Oracle_BomAttrDef and Oracle_BomAttrMap.
4	IBC BOM Advanced Mapping Rule - Custom	ocIBCBOMAdvancedMappingRuleCustom	This is an advanced BML-based BOM Mapping rule. It is used for additional custom logic for building the Ethernet Services BOM hierarchy that may not be applicable to other service domains. For example: • setting the MRC/NRC on the BOM items based on data defined in the OC_PRICELIST data table • tracking which sites were fulfilled earlier with a specific bandwidth, and allow the user to change the bandwidth for the completed site (Modify Site) • tracking which sites were fulfilled earlier to support Add/Delete sites to support changing the site location (Add Site+ Delete Site)

			- tracking which sites were fulfilled earlier so that these sites cannot be simply removed from the config array, but can be disconnected (Delete Site) - additional custom logic to support correct ABO asset action - Add/Delete/Modify
5	IBC BOM Advanced Mapping Rule – Sequence	ocIBCBOMAdvancedMappingRuleSequence	This is an advanced BML-based BOM Mapping rule. It is used for assigning the correct sequence number for each of the BOM items. It is executed last after the entire BOM item hierarchy is setup accordingly.

Configure Rule

The following configure rules are implemented to control the configuration business logic and constraints using CPQ functionality.

Recommendation Rules

Rule Name	Description
Set Customer ID (ocSetCustomerID)	Rule to copy the _customer_id from the Transaction in commerce-side to configure-side of CPQ for view only.
Force Set Attributes (ocForceSetAttributes)	Rule to force-set the following config attributes so that the value is static and read-only in the view. They are not editable by the user. - Effective Number of Sites (ocEffectiveNumberOfSites) - Fulfilled (ocFulfilled) - Asset Key (ocAssetKey) - Original Bandwidth (ocOriginalBandwidth) - Previous Bandwidth (ocPreviousBandwidth)
Set UI Controller (ocSetUIController)	Rule to generate the JavaScript content to control UI behavior such as disabling or enabling the disconnect site checkboxes and delete buttons.
Disable Site Change (ocDisableSiteChange)	Rule to disable the change of site drop down value for fulfilled sites. User can only change the site name value for new sites. For fulfilled sites, the user has to disconnect the fulfilled site and add a new site.
Disable Bandwidth Change (ocDisableBandwidthChange)	Rule to disable the change of bandwidth drop down value for to-be disconnected sites. bandwidth can only be changed for new or active (fulfilled) sites.
Setup Configuration Details (ocSetupConfigurationDetails)	Rule to enable the side Configuration Details panel which allows array set config attributes to be hidden in main table and re-displayed in the Configuration Details panel. The panel is refreshed as user selects a

	different row in the main array table. Also, provide a Save button to support save action of user changes in the Configuration Details panel.
Set Diagnostic Info (ocSetDiagnosticInfo)	Rule to copy BOM and ABO related diagnostic log info to the config view for debugging and troubleshooting purposes-only.

Constraint Rules

Rule Name	Description
Validate Effective Number of Sites (ocValidateEffectiveNumberOfSites)	Rule to enforce the total count of number of sites (effective number) factoring in the number of sites that are new, already fulfilled and to be deleted. The effective number of sites must be greater than or equal to two. Total Effective Number of Sites must be at least 2. ((Fulfilled+New) - Disconnected) >= 2

2. Extensibility with Digital BSS Reference Solution

This present reference solution focus on the extension on CPQ and ICS.

As part of this release, we present 3 extension scenarios to guide the user on what is needed at each application to enable extensions at a solution level, focusing on CPQ and ICS only.

2.1. Adding a new Price List

The Digital BSS reference solution support the following price list: DEFAULT, Business PriceList.

The price list, known as USAGE_TYPE event in charge selector is configured in PDC. In synchronize product flow, the charge offers and its corresponding price list configured in PDC are synced into CPQ price list data tables. In process sales order flow, ICS integration use the lookup table (CommsC3_PriceList) to translate CPQ price list name to a common name, and subsequently, from the common name to the price list name configured in PDC.

Follow these steps to add a new price list in reference solution:

- Add new price list in CPQ dropdown menu
- Add new price list in AIA data value map file, PRICELIST.dvm.
- Add new price list to ICS lookup table CommsC3_PriceList
- Create new charge selector in PDC
- Verify price list synced in CPQ

2.1.1. CPQ

The price list is stored in a dropdown menu in CPQ. Take the following steps to configure price list in CPQ:

- In CPQ Admin page, select “Catalog Definition” quick link
- In pop-up window, expand “List of Models” under Multi Site Instant Business Connect
- Expand “List of Attributes”, and select “Price List”
- Configure desired price list and deploy changes

2.1.2. AIA

The price list domain value map file (PRICELIST.dvm) is stored in AIA metadata repository. This DVM file contains price list name for all participating edge applications (BRM, CPQ, PDC).

To support new price list, modify PRICELIST.dvm file and add a new row to provide the following

- Provide new price list name configured in PDC in PDC_01 and BRM_01 column
- Provide new price list name configured in CPQ dropdown menu to CPQ_01 column
- Provide a meaningful name for COMMON column.

Save changes and upload PRICELIST.dvm file to MDS. For instruction on DVM deployment, refer to Fusion Middleware Documentation 12c for more details.

Below is sample PRICELIST.dvm included in Digital BSS reference solution.

PRICELIST.dvm:

COMMON	CPQ_01	BRM_01	PDC_01
DEFAULT	DEFAULT	*	*
BusinessPL	Business PriceList	BusinessPL	BusinessPL
ConsumerPL	Consumer PriceList	ConsumerPL	ConsumerPL

2.1.3. ICS

The ICS lookup table (CommsC3_PriceList) contains value map for COMMON, CPQ_01 and PDC_01. To support new price list, modify CommsC3_PriceList in ICS server:

- Add a new row in lookup table
- Copy new price list value from PRICELIST.dvm COMMON, CPQ_01 and PDC_01 column and insert into CommsC3_PriceList COMMON, CPQ_01 and PDC_01 column respectively. PRICELIST.dvm is stored in MDS /apps/AIAMetaData/dvm/.

Below is sample CommsC3_PriceList lookup values

CommsC3_PriceList:

COMMON	CPQ_01	PDC_01
DEFAULT	DEFAULT	*
ConsumerPL	Consumer PriceList	ConsumerPL
BusinessPL	Business PriceList	BusinessPL

2.1.4. PDC

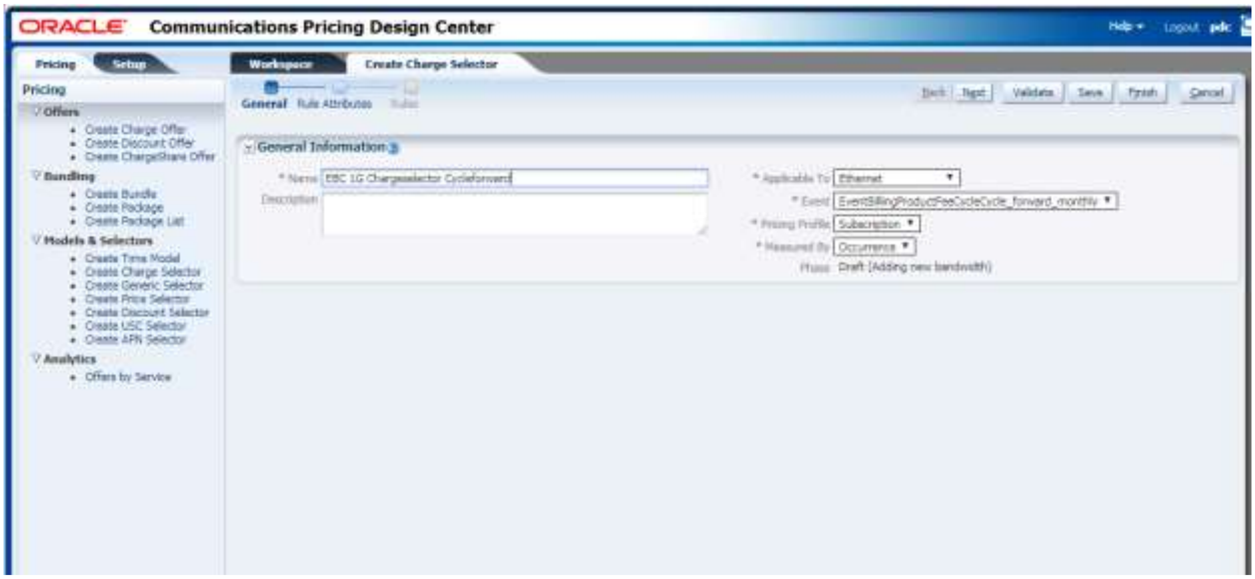
Pricelist in PDC is achieved by using **Charge Selector** for charges. **USAGE_TYPE** element in charge selector rule attributes denotes the pricelist.

