Document Attributes

| Attribute | Value |
| --- | --- |
| Application ID /  Application Name | MOTS ID:  14724 / GCP (Global Computing Platform) |
| Owner | GCP: Ming Ho |
| Owner Contact Information | mh6892, 732-420-7673 |
| Other Attribute | Project ID:  288315 |

## 

Revision History

| Author | Date | Version # | Revision Description |
| --- | --- | --- | --- |
| Ming Ho | 8/10-14/2016 | 0.01 | <288315-US681337> Initial draft |
| Ming Ho | 8/30/2016 | 0.02 | <288315-US681337-upd1>  Remove PVC fields not in change\_pvc table |
| Ming Ho | 9/7/2016 | 0.03 | <288315-US681337-upd2>  Add table definition for tracking status |
| Ming Ho | 9/12/2016 | 0.04 | <288315-US681337-upd3>  Update logic retrieving Asset.service  Listed production SQL retrieving PVC records, with updates to address production tickets (duplicate PVC records; incorrect CER IP or PE IP; VPN values). This update was sent to Dev team a week ago. |
| Ming Ho | 9/13/2016 | 0.05 | <288315-US681337-upd3>  Update logic retrieving Asset.service for PVCs (SQLs fom code)  For service=GMIS, updated logic retrieving managedIndicator. |
| Ming Ho | 9/14/2016 | 0.06 | <288315-US681337-upd3>  Change the fields in response structure to O (optional).  Correct field names; |
| Ming Ho | 10/12/2016 | 0.07 | Removed/crossed out the section for tracking table that was designed by Dev team. It should not be part of the HLD. |
| Ming Ho | 11/7/2016 | 0.08 | 288315-QC 64083,65990  Anne Barrier: made code change to address issue of not data returned for ipv4. |
| Ming Ho | 11/17/2016 | 0.09 | <288315-QC73580>   * Issue: BMP format circuit values are incorrect. * Fix: use site\_id to match PVC and customer record.   E.G.:  Incorrect value:’^^DHEC^^^973^^^801^ATI ‘  Correct value: ‘^^DHEC^^^973801ATI^’ where ^=space. |
| Ming Ho | 3/12/2017 | 0.10 | <CR-162323> To resolve issue with CBUS web service (BAU flow) not able to retrieve PVC records, Add pvcRecoedSelector to the PVC structure and populate, load, sent to CM this field.  Logic setting this field is existing, with minor change.  The was planned to add under <288315.154436> but that CR was cancelled. |
| Archana Gattu | 6/2/2017 | 0.11 | Updates for PID 297409 |
| Archana Gattu | 6/28/2017 | 0.12 | 297409: Updates to support svl flag for FR\_ATM service |
| Archana Gattu | 7/24/2017 | 0.13 | 297409:Updates to DBA requirement to add a\_service and z\_service. |
| Archana Gattu | 7/26/2017 | 0.14 | 297409:Updates to increase vpn\_id fields in PVC table based on system test issue. |
| Archana Gattu | 7/31/2017 | 0.15 | Updates for defect 230097 for PID 297409 |
| Archana Gattu | 7/31/2017 | 0.16 | Updates for PID 297100(oct 2017 rel) |
| Ming Ho | 8/14/2017  8/17/2017  8/28/2017  9/28/2017 | 0.17 | <297409-workItem-251821>  Address issue of VPLS logic not retrieving the AVPN and MIS PVCs for vr1 routers  <297409-workItem-251821-2>  Update the logic for serviceType=VPLS, for retrieving customer records:  Stating executing both existing and new logic;  ~~Excluding cust\_id=395~~  ServiceLine value wil include up to the first ~~8~~ 3 service option values.  <297409-workItem-251821-3>  Update the logic for serviceType=VPLS, for retrieving PVC records:  Adding logic block handling 2,3 PVCsegments  <297409-workItem-251821-4>   * Add conditions in PVC retrieval to limit unwanted/duplicated records (Hina Sardar). * Adding PVC retrieval logic for Ethernet PVCs   <297409-upd5> To address Issue 304559 and 303528  For FR/ATM logic flow, need to handle input shelf in the newly added logic. |
| Archana Gattu | 9/5/2017 | 0.19 | Updates for defect #276859 for PID 297409 |
| Ming Ho | 9/7/2017 | 0.20 | <297409-Def-281399>  For serviceType=VPLS, Include logic to cover services for AVPN, OEW (already coveres), and MIS with the VPLS as dedicated PE for MIS. |
| Archana Gattu | 9/21/2017 | 0.21 | Updates to logic for GMIS for Arista Leaf |
| Archana Gattu | 9/22/2017 | 0.22 | Updates to logic for GMIS for Arista Leaf and Drivnet VPE based on email from INSTAR. |
| Ming Ho | 9/23/2017 | 0.20 | <297409-Def-CR>  The existing slot and port fields in the response are of the PE that is associated to the input component, and has its ports assigned to customer, for the input component+shelf+slot+port.  Add new set of fields including shelf,slot,port for the input device. |
|  |  |  |  |
| Ming Ho | 9/29/2017  10/4/2017 | 0.01 | <297409-CR173072>  Add shelf, slot, port fields to Customer record structure in the response, for the input equipment (PE or Gateway).  These new fields are only populated for service line of AVPN, OEW, and MIS (in logic flow for servive type of FR/ATM, GMIS.  Updated based on review feedback:  Included the new fields for the table change for CBUS;  Included the new fields in Insert statement, when loading the tables. |
| Ming Ho | 11/14/2017 | 0.25 | <297409-ORT-Def>  Issue:  For input VR1 (VPLS) router, the API returns data much less than the data returned from CM Manual process Tool.  Fix:   * Use same logic in VPLS/VR1 router retrieval as that for CM Manual process. * To HLD, this mainly lead to additional logic to exclude segment #3 when covering two segment configuration. The same logic also need to add to PVC retrieval. * Use one block of logic to cover both 2-segment and 3-segment cases, with the input vr1 (VPLS) device on wither second or third segment. |
|  |  |  |  |
| Ming Ho | 3/7/2018  1806 Rel | 0.01 | <301033> US374866   * Re-engineer EDF logic to use the A&AI data sourced from DMaaP instead of the A&AI batch feed. * Support Vyatta uCPE. |

Table of Contents

[Overview 5](#_Toc494877625)

[Problem Statement 5](#_Toc494877626)

[Project Description: 5](#_Toc494877627)

[Design Decisions 5](#_Toc494877628)

[Change Summary <297409-CR173072> 5](#_Toc494877629)

[Change Summary <297100> 5](#_Toc494877630)

[Change summary <297409> 5](#_Toc494877631)

[Change Summary <288315-US681337> 6](#_Toc494877632)

[Change Summary <289116.140768-US636605> 8](#_Toc494877633)

[Flow Diagram 9](#_Toc494877634)

[Change to ARS Acmr5\_cnc\_AllCustomersAffected and Change\_PVC tables: - 297409-CR173072 10](#_Toc494877635)

[Change to GCP\_cnc\_AllCustomersAffected and GCP\_Change\_PVC tables: - 297409-CR173072 11](#_Toc494877636)

[Delete cuatomer records from ARS DB 11](#_Toc494877637)

[InquireCustomerCircuitDetailsByNetworkElement for servicetype=VPLS – 297409-CR173072 12](#_Toc494877638)

[InquireCustomerCircuitDetailsByNetworkElement for servicetype=FR-ATM – 297409-CR173072 14](#_Toc494877639)

[InquireCustomerCircuitDetailsByNetworkElement for servicetype=FR-ATM-RPM – 288315 45](#_Toc494877640)

[InquireCustomerCircuitDetailsByNetworkElement for servicetype=AGN – 288315 57](#_Toc494877641)

[InquireCustomerCircuitDetailsByNetworkElement for servicetype=AGN-RPM – 288315 79](#_Toc494877642)

[InquireCustomerCircuitDetailsByNetworkElement for servicetype=GMIS – 297409-CT173072 86](#_Toc494877643)

[InquireCustomerCircuitDetailsByNetworkElement for servicetype=PVC-ID – 288315 101](#_Toc494877644)

[Tracking Table and Error Handling – <288315-US681337-upd2> 107](#_Toc494877645)

[Common logic – GetUcpeByCircuitID – <289116.140768-US636605> 109](#_Toc494877646)

[Common logic – convert circuit format: CircuitFormatConversion\_ICORE-to-BMP <288315> 111](#_Toc494877647)

[Common logic – Load\_Change\_PVC <288315> 113](#_Toc494877648)

[Common logic – Load\_GCP\_AllCustomersAffected\_sum <288315> 116](#_Toc494877649)

[Alternative Designs 117](#_Toc494877650)

[Assumptions/Risks 117](#_Toc494877651)

[Traceability Matrix 117](#_Toc494877652)

[Pre-Production Disaster Recovery Planning 117](#_Toc494877653)

[Other Plans and References 117](#_Toc494877654)

[Acceptance & Approvals 117](#_Toc494877655)

## Overview

The High Level Design (HLD) describes how an application will implement the architectural concept and meet the requirements allocated to that application. The High Level Design describes the functionality the application will provide, the overall design for providing that functionality and meeting the nonfunctional requirements, and the rationale for choosing that design. The High Level Design also describes how the application will cooperate and interface with other applications to provide an integrated solution that achieves the architectural concept.

The High Level Design covers these topics:

* Problem Statement
* Design Decisions
* Alternative Designs
* Assumptions/Risks
* Traceability Matrix
* Other Plans/References

## Problem Statement

### Project Description:

See SR document.