Creating Charge Selector from UI

To create charge selector in PDC UI, login to PDC system and execute the following

- Select ***Create Charge Selector*** on the left navigation bar

- Provide a ***Name*** to the Charge Selector (e.g EBC 1G Chargeselector Cycleforward)
- Provide a ***Description***
- Select appropriate values for ***Applicable To*** and ***Event***



* Name: EBC 1G Chargeselector Cycleforward

Description:

* Applicable To: Ethernet

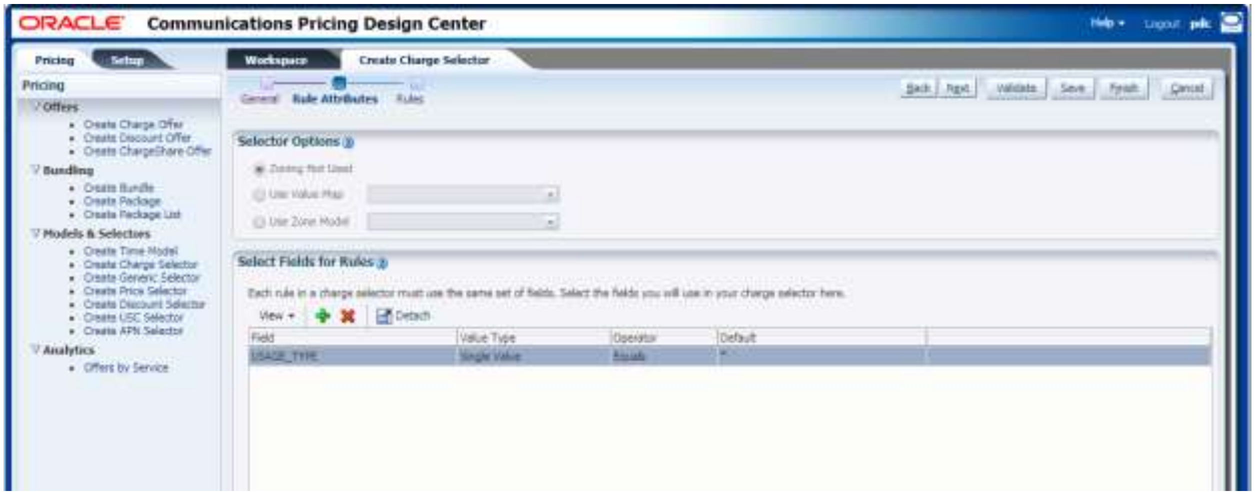
* Event: EventBillingProductFeeCycleCycle_forward_monthly

* Pricing Profile: Subscription

* Measured By: Occurrence

Phase: Draft (Adding new bandwidth)

- Select ***Next***
- On the bottom right pane with heading "***Select Fields for Rules***" select "**+**"
- Expand ***Event***, and locate ***USAGE_TYPE*** from the list. Add with default values



● Zoning Not Used

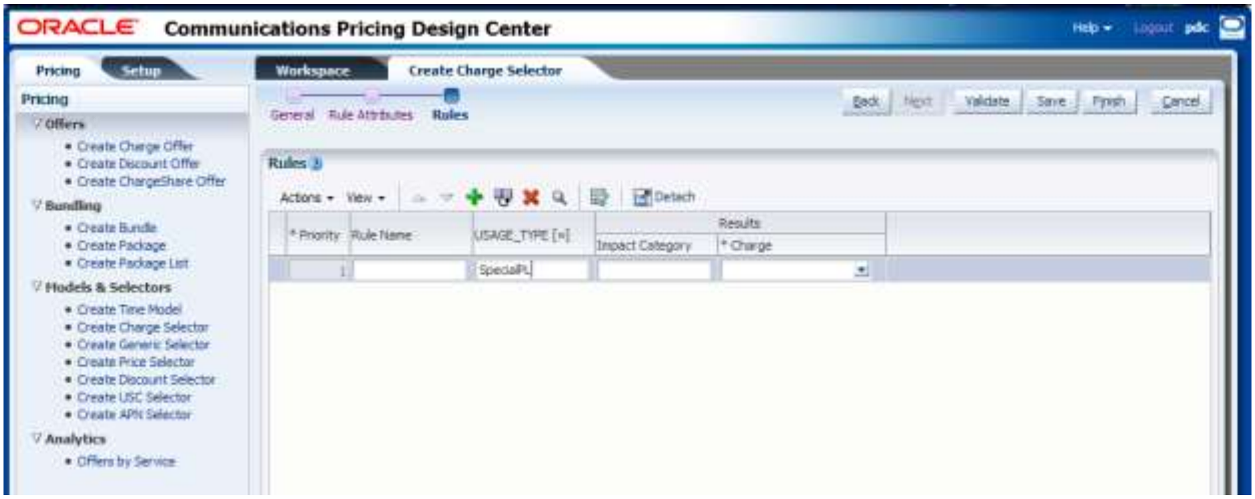
○ Use Value Map

○ Use Zone Model

Each rule in a charge selector must use the same set of fields. Select the fields you will use in your charge selector here.

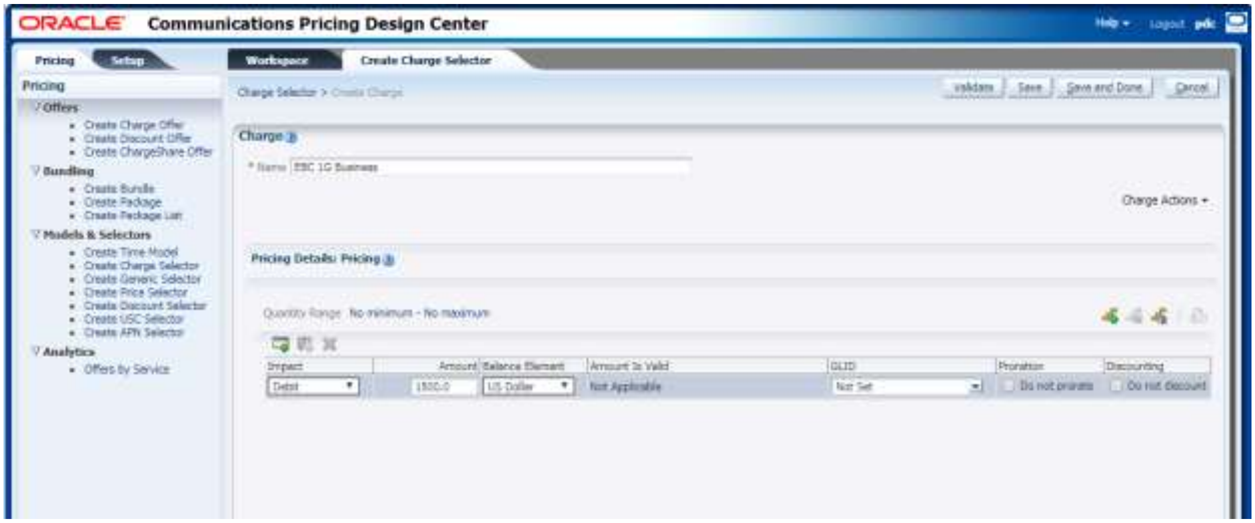
Field	Value Type	Operator	Default
USAGE_TYPE	Single Value	Equals	*

- Select ***Next***
- In the section with heading "***Rules***" select "**+**"
- Provide a value for ***Rule Name***
- Provide a value for ***USAGE_TYPE***. This value is the new pricelist name for PDC/BRM (Below example it's SpecialPL)



Priority: 1; USAGE_TYPE [=]: SpecialPL

- In “* Charge” column, select ***Create*** in dropdown menu
- Provide a ***Name*** (e.g EBC 1G Business) and provide ***Pricing Details***.


ORACLE Communications Pricing Design Center
Help Logout pdc
Pricing Setup
Workspace Create Charge Selector
Charge Selector > Create Charge
Validate Save Save and Done Cancel
Charge
Charge Actions
Pricing Details: Pricing
Quantity Range No minimum - No maximum
Impact Amount Balance Element Amount Is Valid GLID Proration Discounting

* Name: EBC 1G Business

Impact: Debit; Amount: 1500.0; Balance Element: US Dollar; Amount Is Valid: Not Applicable; GLID: Not Set; Proration: ☐ Do not prorate; Discounting: ☐ Do not discount

- Save the charge selector by selecting ***Save and Done***.
- The next step is create a charge offer and associate with charge selector that was just created.
- Once create charge offer is validated and submitted, it will trigger Synchronized Product process flow to sync product and charge offer to CPQ.

Creating Charge Selector from Catalog Operations

Catalog operations charge selector is denoted by json element **charges** in the payload. Element **name** under **chargePlans** of **charges** denotes the pricelist name.

Sample json payload for the above mentioned charge selector

```
{
        "charges": [{
                "name": "EBC 1G Chargeselector Cycleforward",
                "description": "EBC 1G Chargeselector Cycleforward",
                "applicableTo": "Ethernet",
                "type": "RECURRING",
                "frequency": "MONTHLY",
                "chargePlans": [{
                        "name": " EBC 1G Business",
                        "pricingPlan": " SpecialPL"
                }]
        }]
}
```

Verify Price List in CPQ

When a charge offer using charge selector is synchronized from PDC/BRM to CPQ, entry will be populated in OC_PRICELIST.

Navigate to CPQ data tables, OC_PRICELIST, and verify that value provided in PDC ***USAGE_TYPE*** is synced to this data table.

Create a quote in CPQ with the new price list:

- For product that is configured in the new price list, verify that that correct prices are populated for the transaction line(s)
- If a product in the quote does not have a price in the selected price list, verify that DEFAULT price list is populated.

2.2. Integrate with an External Account Master

The reference solution is pre-integrated with Oracle Engagement Cloud R17D. Changes to the following ICS integration projects is required to replace OEC with an alternate account master:

- CommsC3_CPQRetrieveAccDetails – This ICS integration project is invoked from CPQ to retrieve account details to include in CPQ transaction.
- CommsC3_AIARetrieveAcctDetails – This ICS integration is triggered from AIA QueryCustomerPartyListICSProvABCSImpl to retrieve account and billing profile details.

2.2.1. CommsC3_CPQRetrieveAcctDetails

This integration is invoked by CPQ to retrieve the list of accounts. It uses the opportunity name, and returns all the account information in a single CPQ attribute, ocFullAccountList.
When using a different external account master system, this integration must return the account information in the same JSON format:

```
{
        "opportunityId": "",
        "opportunityName": "",
        "customers": [{
                "account": {
                        "name": "",
                        "number": "",
                        "ID": "",
                        "firstName": "",
                        "lastName": "",
                        "email": "",
                        "address": "",
                        "city": "",
                        "state": "",
                        "postalCode": "",
                        "country": ""
                },
                "billing": {
                        "name": "",
                        "number": "",
                        "ID": "",
                        "profileName": "",
                        "profileId": ""
                },
                "service": [{
                        "name": "",
                        "id": "",
                        "number": "",
                        "addressId": ""
                }]
        }]
}
```

CPQ is expecting data for an owner account, a billing account, and one or more service accounts. Depending on the system and data available, some of these, for example the owner and billing accounts, could be the same.

If any errors are encountered, an error could instead be returned. CPQ will present this information to the user. It is returned as JSON in the same ocFullAccountList attribute:

```
{
    "error": "Error text to display"
}
```

Modify this flow by:

- Select and clone the CommsC3_ CPQRetrieveAcctDetails
- Provide the same integration name with a new version. Note that if a different name is provided, the corresponding CPQ integration will need to be changed to call this new name.
- Edit the new integration. The calls to OEC to retrieve the various account information will need to be removed and replaced. The final mapping back to the CPQ attribute will need to be modified to use new data from the alternate account system.
- If an error is encountered in the flow, the error JSON can be returned instead.

2.2.2. CommsC3_AIARetrieveAcctDetails

This integration is invoked by AIA QueryCustomerPartListICSProvABCSImpl composite to retrieve account and billing profile details. It receive one or more account identifier and zero or more billing profile identifier, and query details for all identifiers, and invoke AIA composite to provide query results.

Follow these steps to integrate with an alternate account master:

- Select and clone CommsC3_AIARetrieveAcctDetails in ICS server
- Provide the same integration name (CommsC3_AIARetrieveAcctDetails) and a new version for the new integration project. (**Note**: Providing the same name will avoid the need to change ICS endpoint URL in AIA).
- Edit the new integration, and remove and replace calls to OEC with alternate account master.
- The ICS integration must support input XML message structure defined in AccountEntity.wsdl file, and response to AIA following XML message structure defined in CustomerAccount.wsdl.
- Correlation ID received from AIA must be provided in the response message.
- AIA composite uses several DVM to retrieve data mapping between OEC and BRM. Update the following DVM files to provide data support in alternate account master. Data should be provided in CPQ_01 column:
    - STATE.dvm
    - ADDRESS_COUNTRYID.dvm
    - CUSTOMERPARTY_ACCOUNTTYPECODE.dvm
    - CUSTOMERPARTY_BILLPROFILE_FREQUENCYCODE.dvm
    - CUSTOMERPARTY_PAYPROFILE_PAYMETHODCODE.dvm

  DVM files are located in MDS /apps/AIAMetaData/dvm.

Table below contains a sample input message structure receive from AIA

```
   <CustomerAccountRequest>
      <tns:CorrelationId>37343034343433343839303739343631</tns:CorrelationId>
      <tns:Header>
              <tns:Id>36875186</tns:Id>
      </tns:Header>
      <tns:Accounts>
              <tns:Account>
                      <tns:Id>400000002506101</tns:Id>
              </tns:Account>
              <tns:Account>
                      <tns:Id>400000002506103</tns:Id>
              </tns:Account>
      </tns:Accounts>
      <tns:BillingProfiles>
              <tns:BillingProfile>
                      <tns:Id>400000002743471</tns:Id>
              </tns:BillingProfile>
      </tns:BillingProfiles>
 </CustomerAccountRequest>
```