## Design Decisions

### Change Summary <297409-CR173072>

|  |  |
| --- | --- |
| **Query Name** | **Changes** |
| GetCustInfoQuery for ServiceType = FR/ATM,  GMIS | Add shelf, slot, port fields to Customer record structure in the response, for the input equipment (PE or Gateway).  These new fields are only populated for service line of AVPN, OEW, and MIS (in logic flow for servive type of FR/ATM, GMIS, VPLS)  New fields are added to below tables:  Acmr5\_cnc\_AllCustomersAffected  GCP\_cnc\_AllCustomersAffected |

### Change Summary <297100>

|  |  |
| --- | --- |
| **Query Name** | **Changes** |
| GetCustInfoQuery for ServiceType =  GMIS | **Summary of changes supporting AOTS CM:**   * Enhance Retrival logic for serviceType=GMIS to support new equipments for Arista Leaf and Drivenet VPE * Create data files on Customer records. * BAU part:   Continue load the retrieved Customer records and PVC records into ACR tables ;  Load the retrieved Customer records and PVC records into GCP\_\* replication tables for CBUS  **Code Impacted Applications:**  1.  AOTS CM (20082) >  DMAAP (24167) > CSI GCP (23225) for retrieving impacted asset inventory for CM tickets .  2. GCP  (23225)> ESAR (19906) for supporting changes to CM tables for storing mcn, soc, grc, account hierarchy etc.  3.INSTAR for Inventory data for new Equipments. |

### Change summary <297409>

|  |  |
| --- | --- |
| **Query Name** | **Changes** |
| GetCustInfoQuery for ServiceType =  VPLS | **Summary of changes supporting AOTS CM:**   * Enhance Retrival logic to support new serviceType=VPLS * Increase length of service field from 32 to 80 in both Customer and PVC structure. * EDF will provide multiple service values by concatenating the values with ppe ‘|’ delimiter. * Add new field svlmvlFlag to the PVC records to indicate SVL/MVL * Create data files on Customer records and PVC records for the service types listed above (where the file types are applicable), * EDF shall support a query to pull all customers on all services impacted by a VPLS Planned Maintenance Event. Services Impacted are AVPN, MIS, and OPT-E-WAN. * BAU part:   Continue load the retrieved Customer records and PVC records into ACR tables (include the new fields added for this project);  Load the retrieved Customer records and PVC records into GCP\_\* replication tables for CBUS (include the new fields added for this project);  **Code Impacted Applications:**  1.  AOTS CM (20082) >  DMAAP (24167) > CSI GCP (23225) for retrieving impacted asset inventory for CM tickets .  2.  EM (14148) > CSI GCP (23225) > EDF (23718) for querying by MCN,CUSTID, Org, ACNA/BAN etc. when looking up impacted asset inventory, (including BVOIP indicator), for a CM ticket (e.g. getAfectedCustInfobyChangeID)  3. GCP  (23225)> ESAR (19906) for supporting changes to CM tables for storing mcn, soc, grc, account hierarchy etc.  **Test Impacted Applications:**   1. ICORE (13609) for Service Inventory for Service accounts pertaining to Frame, ATM & IPFR |

### Change Summary <288315-US681337>

|  |  |
| --- | --- |
| **Query Name** | **Changes** |
| GetCustInfoQuery for ServiceType =  FR-ATM  FR-ATM-RPM  AGN  AGN-RPM  GMIS  PVC-ID | **Summary of changes supporting AOTS CM:**   * Creating CSI-API for Asyncronous process to replace existing Web service transactions GetCustInfoQuery (for service Type= FR-ATM, FR-ATM-RPM, AGN, AGN-RPM, GMIS, PVC-ID); * Add new fields to the Customer records and PVC records; * Create data files on Customer records and PVC records for the service types listed above (where the file types are applicable), * BAU part:   Continue load the retrieved Customer records and PVC records into ACR tables (include the new fields added unfrt this project);  Load the retrieved Customer records and PVC records into GCP\_\* replication tables for CBUS (include the new fields added unfrt this project);  **US681337 – GCP EPIC – F18637 – F1.0 Planned/Unplanned Mtc Platform API – updated 7/29, 8/1/2016**  **This HLD covers the requirement for AOTS CM.**  As DBOR of Planned maintt. Service Inventory I want GCP to support the following process flow for enhanced CM ticket notification-  **Process Flow –**   1. AOTS CM will query GCP with CM ticket asset(s) to get impacted asset list for it. 2. GCP will look up inventory (bau) and send 1 file to AOTS with impacted assets and some enhanced info (MCN triplet, account hierarchy and Service names). 3. GCP will store the customer asset info in ESAR tables, formerly known as ARS, tables against the change ID sent by AOTS CM. 4. AOTS CM will make adjustments to the CM ticket and AOTs will send that impacted asset list file to CBUS. 5. AOTS CM will send AOTS CM ticket including impacted asset list to ESAR for reporting. 6. GCP will send the data sets using ESAR CM ticket data as source to EM in response upon a point in time query for customers in EM Profile.   **Inventory Look up (Process steps 1, 2, 3 and 6) –**   1. GCP/EDF will support retrieval of impacted asset inventory (via DMAAP in asynch mode) for CM ticket ed assets from bau data sources (e.g. getCustInfo by assetID, PVCID etc.)           NOTE : There are 5 variations of getcustinfo based on services FR (Frame, Frame-RPM) AGN (AGN, AGN-RPM) and GMIS. The query is supported for PVCID as input.  2.     EDF/GCP will minimally include the following data elements in the response to clients looking up impacted asset inventory for a CM ticket.  1. New Service Line identifier(s) for each customer record  Generic Name : API Service  Line Identifier Value  Frame = FR  ATM=ATM  IPFlex=BVOIP  EVPN=EVPN  IP Frame = IPFR  NB-IPVPN = AVPN  GMIS = GMIS  2. MCN, SOC, GRC, FA/Org, ACNA/BAN etc., including account hierarchy, if applicable for the product/service.  3. Circuit Level 🡪 Aaddress, Zaddress , CLLI, A CLLI , Z CLLI , Provider, Associated Port , Base Channel , Circuit Segment Number, Circuit ID in appropriate format.(e.g. BMP format)  4. PVC Level 🡪 Aaddress , Zaddress , A CLLI, Z CLLI, A CIR, Z CIR, Circuit ID in appropriate format. (e.g. BMP format), VPNID  5. Managed or not managed indicator.   1. BVOIP presence indicator, if applicable.   3.     EDF will allow query by MCN, CUSTID, FA/Org, etc. when looking up impacted asset inventory (including BVOIP indicator) for a CM ticket (eg. getAffectedCustInfobyChangeID,  getAffectedPVCInfoByChangeId, listContactByOrgCdAndAccountType).  4.     EDF will provide mcn and mcn triplet in response to clients looking up impacted asset inventory for a CM ticket (eg. getAffectedCustInfobyChangeID, getAffectedPVCInfoByChangeId, listContactByOrgCdAndAccountType etc.)  5.     EDF will echo back ticket #, and request #, and other relevant info that will be used by clients like AOTS-CM to match the response to the proper request.  NOTE: The enhancements mentioned above will be available/applicable for in-scope service assets only (e.g. FR, ATM). The enhancements will not be available/applicable for assets of out of scope service lines (e.g. PL) and hence will be optional for those.  **Assumptions:**  1. Port ckts are not customer ckts and will not be included in the impacted inventory data sets. (bau)  2. BD Company ID is not needed by CBUS on CM ticket inventory.  3. AOTS-CM query response will be in asynch mode for all inputs (including PVCID)  4. Account hierarchy to provided in Dmaap file to Aots will be established in respective service inventory DBORs by service delivery process for those products / services.  **5. EM will not query GCP with MCN SO BG for out of scope services.**  **6.**The enhancements mentioned in #2 above will be available/applicable for in-scope service assets only (e.g. FR, ATM). The enhancements will not be available/applicable for assets of out of scope service lines (e.g. PL) and hence will be optional for those.  **7. Services impacted for this project are loaded in GDB already. (Yun/ Ming to confirm)**  **Dependency:**  1. GCP production fix for mapping the correct OrgAccount instead of GemsCompanyID in FA/Org response to AOTs and EM will go in before or along with this project.  2. Site Level FA/Org and CER name must be populated in PVC level record for all services. There is a prod defect observed for EVPN.  **Acceptance Criteria:**   1. AOT CM is able to retrieve impacted asset inventory for CM tickets for AVPN, EVPN, GMIS, Frame/ATM and IPFLex assets. (Network layer – Frame, AGN, GMIS) services. 2. The data elements mentioned in Inventory Look up #2 are provided in the response to AOTs for a CM ticket. 3. Latest data set of impacted asset inventory is available to the clients for 18 months. 4. AOTs receives the ticket #, and request #, and other relevant info that will be used to match the response to the proper request for asynchronous responses via DMAAP. 5. EM is able to query by MCN, Org, CUSTID, Org etc. when looking up impacted asset inventory for a CM ticket (eg. getAffectedCustInfobyChangeID, getAffectedPVCInfoByChangeId, listContactByOrgCdAndAccountType etc.).   See volume info in US680427.  **Code Impacted Applications:**  1.  AOTS CM (20082) >  DMAAP (24167) > CSI GCP (23225) for retrieving impacted asset inventory for CM tickets including elements mentioned in #2 in Inventory Look Up Section.  2.  EM (14148) > CSI GCP (23225) > EDF (23718) for querying by MCN,CUSTID, Org, ACNA/BAN etc. when looking up impacted asset inventory, (including BVOIP indicator), for a CM ticket (e.g. getAfectedCustInfobyChangeID, getAffectedPVCInfoByChangeId)  3. GCP  (23225)> ESAR (19906) for supporting changes to CM tables for storing mcn, soc, grc, account hierarchy etc.  **Test Impacted Applications:**   1. ICORE (13609) for Service Inventory for Service accounts pertaining to Frame, ATM & IPFR 2. INSTAR (13672), NC3 (13200) for Service Inventory for Service accounts pertaining to  AVPN, EVPN, GMIS 3. GCP (14724) for Service Inventory for Service accounts pertaining to IPFLex assets 4. EM (14148) >EDF  for retrieving CustID for a given FA/Org. |

### Change Summary <289116.140768-US636605>

|  |  |
| --- | --- |
| **Query Name** | **Changes** |
| GetCustInfoQuery for ServiceType =  FR-ATM  FR-ATM-RPM  AGN  AGN-RPM  GMIS | US636605 – CR140768 – US GCP-AOTS-CM supporting uCPE for change management  **As a** data provider responsible **for** providing GCP service assurance clients Gamma related data, **I want** to enhance existing Query for AOTS-CM **so that** AOTS-CM can retrieve uCPE data for change management need.          The objective is to provide uCPE (host hardware and VNFs) to AOTS-CM when a PE is chosen as the input to existing GCP- AOTS-CM query.  For GCP-AOTS-CM interface, we will enhance  existing Query in CR 140768 ---       getCustInfoQuery  (may support multiple services)        The final query list (based on different services) will be determined in the AID phase after discussion with AOTS-CM.    **Dependency:** The required data are available in the source systems --- A&AI, ~~GDB, CANOPI~~ |

### Flow Diagram

Clients Request to CSI-GCP for data retrieval:

* CSI-GCP sends back acknowledgement to the client with File names which include unique tracking ID;
* CSI-GCP loads the request parameters, along with the unique tracking ID, into a Request table;

Insert Request parameters into Request table

CSI-Adapter

CSI-GCP

Clients

SOAP/HTTPS

ETL job monitors the Request table.