Table below contains a sample response XML message from ICS to AIA

```
<CustomerAccountResponse>
        <nstrgmpr:CorrelationId>37343034343433343839303739343631</nstrgmpr:CorrelationId>
        <nstrgmpr:Accounts>
                <nstrgmpr:Account>
                        <nstrgmpr:Id>400000002506101</nstrgmpr:Id>
                        <nstrgmpr:Name>A4-Billing</nstrgmpr:Name>
        <nstrgmpr:Number>45074</nstrgmpr:Number>
        <nstrgmpr:Status/>
        <nstrgmpr:Type>Residential</nstrgmpr:Type>
        <nstrgmpr:Class>Customer</nstrgmpr:Class>
        <nstrgmpr:Currency>USD</nstrgmpr:Currency>
        <nstrgmpr:Organization>Company-A4</nstrgmpr:Organization>
        <nstrgmpr:PriceList/>
        <nstrgmpr:PrimaryAddress>
            <nstrgmpr:Id>400000002753110</nstrgmpr:Id>
            <nstrgmpr:Line1>55 Billing Way</nstrgmpr:Line1>
            <nstrgmpr:Line2/>
            <nstrgmpr:Line3/>
            <nstrgmpr:Line4/>
            <nstrgmpr:City>San Francisco</nstrgmpr:City>
            <nstrgmpr:State>CA</nstrgmpr:State>
            <nstrgmpr:Country>US</nstrgmpr:Country>
            <nstrgmpr:PostalCode>94023</nstrgmpr:PostalCode>
        </nstrgmpr:PrimaryAddress>
        <nstrgmpr:PrimaryContact>
            <nstrgmpr:Id>400000002517105</nstrgmpr:Id>
            <nstrgmpr:Salutation>Mr.</nstrgmpr:Salutation>
            <nstrgmpr:FirstName>Pitt</nstrgmpr:FirstName>
            <nstrgmpr:MiddleName/>
            <nstrgmpr:LastName>Mally</nstrgmpr:LastName>
            <nstrgmpr:Email>pitt.mally@att.com</nstrgmpr:Email>
            <nstrgmpr:CellPhone/>
            <nstrgmpr:WorkPhone/>
            <nstrgmpr:HomePhone/>
        </nstrgmpr:PrimaryContact>
    </nstrgmpr:Account>
    <nstrgmpr:Account>
        <nstrgmpr:Id>400000002506103</nstrgmpr:Id>
        <nstrgmpr:Name>A4-SiteA</nstrgmpr:Name>
        <nstrgmpr:Number>45075</nstrgmpr:Number>
        <nstrgmpr:Status/>
        <nstrgmpr:Type>Residential</nstrgmpr:Type>
        <nstrgmpr:Class>Customer</nstrgmpr:Class>
        <nstrgmpr:Currency>USD</nstrgmpr:Currency>
        <nstrgmpr:Organization>Company-A4</nstrgmpr:Organization>
        <nstrgmpr:PriceList/>
        <nstrgmpr:PrimaryAddress>
            <nstrgmpr:Id>400000002753111</nstrgmpr:Id>
            <nstrgmpr:Line1>69 SiteA Drive</nstrgmpr:Line1>
            <nstrgmpr:Line2/>
            <nstrgmpr:Line3/>
            <nstrgmpr:Line4/>
            <nstrgmpr:City>San Jose</nstrgmpr:City>
            <nstrgmpr:State>CA</nstrgmpr:State>
            <nstrgmpr:Country>US</nstrgmpr:Country>
            <nstrgmpr:PostalCode>94088</nstrgmpr:PostalCode>
        </nstrgmpr:PrimaryAddress>
        <nstrgmpr:PrimaryContact>
            <nstrgmpr:Id>400000002517106</nstrgmpr:Id>
            <nstrgmpr:Salutation>Mr.</nstrgmpr:Salutation>
            <nstrgmpr:FirstName>Ian</nstrgmpr:FirstName>
            <nstrgmpr:MiddleName/>
            <nstrgmpr:LastName>Mally</nstrgmpr:LastName>
            <nstrgmpr:Email>ian.mally@att.com</nstrgmpr:Email>
            <nstrgmpr:CellPhone/>
            <nstrgmpr:WorkPhone/>
            <nstrgmpr:HomePhone/>
        </nstrgmpr:PrimaryContact>
    </nstrgmpr:Account>
</nstrgmpr:Accounts>
<nstrgmpr:BillingProfiles>
    <nstrgmpr:BillingProfile>
        <nstrgmpr:Id>400000002743471</nstrgmpr:Id>
        <nstrgmpr:Name>A4-Billing</nstrgmpr:Name>
        <nstrgmpr:Address>
            <nstrgmpr:Id>400000002753110</nstrgmpr:Id>
```

```
                                        <nstrgmpr:Line1>55 Billing Way</nstrgmpr:Line1>
                                        <nstrgmpr:Line2/>
                                        <nstrgmpr:Line3/>
                                        <nstrgmpr:Line4/>
                                        <nstrgmpr:City>San Francisco</nstrgmpr:City>
                                        <nstrgmpr:State>CA</nstrgmpr:State>
                                        <nstrgmpr:Country>US</nstrgmpr:Country>
                                        <nstrgmpr:PostalCode>94023</nstrgmpr:PostalCode>
                              </nstrgmpr:Address>
                              <nstrgmpr:BillCurrency>USD</nstrgmpr:BillCurrency>

          <nstrgmpr:PaymentMethod>ORA_ATC_PAY_METHOD_BILL</nstrgmpr:PaymentMethod>
                              <nstrgmpr:Contact>
                                        <nstrgmpr:Id>400000002517105</nstrgmpr:Id>
                                        <nstrgmpr:Salutation>Mr.</nstrgmpr:Salutation>
                                        <nstrgmpr:FirstName>Pitt</nstrgmpr:FirstName>
                                        <nstrgmpr:MiddleName/>
                                        <nstrgmpr:LastName>Mally</nstrgmpr:LastName>
                                        <nstrgmpr:Email>pitt.mally@att.com</nstrgmpr:Email>
                                        <nstrgmpr:CellPhone/>
                              </nstrgmpr:Contact>
                    </nstrgmpr:BillingProfile>
          </nstrgmpr:BillingProfiles>
</CustomerAccountResponse>
```

2.3. Adding a new Product Offer in existing Instant Business Connect Product Model

In this section, we will walk through a scenario to add a new **EBC 10GB Bandwidth** product to existing Instance Business Connect product model.

Table below show the product model included in reference solution:

#	Entity Name	Offer Type	Product Spec
1	IBC Market Offer	Promotion	
1.1	IBC SLA	Simple Product	CE SLA PS
1.2	IBC 10% SLA Discount	Simple Product	CE Discount PS
1.3	Instant Business Connect	Bundle	CE Service PS
1.3.1	IBC Endpoint (BO)	Service Bundle	CE Endpoint PS
1.3.1.1	EBC xxx Bandwidth	Simple Product	CE CoS Bandwidth PS
1.3.1.2	Basic Uni Access	Simple Product	UNI Access PS
1.3.1.3	Endpoint Discount	Simple Product	CE Discount PS

A product can be added to existing product model by following methods:

1. Using PDC UI
2. Using Catalog Operations

User can use any method to introduce a new product, but be aware to continue same method for introduction/update of other products.

Following sections will detail adding a new bandwidth product “**EBC 10GB Bandwidth**” to IBC Endpoint BO.

2.3.1. Using PDC UI

Create charge offer in PDC

This section will detail the PDC workflow for adding a new product. Once this is complete, it will automatically sync using the Synchronize Product flow.

Creating Charge Offers in PDC:

- Login to PDC system
- Select ***Create Charge Offer*** on the left navigation bar
- Provide a ***name*** to the Charge Offer
- Provide a ***description***
- Select the ***type*** of charge offer
- Select ***Applicable To***. Once **Applicable To** is populated, ***Add Charge*** will be enabled in the bottom right pane
- To create charge offer, select one of these options:
  - With charge selector, select ***Add Charge -> Add Existing Charge Selector***, and select the desired charge selector.
  - Without charge selector, select ***Add Charge -> Add New Charge***. Provide **Charge Category**, ***Charge Type*** in pop up and select ***OK***. A new section will be added in the bottom pane with name ***Pricing Details: Pricing***. Provide a value in the ***Amount*** column.
- Save the workspace.
- Validate and submit workspace. This will trigger Synchronize Product flow to synchronize charge offer to CPQ.

Refer to PDC user guide for more details https://docs.oracle.com/cd/E22185_01/index.htm

Verify in CPQ

To verify that product is synced into CPQ, login to CPQ to verify the following

- OC_PRICELIST data table – Verify table contains new charge offer created in PDC
- Parts – search parts, and verify that "Part Search Results" contains new charge offer created in PDC

Enrich in CPQ

a. Data Tables

Add a new row for **EBC 10GB bandwidth**

- **OracleComms**

1. OC_PART_ATTRIBUTES

PartName	ProductSpecification
EBC 10GB Bandwidth	CE Cos Bandwidth PS

- **BOM Tables**

1. Oracle_BomItemDef

VariableName	SequenceNum	ItemId	Name	PartNumber	ParentVariableName	RootVariableName
EBC 10GB Bandwidth	1311	EBC 10GB Bandwidth	EBC 10GB Bandwidth	EBC 10GB Bandwidth	IBC Endpoint (BO)	IBC Market Offer-root

2. Oracle_BomItemMap

VariableName	BomItemVarName	ConfAttrVarName	ConfigAttrVarValue	ParentBomMapVarName	Comments
EBC 10GB Bandwidth	EBC 10GB Bandwidth	ocBandwidth	10 Gbps	ocEthernetServices:ocMultiSiteInstantBusinessConnect:ocInstantBusinessConnect:ocIBCBOMMappingRule	N

3. Oracle_BomAttrDef

VariableName	Name	BomItemVarName	DataType	RootBomItemVarName	ExtendedAttribute	LineAttributeName
EBC 10GB Bandwidth	bandwidth	EBC 10GB Bandwidth	String	IBC Market Offer-root	true	ocBandwidth