* When new request inserted, ETL job processes the request and generates the output files;
* If error/exception occurs in the process, ETL job add standard exception messages to the file header;
* PUSH the completed files (in XML format) into DataRouter/Dmaap Platform;
* DataRouter sends the completed files to Client application or Client pull the file from DataRouter.

Client

DataRouter

ETL Job monitors Request table.

Process request, generate files.

<297409>

|  |  |
| --- | --- |
| **Change Area** | **Comments and Logic** |
| Change to ARS tables:  Acmr5\_cnc\_AllCustomerAffected  Change\_PVC | **Change to ARS tables holding Customer and PVC records (DBA impact):**  **Make changes to the Acmr5\_cnc\_AllCustomersAffected table:**  Change field length:  Service\_line – from varchar2(32) to varchar2(120)  **Make changes to the Change\_PVC table:**  Add following field:  SVL\_MVL\_FLAG- Varchar2(3)-Optional  Change field length:  a\_service -- from varchar2(32) to varchar2(120)  z\_service -- from varchar2(32) to varchar2(120)  a\_vpn\_id—from varchar2(4) to varchar2(12)  z\_vpn\_id—from varchar2(4) to varchar2(12) |

Change to ARS Acmr5\_cnc\_AllCustomersAffected and Change\_PVC tables: - 297409-CR173072

|  |  |
| --- | --- |
| **Change Area** | **Comments and Logic** |
| Change to ARS tables:  Acmr5\_cnc\_AllCustomerAffected  Change\_PVC | **Change to ARS tables holding Customer and PVC records (DBA impact):**  **Make changes to the Acmr5\_cnc\_AllCustomersAffected table:**  Adding the following reportable fields:   * CircuitId\_Bmp\_Format - varchar2(50) * Parent\_org\_group - varchar2(25) * Managed\_Indicator - varchar2(5) * Bvoip\_Indicator - varchar2(5)   Change field length:  Customer\_ip\_address – From varchar2(15) to varchar2(45)  Ip\_address – From varchar2(20) to varchar2(45)  Service\_line – from varchar2(16) to varchar2(32)  <297409-CR173072>  Add following fields  Request\_equip\_shelf varchar2(10) Optional  Request\_equip\_slot varchar2(5) Optional  Request\_equip\_port varchar2(5) Optional  **Make changes to the Change\_PVC table:**  Adding the following reportable fields:   * A\_CircuitId\_Bmp\_Format - varchar2(50) * A\_Parent\_org\_group - varchar2(25) * A\_Managed\_Indicator - varchar2(5) * A\_vpn\_id – number(4) <288315-US681337-upd3> * Z\_CircuitId\_Bmp\_Format - varchar2(50) * Z\_Parent\_org\_group - varchar2(25) * Z\_Managed\_Indicator - varchar2(5) * Z\_vpn\_id – number(4) <288315-US681337-upd3>   Change field length:  A\_cust\_ip\_address – From varchar2(15) to varchar2(45)  Z\_cust\_ip\_address – From varchar2(15) to varchar2(45) |

Change to GCP\_cnc\_AllCustomersAffected and GCP\_Change\_PVC tables: - 297409-CR173072

|  |  |
| --- | --- |
| **Change Area** | **Comments and Logic** |
| Change to ARS tables:  GCP\_cnc\_AllCustomerAffected  GCP\_Change\_PVC | **Change to GCP tables holding Customer and PVC records (DBA impact):**  **Make changes to the GCP\_cnc\_AllCustomersAffected table:**  Adding the following reportable fields:   * CircuitId\_Bmp\_Format - varchar2(50) * Parent\_org\_group - varchar2(25) * Managed\_Indicator - varchar2(5) * Bvoip\_Indicator - varchar2(5)   Change field length:  Customer\_ip\_address – From varchar2(15) to varchar2(45)  Ip\_address – From varchar2(20) to varchar2(45)  Service\_line – from varchar2(16) to varchar2(32)  <297409-CR173072>  Add following fields  Request\_equip\_shelf varchar2(10) Optional  Request\_equip\_slot varchar2(5) Optional  Request\_equip\_port varchar2(5) Optional  **Make changes to the GCP\_Change\_PVC table:**  Adding the following reportable fields:   * A\_CircuitId\_Bmp\_Format - varchar2(50) * A\_Parent\_org\_group - varchar2(25) * A\_Managed\_Indicator - varchar2(5) * A\_vpn\_id – number(4) <288315-US681337-upd3> * Z\_CircuitId\_Bmp\_Format - varchar2(50) * Z\_Parent\_org\_group - varchar2(25) * Z\_Managed\_Indicator - varchar2(5) * Z\_vpn\_id – number(4) <288315-US681337-upd3>   Change field length:  A\_cust\_ip\_address – From varchar2(15) to varchar2(45)  Z\_cust\_ip\_address – From varchar2(15) to varchar2(45) |

Delete cuatomer records from ARS DB

CM passing ChangeID and TransID to GCP, with ServiceType = “DeleteCust” in getCustInfoQuery(), CGP will then delete customer records matching the input ChangeID and TransID, and its related PVC records if exists.

|  |  |
| --- | --- |
| **Change Area** | **Comments and Logic** |
| Delete records from ARS table | **<182639b AOTS/CM-GCP Req. #140>: Refer to PB932 implementation for getCustInfoQuery:**  **This step is only executed under the conditions below:**   * **ServiceType = DeleteCust** * **Both ChangeId and CmdcTransId are in the input.**   Set Input\_Change ID = <Input Change ID>  Set Input\_TransID = <Input CmdcTransID>  **Delete all match records from table Acmr5\_cnc\_AllCustomersAffected:**  Delete from Acmr5\_cnc\_AllCustomerAffected  Where Acmr5\_cnc\_AllCustomerAffected.Change\_Request\_ID = Input\_Change\_ID  And Acmr5\_cnc\_AllCustomerAffected.cmdc\_transid = Input\_transid  **Delete all match records from table Change\_pvc:**  Delete from Change\_pvc  Where Change\_pvc.Change\_Request\_ID = Input\_Change\_ID  And Change\_pvc.cmdc\_transid = Input\_transid  <252833.83800>  **Delete all match records from table GCP\_cnc\_AllCustomer**s**Affected, GCP\_Change\_Pvc, and GCP\_AllCustomer**s**Affected\_sum:**  Delete from Gcp\_cnc\_AllCustomersAffected  Where Change\_Request\_ID = <Input\_Change\_ID>  And cmdc\_transid = Input\_transid  Delete from Gcp\_Change\_pvc  Where Change\_Request\_ID = <Input\_Change\_ID>  And cmdc\_transid = Input\_transid  Delete GCP\_AllCustomersAffected\_sum table  Where Change\_ ID = <Input\_Change\_ID>  </252833.83800> |

### InquireCustomerCircuitDetailsByNetworkElement for servicetype=VPLS – 297409-CR173072

|  |  |
| --- | --- |
| **Query Name** | getCustInfoQuery for servicetype=VPLS  API: InquireCustomerCircuitDetailsByNetworkElement |
| **PIDs** | 297409, CR173072 |
| **Service** | AVPN, MIS, OEW |
| **Data Source** | ICORE, SIDBOR, INSTAR |
| **WSDL File** | NA |
| **Client App** | AOTS CM |
| **Purpose, Usage** | Support VPLS PE for AVPN, MIS, OEW services |

**Query Change Summary**

|  |  |
| --- | --- |
| **Project /Ticket ID** | **Change Summary / Notes** |
| 297409 Oct 2017 | Support AOTS CM ticket notification  This enhancement includes:  Support new ServiceType=VPLS from AOTS as part of input.  Add new field svl\_mvl\_flag to PVC records  Change service field length from 32 to 120 on both Customer and PVC records to support multiple services return.  <297409-workItem-251821>  Address issue of VPLS logic not retrieving the AVPN and MIS PVCs for vr1 routers  <297409-workItem-251821-2>  Update the logic for serviceType=VPLS, for retrieving customer records:  Stating executing both existing and new logic;  ServiceLine value wil include up to the first 3 service option values.  <297409-workItem-251821-3>  Update the logic for serviceType=VPLS, for retrieving PVC records:  Adding logic block handling 2,3 PVCsegments  <297409-workItem-251821-4>   * Add conditions in PVC retrieval to limit unwanted/duplicated records (Hina Sardar). * Adding PVC retrieval logic for Ethernet PVCs   <297409-workitem-281399>  For serviceType=VPLS, Include logic to cover services for AVPN, OEW (already coveres), and MIS with the VPLS as dedicated PE for MIS. |
| 297409 CR173072  Dec’2017 | <297409-CR173072>   * Add shelf, slot, port fields to Customer record structure in the response, for the input equipment (PE or Gateway). * These new fields are only populated for service line of AVPN, OEW, and MIS (in logic flow for servive type of FR/ATM, GMIS, VPLS) * Customer record retrieval logic is updated to include new shelf/slot/port fields in Select clause, and add itional consition to support input SubSlot. * PVC record retrieval is updated with additional consitions to support input SubSlot. |
| 297409 ORT  Nov’2017 | <297409-ORT-Def>  Issue:  For input VR1 (VPLS) router, the API returns data much less than the data returned from CM Manual process Tool.  Fix:   * Use same logic in VPLS/VR1 router retrieval as that for CM Manual process. * To HLD, this mainly lead to additional logic to exclude segment #3 when covering two segment configuration. The same logic also need to add to PVC retrieval. * Use one block of logic to cover both 2-segment and 3-segment cases, with the input vr1 (VPLS) device on wither second or third segment. |
| 301033  June’2018 | <301033> US374866   * Re-engineer EDF logic to use the A&AI data sourced from DMaaP instead of the A&AI batch feed. * Support Vyatta uCPE. |

<297409-workitem-281399>

|  |  |
| --- | --- |
|  | **Comments and Logic** |
| STEP-VPLS-10 | For ServiceType = ‘VPLS’  The following services are supported:   * MIS * AVPN * OPT-E-WAN   For the input VPLS equipment and other parameters (Shelf/slot/port), executing all the following logic steps:   1. Execute Steps in Logic Flow for ServiceType = FR-ATM,  * Create circuit list for the retrieved Access circuits **<ACC-CKT-List>;** * Keep the **CustomerRecordCnt** value (Step 5b in FR-ATM flow), which will be used in the next step. * This is to support AVPN, OEW, and MIX with the VPLS PE as gateway.  1. Execute Steps in Logic Flow for ServiceType = GMIS,  * Excluding the access circuits in **<ACC-CKT-List>;** * When inserting retrieved records into tables, use the above **CustomerRecordCnt** in setting the Customer\_Record\_Selector * This is support VPLS as dedicated PE for MIS, and 40, 100 BG services   Then   * Create data file using the data retrieved from all the above steps; * Insert all the data records retrieved from the above steps into the two (Customer records and PVC records) tables;   **Note:** Per input from Dev/ST team (10/3/2017)   * For serviceType=VPLS, a separate flow of SQLs are created by Dev team to support this service, based on new logic added to FR\_ATM and GMIS. * The update to logic in serviceType=FR\_ATM to support input VPLS-PE is included in the SQLs for VPLS PE, but not in the SQLs for FR\_ATM; * For CR173072, logic changes are implemented in the SQLs for VPLE, not for FR\_ATM/GMIS. * For CR173072, only change to serviceType=FR\_ATM and GMIS is to add new fields in Select clause, and in the Insert statement loading to customerAffected tables. |

### InquireCustomerCircuitDetailsByNetworkElement for servicetype=FR-ATM – 297409-CR173072

**Formally known as “GetCustInfoQuery for servicetype=FR-ATM”**

|  |  |
| --- | --- |
| **Query Name** | getCustInfoQuery for servicetype=FR-ATM  ~~ETL~~ EDB/WS backend data retrieval  API: InquireCustomerCircuitDetailsByNetworkElement |
| **PIDs** | PD064, PD253, PF951, 289116.140768, 288315, 297409, 297409 CR173072, 297100 |
| **Service** | AVPN |
| **Data Source** | ICORE, SIDBOR, CCI |
| **WSDL File** | CmdcWebServicesContract.wsdl |
| **Client App** | AOTS CM |
| **Purpose, Usage** | FR-ATM Customer Inventory Retrieval.  This use case details the customer inventory request from AOTS for Frame ATM data. FR-ATM query is phase 1 and being enhanced in phase 2. GCP will retrieve data from GCP ICORE database. Satisfying P8130 Customer Notification Phase 2 BRD, requirement BRD\_Command\_and\_Control\_Ph2\_110.  Input data combination can be: EQUIP\_NAME; EQUIP\_NAME + SLOT; EQUIP\_NAME + SLOT + PORT. |

**Query Change Summary**

|  |  |
| --- | --- |
| **Project /Ticket ID** | **Change Summary / Notes** |
| PB932 | Lan Tran-Vu:  Initial Issue |
| 10/31/05  Version 1.4 | Lan Tran-Vu:  Update to add Phase 3 requirement |
| MR 22611  12/16/05  Version 1.5 | Lan Tran-Vu:  Update for MR 22611 – eliminate trailing blanks. Change in step 4. |
| P8635.7  Version 1.7 | Add CR P8635.7 change AOTS return tag ACCESS\_ID to CIRCUIT\_ID. Document change only. |
| P95872.0 | Add P9587 change in step 3.b |
| P8A30  Version 3.0 | Add P8A30 requirement. Change in step 3.b |
| PA569  Version 3.1 | Add PA569 requirement. Add RESTR\_IND. Change in step 3.b |
| 10/11/07 | Lan Tran-Vu:  Take out CCI data. Change in step 3.c |
| P8130  2/27/2008 | Gunjan Gupta,Lan Tran-Vu:  Adding PVC data |
| 289116.140768  July’2016 | US636605 – CR140768 – US GCP-AOTS-CM supporting uCPE for change management  Enhance for AOTS-CM to retrieve uCPE data when a PE is chosen as the input.  Dependency: The required data are available in the source systems --- A&AI, ~~GDB, CANOPI~~ |
| 288315  Oct’2016 | US681337 – Support AOTS CM ticket notification.  This enhancement includes  Add new fields to both Customer records and PVC records;  Change the process to Async, and the data retrieval logic is used by ETL to generate files. |
| 297409 Oct 2017 | Support AOTS CM ticket notification  This enhancement includes:  Support new ServiceType=VPLS from AOTS as part of input.  Add new field svl\_mvl\_flag to PVC records  Change service field length from 32 to 120 on both Customer and PVC records to support multiple services return.  <297409-workItem-251821>  Address issue of VPLS logic not retrieving the AVPN and MIS PVCs for vr1 routers  <297409-workItem-251821-2>  Update the logic for serviceType=VPLS, for retrieving customer records:  Stating executing both existing and new logic;  ~~Excluding cust\_id=395~~  ServiceLine value wil include up to the first ~~8~~ 3 service option values.  <297409-workItem-251821-3>  Update the logic for serviceType=VPLS, for retrieving PVC records:  Adding logic block handling 2,3 PVCsegments  <297409-workItem-251821-4>   * Add conditions in PVC retrieval to limit unwanted/duplicated records (Hina Sardar). * Adding PVC retrieval logic for Ethernet PVCs |
| 297409 CR173072  Dec’2017 | <297409-CR173072>   * Add shelf, slot, port fields to Customer record structure in the response, for the input equipment (PE or Gateway). * These new fields are only populated for service line of AVPN, OEW, and MIS (in logic flow for servive type of FR/ATM, GMIS, VPLS) * Customer record retrieval logic is updated to include new shelf/slot/port fields in Select clause, and add itional consition to support input SubSlot. * PVC record retrieval is updated with additional consitions to support input SubSlot. |
| 297409 ORT  Nov’2017 | <297409-ORT-Def>  Issue:  For input VR1 (VPLS) router, the API returns data much less than the data returned from CM Manual process Tool.  Fix:   * Use same logic in VPLS/VR1 router retrieval as that for CM Manual process. * To HLD, this mainly lead to additional logic to exclude segment #3 when covering two segment configuration. The same logic also need to add to PVC retrieval. * Use one block of logic to cover both 2-segment and 3-segment cases, with the input vr1 (VPLS) device on wither second or third segment. |
| 301033  June’2018 | US374866   * Re-engineer EDF logic to use the A&AI data sourced from DMaaP instead of the A&AI batch feed. * Support Vyatta uCPE. |

|  |  |
| --- | --- |
| USE CASE GENERAL INFORMATION | |
| Use Case Name | FR-ATM Customer Inventory Retrieval |
| Use Case ID | UC-GCP-PB932-002 |
| Description | This use case details the customer inventory request from AOTS for Frame ATM data. FR-ATM query is phase 1 and being enhanced in phase 2. GCP(DBOR) will retrieve data from GCP(DBOR) ICORE database. Satisfying P8130 Customer Notification Phase 2 BRD, requirement BRD\_Command\_and\_Control\_Ph2\_110.  Input data combination can be: EQUIP\_NAME; EQUIP\_NAME + SLOT; EQUIP\_NAME + SLOT + PORT.  Data source: ICORE, SIDBOR |
| Responsible Analyst | Lan TranVu |
| Type of Execution | WebService and SQL to Oracle db. |

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE REVISION LOG | | | |
| Reason for Revision: | Adding PVC data | Release Number: | 2.0 |
| Author: | Gunjan Gupta,Lan Tran-Vu | Revised: | 02/27/08 |
| Reason for Revision: | Take out CCI data. Change in step 3.c | Release Number: | 5.3 |
| Author: | Lan Tran-Vu | Revised: | 10/11/07 |
| Reason for Revision: | Add PA569 requirement. Add RESTR\_IND. Change in step 3.b | Release Number: | 3.1 |
| Reason for Revision: | Add P8A30 requirement. Change in step 3.b | Release Number: | 3.0 |
| Reason for Revision: | Add P9587 change in step 3.b | Release Number: | 2.0 |
| Reason for Revision: | Add CR P8635.7 change AOTS return tag ACCESS\_ID to CIRCUIT\_ID. Document change only. | Release Number: | 1.7 |
| reason for Revision: | Update for MR 22611 – eliminate trailing blanks. Change in step 4.  Update for MR 23893 – add SLOT to return data | Release Number: | 1.5 |
| Author: | Lan Tran-Vu | Revised: | 12/16/05 |
| Reason for Revision: | Update to add Phase 3 requirement | Release Number: | 1.4 |
| Author: | Lan Tran-Vu | Revised: | 10/31/05 |
| Reason for Revision: | Initial Issue | Release Number: |  |
| Author: | Lan Tran-Vu | Revised: |  |

<297409>

EDF will support impacted customer retrieval at the

1. Site level,
2. PVC level

similar to serviceType=FR-ATM change tickets.

serviceType=VPLS will return the same number of fields that are return today on both Customer and PVC records as serviceType=FR\_ATM.One new field which will be added on PVC record to identify SVL/MVL (single VLAN/multi VLAN).This new field will be supported for both serviceType VPLS and FR\_ATM

Service field length on both the records will be increased to support multiple values for service.EDF will concatenate the different values using pipe’|’ delimiter.