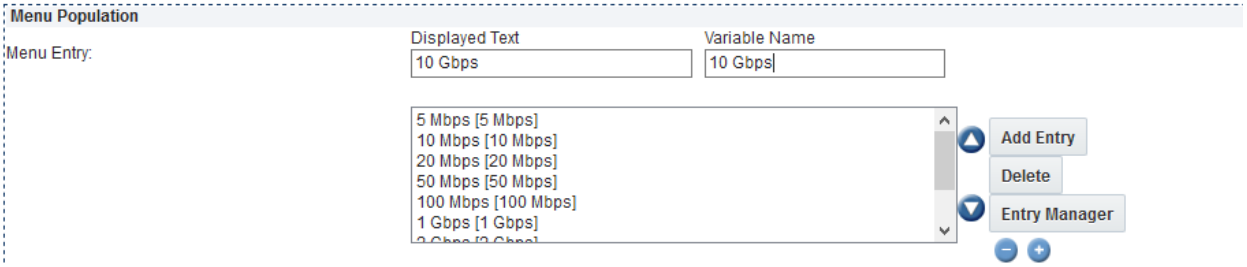
4. Oracle_BomAttrMap

VariableName	TargetVariableName	ConfAttrVarName	BomItemMapVarName	ParentBomMapVarName
EBC 10GB Bandwidth	EBC 10GB Bandwidth	ocBandwidth	EBC 10GB Bandwidth	ocEthernetServices:ocMultiSiteInstantBusinessConnect:ocInstantBusinessConnect:ocIBCBOMMappingRule

Deploy changes after adding row to BOM tables.

b. Product Model bandwidth Menu

- Navigate to bandwidth configurable attribute under “Model: Ethernet Services > Multi Site Instant Business Connect > Instant Business Connect”
- Introduce a new menu entry with *Displayed Text* as 10 Gbps and *Variable Name* as 10 Gbps and click on Add Entry.



Menu Population

Menu Entry:

Displayed Text: 10 Gbps

Variable Name: 10 Gbps

5 Mbps [5 Mbps]
10 Mbps [10 Mbps]
20 Mbps [20 Mbps]
50 Mbps [50 Mbps]
100 Mbps [100 Mbps]
1 Gbps [1 Gbps]

- Click on Update.
- Deploy the product model.

2.3.2. Using Catalog Operations

Create charge offer using REST operation

- Locate **CommsC3_Catalog** Integration calling url from ICS. Sample url will look like https://ics-url-prefix/COMMSC3_CATALOG/v01/metadata

- Use any REST client (example POSTMAN, SOAP, cURL etc.) to use **POST** operation on above endpoint suffixed with /catalog with following json data in body. Example https://ics-url-prefix/COMMSC3_CATALOG/v01/metadata/catalog.
- It’ll need Basic Authorization with ICS username and password

```
{
    "enablePDCChangesetReview": "false",
    "bundles": [{
            "name": "IBC Endpoint (BO)",
            "products": [{
                    "name": "EBC 10GB Bandwidth"
            }]
    }],
    "products": [{
            "name": "EBC 10GB Bandwidth",
            "description": "EBC 10GB Bandwidth",
            "applicableTo": "Ethernet",
            "specification": "CE CoS Bandwidth PS",
            "type": "SUBSCRIPTION",
            "charges": [{
                    "name": "CHGSLDemo10GBBandwidthRecurring",
                    "type": "Recurring",
                    "frequency": "Monthly"
            }],
            "attributes": [{
                    "name": "bandwidth",
                    “type”: “String”
            }]
    }],
    "charges": [{
            "name": "CHGSLDemo10GBBandwidthRecurring",
            "description": "CHGSLDemo10GBBandwidthRecurring",
            "applicableTo": "Ethernet",
            "type": "Recurring",
            "frequency": "Monthly",
            "chargePlans": [{
                    "name": "CHGRateBusinessDemo10GBBandwidthRecurring",
                    "pricingPlan": "BusinessPL"
            },
            {
                    "name": "CHGRateDefaultDemo10GBBandwidthRecurring",
                    "pricingPlan": "Default"
            }]
    }],
    "chargePlans": [{
            "name": "CHGRateBusinessDemo10GBBandwidthRecurring",
            "description": "CHGRateBusinessDemo10GBBandwidthRecurring",
            "applicableTo": "Ethernet",
            "type": "Recurring",
            "frequency": "Monthly",
            "rum": "check",
            "price": "3000",
            "currency": "USD"
```

```
        },
        {
                "name": "CHGRateDefaultDemo10GBBandwidthRecurring",
                "description": "CHGRateDefaultDemo10GBBandwidthRecurring",
                "applicableTo": "Ethernet",
                "type": "Recurring",
                "frequency": "Monthly",
                "rum": "check",
                "price": "3250",
                "currency": "USD"
        }]
    }
```

Above operation will create a changeset in PDC with the product along with charge selector and chargeRatePlan(s) which will be auto-submitted. Upon submission it triggers the synchronize product flow which will update CPQ parts and OC_PRICELIST.

Above operation also will include this product under **IBC Endpoint (BO)**

Verify in CPQ

a. Parts
    - A part with name EBC 10 GB Bandwidth would have created with needed details.

b. Data Tables
    - OracleComms

1. OC_BUNDLE

productName	parent
EBC 10 GB Bandwidth	IBC Endpoint (BO)

2. **OC_PART_ATTRIBUTES**

PartName	ProductSpecification
EBC 10 GB Bandwidth	CE Cos Bandwidth PS

3. **OC_PROD_ATTRIBUTES**

ProdutName	AttributeName	DataType
EBC 10 GB Bandwidth	bandwidth	String

4. **OC_PRICELIST**

priceListName	productName	priceChargeType	currency	listPrice
Business Pricelist	EBC 10 GB Bandwidth	RECURRING	USD	3000
DEFAULT	EBC 10 GB Bandwidth	RECURRING	USD	3250

- **BOM Tables**

5. **Oracle_BomItemDef**

VariableName	SequenceNum	ItemId	Name	PartNumber	ParentVariableName	RootVariableName
EBC 10GB Bandwidth-xxxx	1311	EBC 10GB Bandwidth-xxxx	EBC 10GB Bandwidth	EBC 10GB Bandwidth	IBC Endpoint (BO)-x1x1	IBC Market Offer-root

6. **Oracle_BomAttrDef**

VariableName	Name	BomItemVarName	DataType	RootBomItemVarName	ExtendedAttribute	LineAttributeName
EBC 10GB Bandwidth-bandwidth-yyyy	bandwidth	EBC 10GB Bandwidth-xxxx	String	IBC Market Offer-root		

Enrich in CPQ

a. Data Tables

Add a new row for **EBC 10GB bandwidth**

- BOM Tables

1. **Oracle_BomItemMap**

VariableName	BomItemVarName	ConfAttrVarName	ConfigAttrVarValue	ParentBomMapVarName	Comments
EBC 10GB Bandwidth	EBC 10GB Bandwidth-xxxx	ocBandwidth	10 Gbps		N

2. **Oracle_BomAttrMap**

VariableName	TargetVariableName	ConfAttrVarName	BomItemMapVarName	ParentBomMapVarName
EBC 10GB Bandwidth	EBC 10GB Bandwidth-bandwidth-yyyy	ocBandwidth	EBC 10GB Bandwidth	ocEthernetServices:ocMultiSiteInstantBusinessConnect:ocInstantBusinessConnect:ocIBCBOMMappingRule

Deploy changes after adding row to BOM tables.

Modify in following tables

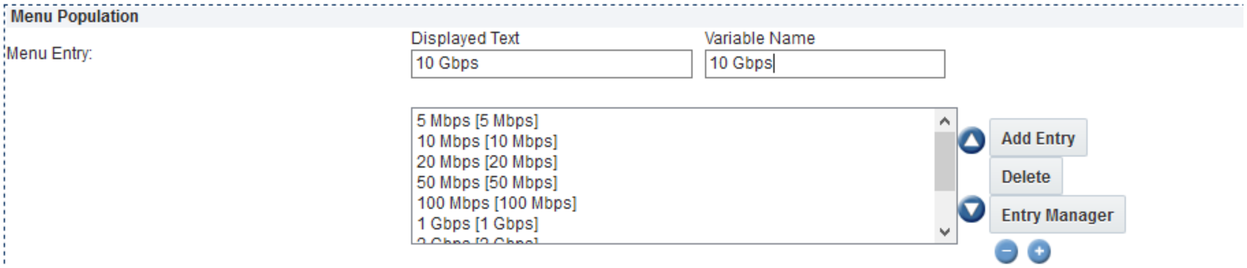
- BOM Tables

1. **Oracle_BomAttrDef**

VariableName	Name	BomItemVarName	DataType	RootBomItemVarName	ExtendedAttribute	LineAttributeName
EBC 10GB Bandwidth-yyyy	bandwidth	EBC 10GB Bandwidth-xxxx	String	IBC Market Offer-root	true	ocBandwidth

b. Product Model bandwidth Menu

- Navigate to bandwidth configurable attribute under “Model: Ethernet Services > Multi Site Instant Business Connect > Instant Business Connect”
- Introduce a new menu entry with *Displayed Text* as 10 Gbps and *Variable Name* as 10 Gbps and click on Add Entry.



Menu Population

Menu Entry:

Displayed Text: 10 Gbps

Variable Name: 10 Gbps

5 Mbps [5 Mbps]
10 Mbps [10 Mbps]
20 Mbps [20 Mbps]
50 Mbps [50 Mbps]
100 Mbps [100 Mbps]
1 Gbps [1 Gbps]

- Click on Update.
- Deploy the product model.

2.4. Billing Care UI Integration

The billing care UI is an Oracle Engagement Cloud implementation to embed the BRM Billing Care pages. Customizations must be performed on both BRM and Engagement Cloud for this to function.

1. Billing Care
    - The Billing Care web service must be configured to allow display of the Billing Care UI through external pages
2. Engagement Cloud
    - The cross reference web service will be registered with engagement cloud so it can be used to determine the mappings of the account IDs between Engagement Cloud and Billing Care.
    - The URL for billing care is registered in Engagement Cloud so it can be easily reused between the billing care tabs.
    - The new tabs for billing care are added in Engagement Cloud.
    - A new role can be added in Engagement Cloud to control which users get the layout with the new tabs.

2.4.1. BRM Billing Care Configuration

The BRM Billing Care web service is configured with security turned on by default. This includes the option to prevent clickjacking through the use of the X-Frame-Options HTML header. By default, this is set to SAMEORIGIN, which will prevent the billing care UI from being shown if included in an iframe from another location. It is up to the user to determine the correct security configuration for this in their environment, but may include:

1. Use a new plan.xml when deploying the billing care web service to customize the X-Frame-Options value. This would allow overriding it to support ALLOW-FROM, to enable viewing from the Engagement Cloud system on IE compatible browsers.
2. The web.xml inside billing care can be modified to remove the use of the X-Frame-Options, although this is not recommended for customer deployments.

The following is an example plan.xml for customizing the behavior:

```
<?xml version='1.0' encoding='UTF-8'?>
```

```
<deployment-plan xmlns="http://xmlns.oracle.com/weblogic/deployment-
plan" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.oracle.com/weblogic/deployment-
plan http://xmlns.oracle.com/weblogic/deployment-plan/1.0/deployment-
plan.xsd" global-variables="false">
  <application-name>BillingCare.war</application-name>
  <variable-definition>
      <variable>
          <name>newVariable</name>
          <value>Key123456</value>
      </variable>
  </variable-definition>
  <module-override>
    <module-name>BillingCare.war</module-name>
    <module-type>war</module-type>
                <module-descriptor external="false">
      <root-element>web-app</root-element>
      <uri>WEB-INF/web.xml</uri>
      <variable-assignment>
         <name>newVariable</name>
         <xpath>/web-app/filter/[filter-
name="ClickjackFilterSameOrigin"]/filter-class</xpath>
      </variable-assignment>
    </module-descriptor>
  </module-override>
</deployment-plan>
```

2.4.2. Engagement Cloud Customizations

Before beginning the customizations in Engagement Cloud, the user must first create and activate a sandbox for the billing care customizations. Refer to Overview of Working in an Oracle Sales Cloud Sandbox for instruction on sandbox creation

2.4.2.1. Cross Reference Registration

The cross reference web service will be used by the billing care tabs. This web service will provide the mappings from the Engagement Cloud IDs to those used by Billing Care. This must be registered in the Application Composer before it can be used in the layout tabs.

Navigate to the "Navigator->Tools->Application Composer->Common Setup->Web Services". The Application Composer is located in the Tools section of the Navigator menu.

Create a new web service reference, selecting the type "REST", with a variable name XREF_REST_LOOKUP. For this new web service reference, specify the URL as "https://HOST:PORT/integration/flowapi/rest/COMMSC3_QUERYAIAXR_POST/v01/xref/querypost", where HOST and PORT are substituted by the actual hostname and port for the ICS integration system with the CommsC3_PostAIAXRef integration. The correct host and port can be seen by looking at the details of the activated integration in ICS.

Create a new credential key and specify the login credentials for ICS.

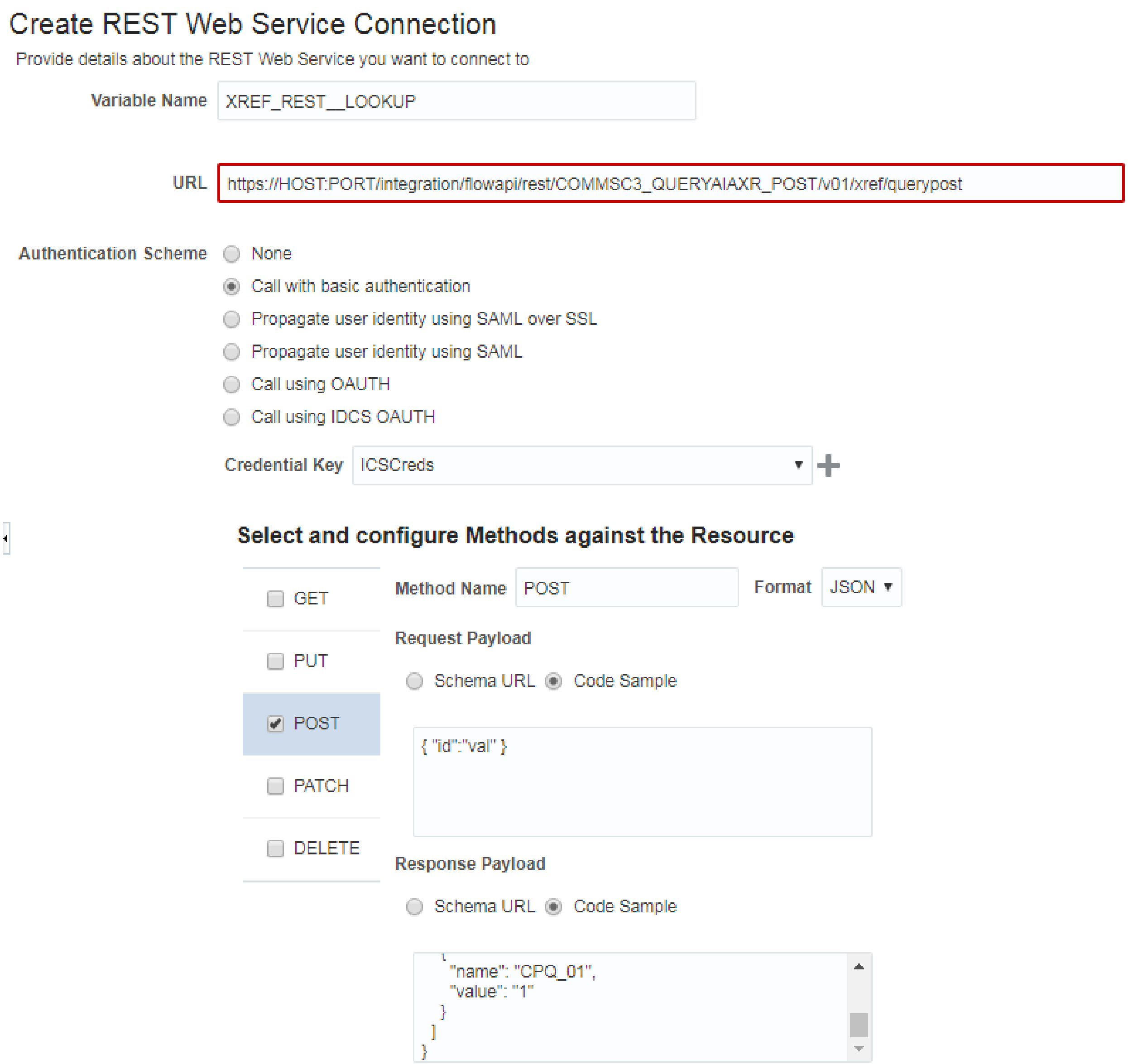
The "POST" method should be selected, using the JSON format. The following can be used as the code sample for the request payload:

```
{

  "id":"val"

}
```