</297409>

Return Structures:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| FR-ATM Query Response | | | | | |
| Field | R/O/C | Data Type | Occur | Notes | ICORE DB  Table Name.column Name |
| Header | R | N/A | 1 | Begin Tag |  |
| Status | R | AN (3) | 1 | The status of the transaction.  Valid Values:  ‘0’ for Success  ‘1’ for Warning  2 for Backend system is down |  |
| Header | R | N/A | 1 | End Tag |  |
| Body | R | N/A | 1:N | Begin Tag |  |
| Customer Information Response | R | N/A | 0:N | Begin Tag |  |
| CUSTOMER\_RECORD\_SELECTOR | R | Int | 1 | No trailing blanks |  |
| Recordcount | R | Int | 1 |  |  |
| Customer ID | R | Int | 1 | No trailing blanks | CUSTOMER.CUST\_ID |
| Customer Name | R | A24 | 1 | No trailing blanks | CUSTOMER.CUST\_NAME |
| Customer MCN | ~~R~~ O | A9 | 1 | No trailing blanks | CUSTOMER.CUST\_MCN |
| Location ID | ~~R~~ O | Int | 1 | No trailing blanks | PREMISE.LOC\_ID |
| Site Number | ~~R~~ O | Int | 1 | No trailing blanks | SITE.SITE\_NUM |
| Site Protocol | ~~R~~ O | A4 | 1 | No trailing blanks | SITE.PROTOCOL |
| Site TA | ~~R~~ O | A12 | 1 | No trailing blanks | SITE.TA\_NAME |
| Global DLCI | ~~R~~ O | Int | 1 | No trailing blanks | SITE.SITE\_GLBL\_DLCI |
| CircuitId | O | A50 | 1 |  | CUST\_ACCESS.ACC\_CKT  Note: This field has value directly retrieved from source DB, without format conversion.  seems BMP/normalized format being returned currently;  Check whether it’s converted already by GCP or ICORE.  Then determined whether new field is needed |
| circuitIdBmpFormat | O | A50 | 1 |  | <288315> Customer access circuit in BMP (normalized) format.  Deived value.  Note:  No conversion for International circuit. |
| TypeOfService | R | A7 | 1 |  | Derived |
| Shelf | O | A10 | 1 | No trailing blanks | PORT\_ASGMT.SHELF |
| Port | O | Varchar2(20) | 1 | No trailing blanks | PORT\_ASGMT.PORT |
| Slot | O | Varchar2(20) | 1 | No trailing blanks | PORT\_ASGMT.SLOT |
| Address | ~~R~~ O | A50 | 1 | No trailing blanks | PREMISE.PREM\_ADDRESS |
| City | ~~R~~ O | A25 | 1 | No trailing blanks | PREMISE.PREM\_CITY |
| State | ~~R~~ O | A2 | 1 | No trailing blanks | PREMISE.PREM\_STATE |
| Country | ~~R~~ O | A20 | 1 | No trailing blanks | PREMIE.PREM\_COUNTRY |
| CLLI | O | A11 | 1 | No trailing blanks | SITE.CLLI |
| Port Speed | ~~R~~ O | Int | 1 | No trailing blanks | PORT\_ASGMT.PORT\_SPEED |
| CER Name | O | A50 | 1 | No trailing blanks | SIDBOR DB  FWTOPOLOGY.LOCAL\_ROUTER |
| managedIndicator | O | A5 | 1 | No trailing blanks | <288315> Values: Y, N, or null if no CPE exist |
| bvoipIndicator | O | A5 | 1 | No trailing blanks | <288315> Values: Y, N, or null |
| requestEquipShelf | O | Varchar2(10) | 1 | No trailing blanks | <297409-CR173072> Shelf for input device (which can be PE or gateway) related to this customer record. |
| requestEquipSlot | O | Varchar2(5) | 1 | No trailing blanks | <297409-CR173072> Slot for input device (which can be PE or gateway) related to this customer record. |
| requestEquipPort | O | Varchar2(5) | 1 | No trailing blanks | <297409-CR173072> Port for input device (which can be PE or gateway) related to this customer record. |
| OrgGroup | O | A11 | 1 | No trailing blanks | SIDBOR DB  ~~ASSET.SRC\_ORG\_ID~~  ASSET.ORG\_CD |
| parentOrgGroup | O | A25 | 1 | No trailing blanks | <288315>  As of now after discussion, there is no current plan to use by CM for this project.  This may be used by EM. |
| ManageOrg | O | A11 | 1 | No trailing blanks | SIDBOR DB  ASSET.MANAGING\_ORG\_CD |
| ActiveOrg | O | A12 | 1 | No trailing blanks | SIDBOR DB  ASSET.ACTIVE\_ORG\_CD |
| FunctionalArea | O | A12 | 1 | No trailing blanks | SIDBOR DB  ASSET.functional\_area |
| CCI | O | A18 | 1 | Blanks | CCI DB.  DNB.DUNS\_NB |
| Parent Company Name | O | A90 |  | Blanks | CCI DB  DNB.PRNTHQ\_BIZ\_NM |
| **<CpeDetailsList>** | O | Complex  Array | 0:1 |  | <289116.140768-US636605>   * This list/array is to hold CPE records for this customer (customer access circuit). * For 1607 release, only uCPE data is supported. * See below for structure definition. |
| Customer Information Response | R | N/A |  | End Tag |  |
| Body | R | N/A | 1 | EndTag |  |

**CpeDetailsList Structure: Complex/Array <289116.140768-US636605>**

Occurrence 0:1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **GCP WSDL Name** | **Data Type, Length** | R/O/C | AOTS Field Mapping  GUI Label  SIZE  **DB Name** | Notes |
| <CpeDetails> | Complex | R | Occurrence 1:20 |  |
| deviceName | Varchar2(50) | R | Device Name  Varchar2(20) | This can be Host Name, PTNii Name |
| deviceClli | Varchar2(50) | O | Device CLLI  Varchar2(15) | Device CLLI code |
| deviceModel | Varchar2(50) | O | Device Model  Varchar2(50) |  |
| equipmentVendor | Varchar2(50) | O | Vendor  Varchar2(50) |  |
| managedBy | Varchar2(50) | O | Managed By  Varchar2(50) | Specify the device is managed by AT&T or customer |
| deviceType | Varchar2(50) | O | Device Type  Varchar2(50) | Sample values:  UCPE-JUNIPER UCPE  VNF-FW  VNF-RT  VNF-SW  VNF-ZZ  RR-MRS  RR-AVPM-M  ~~CPE – to be used in future when other CPEs are supported.~~ |
| </CpeDetails> |  |  |  |  |

|  |  |
| --- | --- |
|  | **Comments and Logic** |
| 297409  GetCustInfoQuery for ServiceType =  VPLS | EDF will send below mentioned new field along with the existing fields that are sent to AOTS CM today for serviceType=VPLS and FR-ATM  Logic to Retrieve new field SVL\_MVL\_Indicator(from ICORE)  EDF will use same logic for PVC records as used for FR-ATM today.  PVC.PVC\_ID will be used to map to pvc\_extension.PVC\_ID.  Select mvl from pvc\_extension pe,pvc p  Where  Pe.pvc\_id=p.pvc\_id(this will be pvc id from PVC Records)  If value is Y, then set SVL\_MVL\_Flag=’MVL’  If value is null/blank, set SVL\_MVL\_Flag=’SVL’  Logic for service\_line for Servicetype=VPLS  Defect #276859: The only services supported for servicType=VPLS are  MIS  AVPN  OPT-E-WAN  EDF will return the service\_line data only one of these services mentioned above are associated at customer or PVC level for serviceType=VPLS  EDF will use below mention logic to derive the service\_line field for serviceType=VPLS for Customer Record  Select UNIQUE service\_option.serv\_opt  From  service\_asgmt,  service\_option  Where service\_asgmt.site\_id = <site\_id from Customer Record logic>  And service\_asgmt.cust\_id = <cust\_id from customer Record logic>  And service\_asgmt.serv\_opt\_id = service\_option.serv\_opt\_id  If serv\_opt=’NB-IPVPN’, then set service\_line =’AVPN’  If serv\_opt =’ETHERNET, then set service\_line=’MIS’  If serv\_opt=’OPT-E-WAN’, then set service\_line=’OPT-E-WAN’  Note.In existing Customer record logic for FR-ATM,Enhance the query to extract the site\_id use this field in above logic to extract the service  <297409-workItem-251821-2>  Concadinate up to the first ~~8~~ 3 serv\_opt values and populate to the service\_line field.  EDF will use below mention logic to derive the service\_line field for serviceType=VPLS for PVC Record  Defect 276859:  PVC level can have only one service associated to it.  The only valid values for service line for serviceType=VPLS are MIS,OPT-E-WAN and AVPN  Use PVC\_ID from the PVC records logic to retrive serv\_opt field from ICORE.service\_option table  select  so.serv\_opt  from  pvc p,  site s,  service\_asgmt sa,  service\_option so  where  (pvc.pvc\_lsite\_id=s.site\_id or pvc.pvc\_rsite\_id=s.site\_id)--< this will be pvc\_lsite\_id or pvc\_rsite\_id from PVC record) and  sa.site\_id=s.site\_id  and  sa.serv\_opt\_id=so.serv\_opt\_id  If serv\_opt=’NB-IPVPN’, then set service\_line =’AVPN’  If serv\_opt =’ETHERNET, then set service\_line=’MIS’  If serv\_opt=’OPT-E-WAN’, then set service\_line=’OPT-E-WAN’  Note.In existing PVC record logic fro FR-ATM,Enhance the query to extract the pvc\_lsite\_id and pvc\_rsite\_id and use those field in above logic to extract the service  <297409-workItem-251821-2>  Concadinate up to the first ~~8~~ 3 serv\_opt values and populate to the service\_line field at PVC level. |