The following can be used as the code sample for the response payload:

```
{
  "schemeID": "CUSTOMERPARTY_ACCOUNTID",
  "schemeAgency":
  [
    {
      "name": "COMMON",
      "value": "1"
    },
    {
      "name": "BRM_01",
      "value": "0.0.0.1 /account 1 0"
    },
    {
      "name": "CPQ_01",
      "value": "1"
    }
  ]
}
```

The following shows a sample web service configuration details page:


Create REST Web Service Connection

Provide details about the REST Web Service you want to connect to

Variable Name XREF_REST__LOOKUP

URL https://HOST:PORT/integration/flowapi/rest/COMMSC3_QUERYAIAXR_POST/v01/xref/querypost

Authentication Scheme
- ○ None
- ● Call with basic authentication
- ○ Propagate user identity using SAML over SSL
- ○ Propagate user identity using SAML
- ○ Call using OAUTH
- ○ Call using IDCS OAUTH

Credential Key ICSCreds

Select and configure Methods against the Resource

- ☐ GET
- ☐ PUT
- ☑ POST
- ☐ PATCH
- ☐ DELETE

Method Name POST **Format** JSON

Request Payload

○ Schema URL ● Code Sample

```
{ "id":"val" }
```

Response Payload

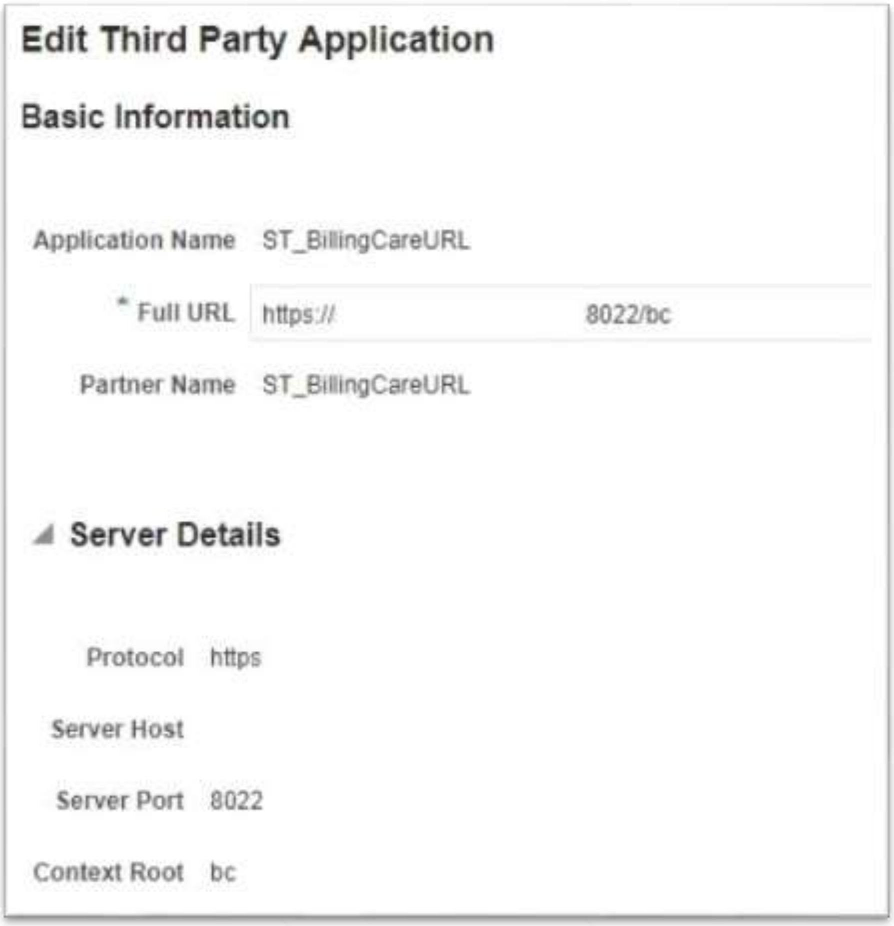
○ Schema URL ● Code Sample

```
    "name": "CPQ_01",
    "value": "1"
   }
  ]
}
```

2.4.2.2. Configuring the Billing Care URL

Navigate to the "Navigator->Setup and Maintenance->Tasks (menu on the right)->Manage Custom Setup Content->Topology Definition->Manage Third Party Applications"

Create a new third party application configuration fo Billing Care. Use the application name ST_BillingCareURL, and specify the URL as "https://BRM_HOST:PORT/bc", where BRM_HOST and PORT are substituted with the hostname and port for Billing Care.



Edit Third Party Application

Basic Information

Application Name ST_BillingCareURL

* Full URL https:// 8022/bc

Partner Name ST_BillingCareURL

Server Details

Protocol https

Server Host

Server Port 8022

Context Root bc

2.4.2.3. Creating a New Layout with Billing Care Subtabs

Navigate to the "Navigator->Application Composer->" (Application Composer is in the Tools section of the Navigator).

Expand the "Objects->Standard Objects->Account" in the tree and select the "Pages" link.

One of the existing details page layouts can be used as the base for adding the new tabs. Select the desired layout (e.g. ORATC Detail Page Layout), and then pick "Duplicate" and assign a name to the new layout (e.g. Digital BSS Details Layout).

Next, to add the new subtabs, scroll to the end of the subtabs and select the "Add" subtab, then pick "Web content".

For the first subtab, give a label of "Billing Details", select an icon and enter the follow for the script.

```
// Get the base URL as configured
def myURL =
adf.util.__ORATC__O_INT_SBL_GetURLFromTopologyManager("ST_BillingCareUR
L");
Map requestMap = new HashMap();
requestMap.put("id", String.valueOf(PartyId));
Map lookupMap = (Map)adf.webServices.XREF_REST_LOOKUP.POST(requestMap);
```

```
Collection<Map> schemeAgencies =
(Collection<Map>)lookupMap.get("schemeAgency");
String value ="NO_VALUE";
String name = "";
for(Map  schemeAgency : schemeAgencies) {
  name = schemeAgency.get("name");
  if("BRM_01".equals(name)) {
      value =  schemeAgency.get("value");
      String[] accountParts = value.split(" ");
      myURL = myURL + "/index_embedded.html#billDetails?accountId=" +
              accountParts[0] + "+-" + accountParts[1].substring(1)
+ "+" + accountParts[2];
      return (myURL);
  }
}
println("No BRM account ID for this account");
```

Once that subtab is saved, select the “Add” subtab again.

For this subtab, give a label of “Billing Assets”, select an icon and enter the following for the script.

```
// Get the base URL as configured
def myURL =
adf.util.__ORATC__O_INT_SBL_GetURLFromTopologyManager("ST_BillingCareUR
L");
Map requestMap = new HashMap();
requestMap.put("id", String.valueOf(PartyId));
Map lookupMap = (Map)adf.webServices.XREF_REST_LOOKUP.POST(requestMap);
if (lookupMap != null) {
   Collection<Map> schemeAgencies =
(Collection<Map>)lookupMap.get("schemeAgency");
   String value ="NO_VALUE";
   String name = "";
   for(Map  schemeAgency : schemeAgencies) {
     name = schemeAgency.get("name");
     if("BRM_01".equals(name)) {
         value =  schemeAgency.get("value");
         String[] accountParts = value.split(" ");
         myURL = myURL
+ "/index_embedded.html#customerAssets?accountId=" +
                 accountParts[0] + "+-" + accountParts[1].substring(1)
+ "+" + accountParts[2];
         return (myURL);
     }
   }
}
println("No BRM account ID for account " + PartyId);
```

2.4.2.4. Assigning Role to New Layout

Roles can be used to control when the new layout with the Billing Care tabs is shown. This is optional and the layout can be made the default for all users.

The Security Console is used to create a new role and assign to users. It is found under Navigator->Security Console. Select Roles, if it isn't selected by default, and then "Create Role". Give a role name (e.g. "Digital BSS Sales Role"), a role code (e.g. "DIGITAL_BSS_SALES_ROLE"), and a role category. The role category should be "CRM - Job Roles" for this role to show up as a layout selectable role. No security policies need to be chosen, so "Next" can be selected until presented with the option of assigning to users. Select "Add Users", then search and add all the desired users. This assignment of roles to users can also be done from the user accounts screen. Once this is completed, select "Next" and then "Save and Close".