|  |  |  |  |
| --- | --- | --- | --- |
| IDEAL COURSE | | | |
| # | Step Description | | |
| 1. | Use Case begins | | |
| 2. | DBOR received query getCustInfoQuery request from AOTS with SERVICE TYPE = ‘FR-ATM’  For each record in <InputEquipmentList> <288315> Support list of input equipment names  ~~Set Input\_EquipName = AOTS.EQUIP\_NAME~~  ~~Set Input\_Change Id = AOTS. Change Id~~  ~~Set Input\_ CmdcTransID = AOTS. Cmdc Trans ID~~  Set Input\_EquipName = Input.equipmentName  Set Input\_ChangeId = Input.Change Id  Set Input\_ CmdcTransID = Input.CmdcTransactionID  If Input SLOT and/or Input PORT are populated  Set Input\_Slot = Input.SLOT  Set Input\_Port = Input.PORT  Set Input\_Shelf = Input.SHELF  Set Input\_Sub SLot = Input.SUBSLOT  Else  End If | | |
| 3.a | Retrieve Customer records for input parameters, from ICORE (GCP replication) tables:  <288315> Add new fields  circuitIdBmpFormat  bvoipIndicator  **Note <288315>:**   * Access circuit in ICORE format has ‘.’ As delimiter to separate fields in the circuitID, as shown below. * The logic below checks these format and determine whether it’s ICORE format that can be convered to BMP format. For circuits other than these format will not get converted.   **DHEC.416129.200.ATI --‘.’ Delimiter starts at 5th position**  **23.HFFM.000433..SUV --‘.’ Delimiter starts at 3rd position**  **<297409-workItem-251821-2>**  **Execute both Step 3.a-10 and 3.a-20 for Customer records retrieval**  **Step 3.a-10: Retrieve Customer records for the input router**  Select  Customer.cust\_id AS customerID,  Customer.cust\_name AS customerName,  Customer.cust\_mcn AS customerMcn,  Site.site\_num,  Site.protocol,  Site.ta\_name,  Site.site\_glbl\_dlci,  Cust\_access.acc\_ckt AS circuitID,  Decode (Cust\_access.acc\_ckt,  Instr (Cust\_access.acc\_ckt, ‘.’, 1, 1) = 3, ‘Y’,  Instr (Cust\_access.acc\_ckt, ‘.’, 1, 1) = 5, ‘Y’,  ‘N’ ) cktConversionFlag, <288315> Used to mark whether the circuit format can be converted.  Port\_asgmt.port,  Port\_asgmt.slot,  Port\_asgmt.Shelf—added for defect id **230097 for PID 297409**  Port\_asgmt.port\_speed,  Site.clli,  Premise.prem\_address,  Premise.prem\_city,  Premise.prem\_state,  Premise.prem\_country,  Port\_asgmt.shelf AS requestEquipShelf, <297409-CR173072> input router is PE  Port\_asgmt.slot AS requestEquipSlot,  Port\_asgmt.port AS requestEquipPort  From  EQUIPMENT,  CUSTOMER,  CUST\_ACCESS,  SITE,  PREMISE, ~~PVC~~, ~~IPFR~~,  PORT\_ASGMT,  EQUIP\_MODEL,  SLOT\_MAP  Where  EQUIPMENT.equip\_id = PORT\_ASGMT.equip\_id  And PORT\_ASGMT.site\_id = CUST\_ACCESS.site\_id  And CUSTOMER.cust\_id = CUST\_ACCESS.cust\_id  And PORT\_ASGMT.site\_id = SITE.site\_id  And SITE.prem\_loc\_id = PREMISE.loc\_id  ~~And PVC.pvc\_lsite\_id = SITE.site\_id~~  ~~And PVC.pvc\_id = IPFR.pvc\_id~~  ~~And PVC.pvc\_rsite\_id = SITE.site\_id~~  And SITE.loc\_id = PREMISE.loc\_id  And EQUIPMENT.equip\_name = Input\_EquipName  And PORT\_ASGMT.shelf= Input\_shelf (only if Input\_Shelf is populated)  And Slot\_Map.Char\_Slot = Input\_Subslot (only if Input\_Subslot is populated)  And PORT\_ASGMT.slot = SLOT\_MAP.int\_slot  And PORT\_ASGMT.port = Input\_port (only if Input\_Shelf and Input\_slot and Input\_port is populated)  And PORT\_ASGMT.slot = Input\_slot (only if Input\_Shelf and Input\_slot is populated)  ~~And PORT\_ASGMT.slot = Input\_slot (only if Input\_slot is populated)~~  **<297409-workItem-251821>**  Address issue of VPLS logic not retrieving the AVPN and MIS PVCs for vr1 routers  Input VPLS-PE (vr1) router supporting AVPN did not return customer records fro mthe above logic.  Fix: In retrieval logic, include checking on PVC Segment number 2 and 3 for the input VPLS router.  ~~For input VPLS (vr1) device, if no customer records returned above logic, do the following:~~ <297409-workItem-251821-2>  **Step 3.a-20: Additional logic retrieve Customer records for the input VPLS-PE (vr1) device**  ~~--for 3 PVC segments~~  --<297409-ORT-Def>  This logic covers both 2-segment and 3-segment cases, with VR1(VPLS) router as gateway on either second or third segmenet.  One Segment instance (without specifying its sig\_no) is used to compare to the input equipment, and this segment can be either #2 or #3 segment.  SELECT Unique  Customer.cust\_id AS customerID,  Customer.cust\_name AS customerName,  Customer.cust\_mcn AS customerMcn,  Site.site\_num,  Site.protocol,  Site.ta\_name,  Site.site\_glbl\_dlci,  Cust\_access.acc\_ckt AS circuitID,  Decode (Instr(Cust\_access.acc\_ckt, '.', 1, 1),  3, 'Y',  5, 'Y',  'N') cktConversionFlag,  Port\_asgmt.port,  Port\_asgmt.slot,  Port\_asgmt.Shelf,  Port\_asgmt.port\_speed,  Site.clli,  Premise.prem\_address,  Premise.prem\_city,  Premise.prem\_state,  Premise.prem\_country,  <297409-CR173072> For input gateway, depends on the conditions in Where clause:  If Local shelf/slot/port are used to compare input parameters, use the Segment.lshelf, lslot, lport here.  If Remote shelf/slot/port are used to compare input parameters, use the Segment.rshelf, rslot, rport here.  Segment.lshelf AS requestEquipShelf,  Segment.lslot AS requestEquipSlot,  Segment.lport AS requestEquipPort  Or  Segment.rshelf AS requestEquipShelf,  Segment.rslot AS requestEquipSlot,  Segment.rport AS requestEquipPort  </297409-CR173072>  FROM  segment Segment, --input device end  segment Segment1, --customer end  segment Segment2,  customer,  cust\_access,  site,  premise,  pvc,  port\_asgmt,  slot\_map <297409-CR173072>  Where 1=1  And ((Segment.lequip\_name = <input\_equip\_name>  And Segment.lshelf = Input\_shelf --only if Input shelf is populated <297409-upd5>  And Segment.lslot = Input\_slot --only if Input slot is populated  ~~And Segment.lslot = Input\_Subslot --only if Input subslot is populated~~  And Slot\_Map.Char\_Slot = Input\_Subslot --only if Input\_Subslot is populated <297409-CR173072>  And Segment.lslot = Slot\_Map.int\_slot (+) --only if Input\_Subslot is populated <297409-CR173072>  And Segment.lport = Input\_port --only if Input\_port is populated  )  Or (Segment.requip\_name = <input-equip\_name>  And Segment.rshelf = Input\_shelf --only if Input shelf is populated <297409-upd5>  And Segment.rslot = Input\_slot --only if Input slot is populated  ~~And Segment.rslot = Input\_Subslot --only if Input subslot is populated~~  And Slot\_Map.Char\_Slot = Input\_Subslot --only if Input\_Subslot is populated <297409-CR173072>  And Segment.rslot = Slot\_Map.int\_slot (+) --only if Input\_Subslot is populated <297409-CR173072>  And Segment.rport = Input\_port --only if Input\_port is populated  )  )  And Segment.pvc\_id = Segment1.pvc\_id  And Segment.pvc\_id = Segment2.pvc\_id  and Segment1.seg\_no = 1  and Segment2.seg\_no = 2  ~~and Segment3.seg\_no = 3~~ <297409-ORT-Def>  And customer.cust\_id = segment1.lcust\_id  ~~And customer.cust\_id != 395 <297409-workItem-251821-2>~~  And cust\_access.site\_id = segment1.lsite\_id  And site.site\_id = segment1.lsite\_id  And premise.loc\_id = site.prem\_loc\_id  ~~And pvc.pvc\_id = segment.pvc\_id~~  And PORT\_ASGMT.site\_id = CUST\_ACCESS.site\_id  And PORT\_ASGMT.site\_id = SITE.site\_id  --<297409-ORT-Def> This section is REMOVED – it’s no longer needed. The above logic covers both 2-segment and 3-segment cases, with VR1(VPLS) router as gateway.  ~~--for 2 PVC segments~~  ~~Union~~  ~~SELECT Unique~~  ~~Customer.cust\_id AS customerID,~~  ~~Customer.cust\_name AS customerName,~~  ~~Customer.cust\_mcn AS customerMcn,~~  ~~Site.site\_num,~~  ~~Site.protocol,~~  ~~Site.ta\_name,~~  ~~Site.site\_glbl\_dlci,~~  ~~Cust\_access.acc\_ckt AS circuitID,~~  ~~Decode (Instr(Cust\_access.acc\_ckt, '.', 1, 1),~~  ~~3, 'Y',~~  ~~5, 'Y',~~  ~~'N') cktConversionFlag,~~  ~~Port\_asgmt.port,~~  ~~Port\_asgmt.slot,~~  ~~Port\_asgmt.Shelf,~~  ~~Port\_asgmt.port\_speed,~~  ~~Site.clli,~~  ~~Premise.prem\_address,~~  ~~Premise.prem\_city,~~  ~~Premise.prem\_state,~~  ~~Premise.prem\_country,~~  ~~<297409-CR173072> For input gateway, depends on the conditions in Where clause:~~  ~~If Local shelf/slot/port are used to compare input parameters, use the Segment.lshelf, lslot, lport here.~~  ~~If Remote shelf/slot/port are used to compare input parameters, use the Segment.rshelf, rslot, rport here.~~  ~~Segment.lshelf AS requestEquipShelf,~~  ~~Segment.lslot AS requestEquipSlot,~~  ~~Segment.lport AS requestEquipPort~~  ~~Or~~  ~~Segment.rshelf AS requestEquipShelf,~~  ~~Segment.rslot AS requestEquipSlot,~~  ~~Segment.rport AS requestEquipPort~~  ~~</297409-CR173072>~~  ~~FROM~~  ~~segment Segment, --inpur device end~~  ~~segment Segment1, --customer end~~  ~~customer,~~  ~~cust\_access,~~  ~~site,~~  ~~premise,~~  ~~pvc,~~  ~~port\_asgmt,~~  ~~slot\_map <297409-CR173072>~~  ~~Where 1=1~~  ~~And ((Segment.lequip\_name = <input\_equip\_name>~~  ~~And Segment.lshelf = Input\_shelf --only if Input shelf is populated <297409-upd5>~~  ~~And Segment.lslot = Input\_slot --only if Input slot is populated~~  ~~And Slot\_Map.Char\_Slot = Input\_Subslot --only if Input\_Subslot is populated <297409-CR173072>~~  ~~And Segment.lslot = Slot\_Map.int\_slot (+) --only if Input\_Subslot is populated <297409-CR173072>~~  ~~And Segment.lport = Input\_port --only if Input\_port is populated~~  ~~)~~  ~~Or (Segment.requip\_name = <input-equip\_name>~~  ~~And Segment.rshelf = Input\_shelf --only if Input shelf is populated <297409-upd5>~~  ~~And Segment.rslot = Input\_slot --only if Input slot is populated~~  ~~And Slot\_Map.Char\_Slot = Input\_Subslot --only if Input\_Subslot is populated <297409-CR173072>~~  ~~And Segment.rslot = Slot\_Map.int\_slot (+) --only if Input\_Subslot is populated <297409-CR173072>~~  ~~And Segment.rport = Input\_port --only if Input\_port is populated~~  ~~)~~  ~~)~~  ~~And Segment.pvc\_id = Segment1.pvc\_id~~  ~~and Segment1.seg\_no = 1~~  ~~and Segment.seg\_no = 2~~  ~~And customer.cust\_id = segment1.lcust\_id~~  ~~And cust\_access.site\_id = segment1.lsite\_id~~  ~~And site.site\_id = segment1.lsite\_id~~  ~~And premise.loc\_id = site.prem\_loc\_id~~  ~~And PORT\_ASGMT.site\_id = CUST\_ACCESS.site\_id~~  ~~And PORT\_ASGMT.site\_id = SITE.site\_id~~  **</297409-workItem-251821>**  <297409-workitem-281399>  Created <ACC-CKT-List> - a CircuitID list (contains the circuitIDs retrieved from Cust\_access.acc\_ckt in above logic)  **Derive TypeOfService data**   * If SITE.protocol is ‘FR’   If PREMISE.prem\_country is ‘USA’  Set TypeOfService = ‘FR’  Else  Set TypeOfService = ‘IFR’  EndIf  EndIf   * If SITE.protocol is ‘ATM’   If PREMISE.prem\_country is ‘USA’  Set TypeOfService = ‘ATM’  Else  Set TypeOfService = ‘IATM’  EndIf  EndIf   * If SITE.protocol is not ‘ATM’ or ‘FR’   Set TypeOfService = SITE.protocol  EndIf  <288315> Derive bvoipIndiactor from ICORE data – For each record retrieved above:  Select service\_option.\*  From  service\_asgmt,  service\_option  Where service\_asgmt.site\_id = <site\_id>  And service\_asgmt.cust\_id = <cust\_id>  And service\_asgmt.serv\_opt\_id = service\_option.serv\_opt\_id  And upper(service\_option.serv\_opt) = ‘BVOIP’  If this returned data, then  Set bvoipIndicator = ‘Y’,  Else  Set bvoipIndicator = ‘N’,  End If  <288315>  Convert the acc\_ckt (in ICORE format) to normalized/BMP format.  This is only for domestic circuit. For international circuit, leave it as is.  If cktConversionFlag = Y, then  Call Common logic –CircuitFormatConversion\_ICORE\_to\_BMP (<acc\_ckt>) -> circuitIdBmpFormat  Else  Set circuitIdBmpFormat = <acc\_ckt>  End If  Retrieve all records from ICORE tables as described in “FR-ATM query response”  <289116.140768-US636605>  **For each Customer record retrieved above:**  call GetUcpeByCircuitID (cust\_access.acc\_ckt) and populate CpeDetailsList  ~~If no data found return error code ‘0000’~~ | | |
| 3.b | **For each Customer record**, retrieve data from GCP **SIDBOR** database using ICORE acc\_ckt data from step 3.a  **Note:** SiDBOR contains only managed devices. So if data is retrieved, then it’s managed. Otherwise (no CER Name), then leave this field blank.  <288315> Add new fields  managedIndicator  parentOrgGroup  **Input: <acc\_ckt> from ICORE, Step 3.a**  Select  FWTOPOLOGY.local\_router AS cerName,  ~~ASSET.src\_org\_id,~~  ASSET.org\_cd AS orgGroup, <288315>  ASSET.managing\_org\_cd AS manageOrg,  ASSET.active\_org\_cd AS activeOrg,  ASSET.functional\_area AS functionalArea,  Decode (Upper(ASSET.service), <288315-US681337-upd3>  ‘IPEFX’, ‘BVOIP’,  ‘IPEFR’, ‘IPFR’,  ‘FRAME’, ‘FR’,  ASSET.service) AS service,  CLIENT\_ORG.access\_type,  CLIENT\_ORG.org\_cd\_prnt AS parentOrgGroup, <288315>  ‘Y’ managedIndicator <288315>  From  FWTOPOLOGY, --from FrameWorks (sourced from ICORE, NC3, …)  ASSET,  CLIENT\_ORG  Where 1=1  And FWTOPOLOGY.local\_access\_ckt = <acc\_ckt>  And FWTOPOLOGY.functional\_area = ASSET.functional\_area  And FWTOPOLOGY.local\_router = ASSET.asset\_nm  And ASSET.org\_cd = CLIENT\_ORG.org\_cd  And ASSET.functional\_area = CLIENT\_ORG.functional\_area  If no data found  Set CERName = blank  Set OrgGroup = blank  Set ManageOrg = blank  Set ActiveOrg = blank  Set FUNCTIONALAREA= blank  Set Service = blank  Set access\_type = blank  Set parentOrgGroup = blank <288315>  Set managedIndicator = ‘N’ <288315>  End If | | |
| 3.c | Document v 5.3 NOTE: this step needs to be commented out until further development with CCI  **Input: <customer\_mcn> from ICORE, Step 3.a**  Retrieve from GCP(DBOR) CCI database using ICORE customer\_mcn data from step 3.a  Match CUSTOMER.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  and CUSTOMER.cust\_mcn (pos. 7-9) =  MCN\_ENTRPRS\_XREF.MCN\_SFX  and XREF\_MCN\_ENTRPRS.ENT\_PTY\_ID = PARTY.ENTRPRS\_PTY\_ID  and PARTY.DUNS\_NB = DNB.DUNS\_NB  Retrieve DNB.(DUNS\_NB, PRNTHQ\_BIZ\_NM)  Set CCI = DUNS\_NB  Set PARENT NAME = PRNTHQ\_BIZ\_NM  Else  IF CUSTOMER.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  and XREF\_MCN\_ENTRPRS.ENT\_PTY\_ID = PARTY.ENTRPRS\_PTY\_ID  and PARTY.DUNS\_NB = DNB.DUNS\_NB  Retrieve DNB.(DUNS\_NB, PRNTHQ\_BIZ\_NM)  Set CCI = DUNS\_NB  Set PARENT NAME = PRNTHQ\_BIZ\_NM    Else  Set CCI = blank  Set PARENT NAME = blank  EndIf  EndIf | | |
| 4 | | **Retrieve PVC records from ICORE (GCP replication) tables**:  <288315> Adding new fields for both A and Z ends  circuitIdBmpFormat  Retrieve From **PVC**  **<297409-workItem-251821-3>**  **Execute both Step 4.A-10 and 4.A-20 for PVC records retrieval**  **STEP 4.A-10: Retrieve PVC records**  Select  PVC.pvc\_id,  PVC.rSwitch, PVC.rSlot, PVC.rPort, PVC.pvc\_rdlci, PVC.pvc\_rontrcir, PVC.pvc\_rVci, PVC.pvc\_rVpi,  PVC.lSwitch, PVC.lSlot, PVC.lPort, PVC.pvc\_ldlci, PVC.pvc\_lontrcir, PVC.pvc\_lVci, PVC.pvc\_lVpi,  VPNA.vpn\_name, VPNZ.vpn\_name,  IPFRA.vpn\_id, IPFRZ.vpn\_id  SiteA.site\_id, SiteA.full\_port\_speed, SiteA.grc, SiteA.Clli  SiteZ.site\_id, SiteZ.full\_port\_speed, SiteZ.grc, SiteZ.Clli  CustomerA.cust\_id, CustomerA.cust\_name, CustomerA.cust\_mcn,  CustomerZ.cust\_id, CustomerZ.cust\_name, CustomerZ.cust\_mcn,  PremiseA.loc\_id, PremiseA.prem\_address, PremiseA.prem\_city, PremiseA.prem\_state, PremiseA.prem\_country,  PremiseZ.loc\_id, PremiseZ.prem\_address, PremiseZ.prem\_city, PremiseZ.prem\_state, PremiseZ.prem\_country,  Cust\_accessA.acc\_ckt,  Cust\_accessZ.acc\_ckt,  Decode (Cust\_accessA.acc\_ckt,  Instr (Cust\_accessA.acc\_ckt, ‘.’, 1, 1) = 3, ‘Y’,  Instr (Cust\_accessA.acc\_ckt, ‘.’, 1, 1) = 5, ‘Y’,  ‘N’ ) cktConversionFlagA, <288315> Used to mark whether the circuit format can be converted.  Decode (Cust\_accessZ.acc\_ckt,  Instr (Cust\_accessZ.acc\_ckt, ‘.’, 1, 1) = 3, ‘Y’,  Instr (Cust\_accessZ.acc\_ckt, ‘.’, 1, 1) = 5, ‘Y’,  ‘N’ ) cktConversionFlagZ, <288315> Used to mark whether the circuit format can be converted.  From  PVC,  IPFR IpfrA, IPFR IpfrZ,  VPN vpnA, VPN vpnZ,  Customer CustomerA, Customer CustomerZ,  Cust\_access Cust\_accessA, Cust\_access Cust\_accessZ,  SITE SiteA, SITE SiteZ,  PREMISE PremiseA, PRIMESE PremiseZ,  PORT\_ASGMT,  EQUIPMENT  Where  SITE.site\_id = PORT\_ASGMT.site\_id(+) And PORT\_ASGMT.site\_id = CUST\_ACCESS.site\_id(+) And PORT\_ASGMT.equip\_id = EQUIPMENT.equip\_id(+)  And SiteZ.prem\_loc\_id = PremiseZ.loc\_id(+) And SiteA.prem\_loc\_id = PremiseA.loc\_id And SiteZ.site\_id = PVC.pvc\_rsite\_id(+) And SiteA.site\_id = PVC.pvc\_lsite\_id And PVC.pvc\_id = IpfrA.pvc\_id(+)  And PVC.pvc\_id = IpfrZ.pvc\_id(+)  And IpfrA.vpn\_id = VpnA.vpn\_id  And IpfrZ.vpn\_id = VpnZ.vpn\_id  And PVC.pvc\_rcust\_id = customerZ.cust\_id  And PVC.pvc\_lcust\_id = customerA.cust\_id  And Cust\_accessZ.cust\_id = CustomerZ.cust\_id(+)  And Cust\_accessA.cust\_id = CustomerA.cust\_id(+)  And EQUIPMENT.equip\_name = Input\_EquipName  And PORT\_ASGMT.port = Input\_port (only if Input\_port is populated)  And PORT\_ASGMT.slot = Input\_slot (only if Input\_slot is populated)  And EQUIPMENT.equip\_name = Input\_EquipName  And PORT\_ASGMT.shelf= Input\_shelf (only if Input\_Shelf is populated)  And Slot\_Map.Char\_Slot = Input\_Subslot (only if Input\_Subslot is populated)  And PORT\_ASGMT.slot = SLOT\_MAP.int\_slot <297409-CR173072>  And PORT\_ASGMT.port = Input\_port (only if Input\_Shelf and Input\_slot and Input\_port is populated)  And PORT\_ASGMT.slot = Input\_slot (only if Input\_Shelf and Input\_slot is populated)  ~~Match~~  ~~AND SITEA.site\_id = CUST\_PREM\_EQUIPA.