The next step requires assigning the role to the layout. Navigate to the Application Composer (Navigator->Application Composer), select the account pages (Objects->Standard Objects->Account->Pages). From here, a role can be selected for the layout created above. This ensures the layout is only picked for users that have this role applied.

2.4.2.5. Publishing the Changes

All the changes made in Engagement Cloud are contained in the sandbox. The user can test these changes from within that sandbox, but other users will not see them. The next step is to publish the sandbox, done from the user account menu->Manage Sandboxes. It contains the action to publish the sandbox. Once this is done, the changes will be available to all users of Engagement Cloud.

2.5. Customer Care Oracle Engagement Cloud Customization

2.5.1. Display Synchronization Status

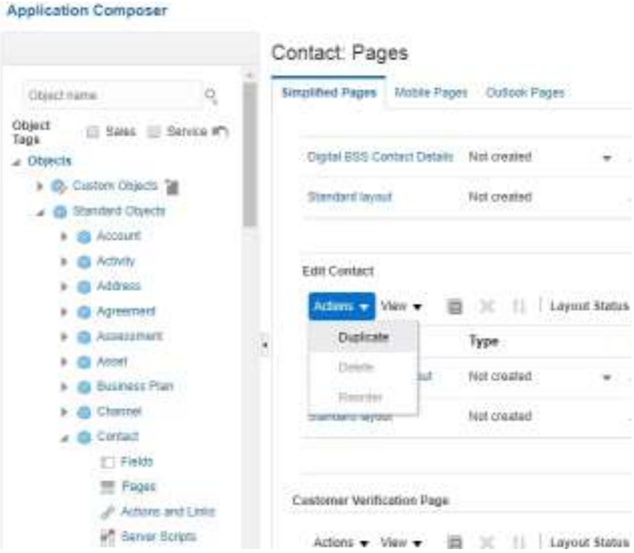
This chapter describes the steps to add new attribute to Oracle Engagement Cloud Account and Contact Overview user interface.

The CX Integration Pack Reference Solution for Synchronize Account Update from Oracle Engagement Cloud to BRM flow returns a synchronization status to Engagement Cloud. To display status message, use Oracle Engagement run time tool to customize the user interface.

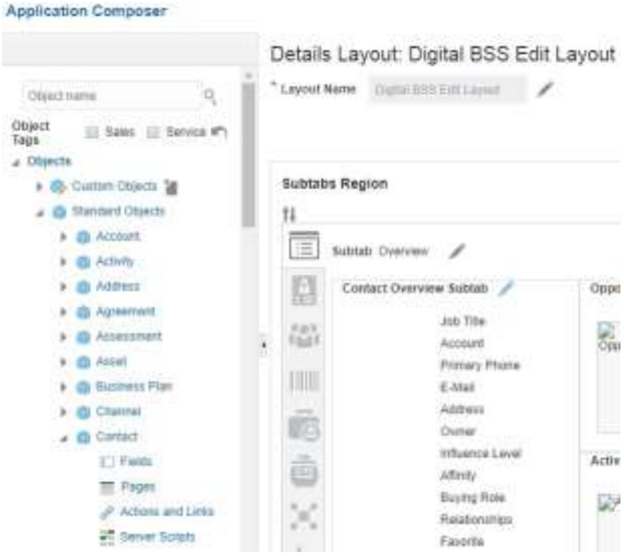
Here are the steps to customize Oracle Engagement Cloud user interface to display synchronization status message in Account and Contact overview page:

- Before beginning the customization, user must create and activate a sandbox for customization.
- Add new attribute to layout
    - Navigate to **Navigator** -> **Application Composer**
    - Expand **Objects** -> **Standard Objects** -> **Account**
    - Select **Pages**
    - Scroll to **Details Page Layouts section** and select layout created in billing care customization (for example: Digital BSS Details Layout)
    - Select **Edit** in **Account Overview subtab Form** section

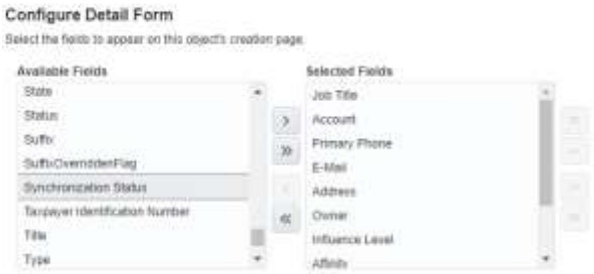
- Move **Synchronization Status** attribute from Available Fields column to Selected Fields column
- Select **Save and Close**
- Expand **Objects** -> **Standard Objects** -> **Contact**
- Select **Pages**
- Scroll to **Edit Contact**, select **Actions** -> **Duplicate**. User **Standard Layout** for source layout and assign a new to new layout (for example: Digital BSS Contact Detail Layout)



- Select **Save and Edit**
- Select **Edit** on **Contact Overview Subtab**



- Move **Synchronization Status** attribute from Available Fields column to Selected Fields column



Configure Detail Form

Select the fields to appear on this object's creation page.

   - Save changes
   - Optional, if custom role is define for Digital BSS customization, apply custom role to new layout.
- Navigate to Account detail page to verify that **Synchronization Status** is added to Account overview page.
- Navigate to Contact detail page to verify that **Synchronization Status** is added to Contact overview page
- Navigate to **user menu** -> **Manage Sandboxes** to publish changes made in sandbox

3. Related Trainings

This document serves only to demonstrate features in the installed Oracle Communications Digital BSS Reference Solution. For more in-depth training, please consult courseware available through Oracle University.

3.2. Oracle University Training and Courseware

Further training, including recordings illustrating the key features of Oracle Communications Digital BSS and the pre-built integrations are available through the Oracle University Web site at: http://education.oracle.com. The trainings are available in **Classroom** and **Live Virtual Class** formats.

For training on Oracle Communications Digital BSS on premises applications, navigate to Training > Industries > Oracle Communications.

- For OSM O2A related training, select Order and Service Management > OSM Foundations III: AIA Order to Activate Solution
- For Oracle AIA and pre-built integration related trainings, select AIA Communications
- For Oracle BRM and PDC related trainings, select Billing and Revenue Management

For training on Oracle Communications Digital BSS pre-integrated cloud applications

- Navigate to Training > Oracle Cloud > Oracle Platform as a Service (PaaS) > Integration for ICS related training
- Navigate to Oracle Cloud > Oracle CPQ Cloud (Configure, Price and Quote) for CPQ related training

3.3. Troubleshooting

ICS

To verify execution status on an ICS integration instance, login to ICS server.

- Select Monitor->Tracking. Provide CPQ order number in search field to find CommsC3_CPQRetrieveAcctDetails and CommsC3_CPQCreateSalesOrder instance for this CPQ order
- Select Monitor->Integrations to get a summary Received/Processed/Success/Errors messages.
- To enable more tracing, integration can be activated with Enable tracing” and “Include payload” for enable more logging information.
- To view errors and audit trail, select failed integration instance, and select menu->View Errors or menu->View Audit Trail.

AIA

AIA logs are located in AIA installed host. This is a sample AIA log files and directory:

- /private/aiacom_test_install/Middleware_wls/user_projects/domains/aiafp/servers/soa_server1/logs, where aiafp is domain name, and soa_server1 is SOA managed server name
- soa_server1.log
- soa_server-diagnostic.log

BRM

BRM logs are located in BRM installed host. This is a sample BRM log files and directory:

- /private/aiacom_test_install/brm/portal/7.5/var/cm/cm.pinlog
- /private/aiacom_test_install/brm/portal/7.5/var/dm_oracle/dm.pinlog

4. Acronyms

The following acronyms are used in this document:

Acronym	Word, Name, or Phrase	Context
BRM	Billing and Revenue Management	Oracle product
CRM	Customer Relationship Management	Enterprise software category
EM	Enterprise Manager	Oracle product
GUI	Graphical User Interface	Software interface
O2A	Order to Activate	Oracle solution
O2C	Order to Cash	Oracle solution
OSM	Order and Service Management	Oracle product
PIP	Process Integration Pack	Oracle AIA products
UI	User Interface	Software interface
AIA	Application Integration Architecture	Oracle AIA products
PDC	Pricing Design Center	Oracle product
OEM	Oracle Enterprise Manager	Oracle product
OSC	Oracle Sales Cloud	Oracle product
OEC	Oracle Engagement Cloud	Oracle product
ICS	Oracle Integration Cloud Service	Oracle product