~~  ~~And SITEZ.site\_id = CUST\_PREM\_EQUIPZ.~~  **<297409-workItem-251821-3>**  Address issue of VPLS logic not retrieving the AVPN and MIS PVCs for vr1 routers  In retrieval logic, include checking on PVC Segment number 2 and 3 for the input router.  **--<297409-ORT-Def>**  This logic covers both 2-segment and 3-segment cases, with VR1(VPLS) router as gateway on either second or third segmenet.  One Segment instance (without specifying its sig\_no) is used to compare to the input equipment, and this segment can be either #2 or #3 segment.  **Step 4.A-20: Additional logic retrieve PVC records for the input vr1 device**  Select  PVC.pvc\_id,  PVC.rSwitch, PVC.rSlot, PVC.rPort, PVC.pvc\_rdlci, PVC.pvc\_rontrcir, PVC.pvc\_rVci, PVC.pvc\_rVpi,  PVC.lSwitch, PVC.lSlot, PVC.lPort, PVC.pvc\_ldlci, PVC.pvc\_lontrcir, PVC.pvc\_lVci, PVC.pvc\_lVpi,  VPNA.vpn\_name, VPNZ.vpn\_name,  IPFRA.vpn\_id, IPFRZ.vpn\_id  SiteA.site\_id, SiteA.full\_port\_speed, SiteA.grc, SiteA.Clli  SiteZ.site\_id, SiteZ.full\_port\_speed, SiteZ.grc, SiteZ.Clli  CustomerA.cust\_id, CustomerA.cust\_name, CustomerA.cust\_mcn,  CustomerZ.cust\_id, CustomerZ.cust\_name, CustomerZ.cust\_mcn,  PremiseA.loc\_id, PremiseA.prem\_address, PremiseA.prem\_city, PremiseA.prem\_state, PremiseA.prem\_country,  PremiseZ.loc\_id, PremiseZ.prem\_address, PremiseZ.prem\_city, PremiseZ.prem\_state, PremiseZ.prem\_country,  Cust\_accessA.acc\_ckt,  Cust\_accessZ.acc\_ckt,  Decode (Cust\_accessA.acc\_ckt,  Instr (Cust\_accessA.acc\_ckt, ‘.’, 1, 1) = 3, ‘Y’,  Instr (Cust\_accessA.acc\_ckt, ‘.’, 1, 1) = 5, ‘Y’,  ‘N’ ) cktConversionFlagA, <288315> Used to mark whether the circuit format can be converted.  Decode (Cust\_accessZ.acc\_ckt,  Instr (Cust\_accessZ.acc\_ckt, ‘.’, 1, 1) = 3, ‘Y’,  Instr (Cust\_accessZ.acc\_ckt, ‘.’, 1, 1) = 5, ‘Y’,  ‘N’ ) cktConversionFlagZ, <288315> Used to mark whether the circuit format can be converted.  From  PVC,  IPFR IpfrA, IPFR IpfrZ,  VPN vpnA, VPN vpnZ,  Customer CustomerA, Customer CustomerZ,  Cust\_access Cust\_accessA, Cust\_access Cust\_accessZ,  SITE SiteA, SITE SiteZ,  PREMISE PremiseA, PRIMESE PremiseZ,  PORT\_ASGMT,  EQUIPMENT  segment Segment,  segment Segment1,  segment Segment2,  slot\_map <297409-CR173072>  Where 1=1  And ((Segment.lequip\_name = <input\_equip\_name>  And Segment.lshelf – Input\_shelf --only if Input shelf is populated <297409-upd5>  And Segment.lslot – Input\_slot --only if Input slot is populated  ~~And Segment.lslot – Input\_Subslot --only if Input Subslot is populated~~  And Slot\_Map.Char\_Slot = Input\_Subslot --only if Input\_Subslot is populated <297409-CR173072>  And Segment.lslot = Slot\_Map.int\_slot (+) --only if Input\_Subslot is populated <297409-CR173072>  And Segment.lport = Input\_port --only if Input\_port is populated  )  Or (Segment.requip\_name = <input-equip\_name>  And Segment.rshelf – Input\_shelf --only if Input shelf is populated <297409-upd5>  And Segment.rslot – Input\_slot --only if Input slot is populated  ~~And Segment.rslot – Input\_Subslot --only if Input Subslot is populated~~  And Slot\_Map.Char\_Slot = Input\_Subslot --only if Input\_Subslot is populated <297409-CR173072>  And Segment.rslot = Slot\_Map.int\_slot (+) --only if Input\_Subslot is populated <297409-CR173072>  And Segment.rport = Input\_port --only if Input\_port is populated  )  )  And Segment.pvc\_id = Segment1.pvc\_id  And Segment.pvc\_id = Segment2.pvc\_id  And Segment1.seg\_no = 1  And Segment2.seg\_no = 2  ~~And Segment3.seg\_no = 3~~ <297409-ORT-Def>  And Segment.pvc\_id = pvc.pvc\_id  AND (segment1.lequip\_name = equipment.equip\_name  OR segment1.requip\_name = equipment.equip\_name  )  And SITE.site\_id = PORT\_ASGMT.site\_id(+) And PORT\_ASGMT.site\_id = CUST\_ACCESS.site\_id(+) And PORT\_ASGMT.equip\_id = EQUIPMENT.equip\_id(+)  And SiteZ.prem\_loc\_id = PremiseZ.loc\_id(+) And SiteA.prem\_loc\_id = PremiseA.loc\_id And SiteZ.site\_id = PVC.pvc\_rsite\_id(+) And SiteA.site\_id = PVC.pvc\_lsite\_id And PVC.pvc\_id = IpfrA.pvc\_id(+)  And PVC.pvc\_id = IpfrZ.pvc\_id(+)  And IpfrA.vpn\_id = VpnA.vpn\_id  And IpfrZ.vpn\_id = VpnZ.vpn\_id  And PVC.pvc\_rcust\_id = customerZ.cust\_id  And PVC.pvc\_lcust\_id = customerA.cust\_id  ~~And Cust\_accessZ.cust\_id = CustomerZ.cust\_id(+)~~ <297409-workItem-251821-4>  And Cust\_accessA.cust\_id = CustomerA.cust\_id(+)  And cust\_accessZ.cust\_id (+) = customerZ.cust\_id <297409-workItem-251821-4>  And cust\_accessZ.site\_id (+) = siteZ.site\_id <297409-workItem-251821-4>  ~~And EQUIPMENT.equip\_name = Input\_EquipName~~  ~~And PORT\_ASGMT.port = Input\_port (only if Input\_port is populated)~~  ~~And PORT\_ASGMT.slot = Input\_slot (only if Input\_slot is populated)~~  --<297409-ORT-Def> This section is REMOVED – it’s no longer needed. The above logic covers both 2-segment and 3-segment cases, with VR1(VPLS) router as gateway.  ~~Union~~  ~~Select~~  ~~PVC.pvc\_id,~~  ~~PVC.rSwitch, PVC.rSlot, PVC.rPort, PVC.pvc\_rdlci, PVC.pvc\_rontrcir, PVC.pvc\_rVci, PVC.pvc\_rVpi,~~  ~~PVC.lSwitch, PVC.lSlot, PVC.lPort, PVC.pvc\_ldlci, PVC.pvc\_lontrcir, PVC.pvc\_lVci, PVC.pvc\_lVpi,~~  ~~VPNA.vpn\_name, VPNZ.vpn\_name,~~  ~~IPFRA.vpn\_id, IPFRZ.vpn\_id~~  ~~SiteA.site\_id, SiteA.full\_port\_speed, SiteA.grc, SiteA.Clli~~  ~~SiteZ.site\_id, SiteZ.full\_port\_speed, SiteZ.grc, SiteZ.Clli~~  ~~CustomerA.cust\_id, CustomerA.cust\_name, CustomerA.cust\_mcn,~~  ~~CustomerZ.cust\_id, CustomerZ.cust\_name, CustomerZ.cust\_mcn,~~  ~~PremiseA.loc\_id, PremiseA.prem\_address, PremiseA.prem\_city, PremiseA.prem\_state, PremiseA.prem\_country,~~  ~~PremiseZ.loc\_id, PremiseZ.prem\_address, PremiseZ.prem\_city, PremiseZ.prem\_state, PremiseZ.prem\_country,~~  ~~Cust\_accessA.acc\_ckt,~~  ~~Cust\_accessZ.acc\_ckt,~~  ~~Decode (Cust\_accessA.acc\_ckt,~~  ~~Instr (Cust\_accessA.acc\_ckt, ‘.’, 1, 1) = 3, ‘Y’,~~  ~~Instr (Cust\_accessA.acc\_ckt, ‘.’, 1, 1) = 5, ‘Y’,~~  ~~‘N’ ) cktConversionFlagA, <288315> Used to mark whether the circuit format can be converted.~~  ~~Decode (Cust\_accessZ.acc\_ckt,~~  ~~Instr (Cust\_accessZ.acc\_ckt, ‘.’, 1, 1) = 3, ‘Y’,~~  ~~Instr (Cust\_accessZ.acc\_ckt, ‘.’, 1, 1) = 5, ‘Y’,~~  ~~‘N’ ) cktConversionFlagZ, <288315> Used to mark whether the circuit format can be converted.~~  ~~From~~  ~~PVC,~~  ~~IPFR IpfrA, IPFR IpfrZ,~~  ~~VPN vpnA, VPN vpnZ,~~  ~~Customer CustomerA, Customer CustomerZ,~~  ~~Cust\_access Cust\_accessA, Cust\_access Cust\_accessZ,~~  ~~SITE SiteA, SITE SiteZ,~~  ~~PREMISE PremiseA, PRIMESE PremiseZ,~~  ~~PORT\_ASGMT,~~  ~~EQUIPMENT~~  ~~segment Segment,~~  ~~segment Segment1,~~  ~~slot\_map <297409-CR173072>~~  ~~Where 1=1~~  ~~And ((Segment.lequip\_name = <input\_equip\_name>~~  ~~And Segment.lshelf = Input\_shelf --only if Input shelf is populated <297409-upd5>~~  ~~And Segment.lslot = Input\_slot --only if Input slot is populated~~  ~~And Slot\_Map.Char\_Slot = Input\_Subslot --only if Input\_Subslot is populated <297409-CR173072>~~  ~~And Segment.lslot = Slot\_Map.int\_slot (+) --only if Input\_Subslot is populated <297409-CR173072>~~  ~~And Segment.lport = Input\_port --only if Input\_port is populated~~  ~~)~~  ~~Or (Segment.requip\_name = <input-equip\_name>~~  ~~And Segment.rshelf – Input\_shelf --only if Input shelf is populated <297409-upd5>~~  ~~And Segment.rslot – Input\_slot --only if Input slot is populated~~  ~~And Slot\_Map.Char\_Slot = Input\_Subslot --only if Input\_Subslot is populated <297409-CR173072>~~  ~~And Segment.rslot = Slot\_Map.int\_slot (+) --only if Input\_Subslot is populated <297409-CR173072>~~  ~~And Segment.rport = Input\_port --only if Input\_port is populated~~  ~~)~~  ~~)~~  ~~And Segment.pvc\_id = Segment1.pvc\_id~~  ~~And Segment1.seg\_no = 1~~  ~~And Segment.seg\_no = 2~~  ~~And Segment.pvc\_id = pvc.pvc\_id~~  ~~AND (segment1.lequip\_name = equipment.equip\_name~~  ~~OR segment1.requip\_name = equipment.equip\_name~~  ~~)~~  ~~And SITE.site\_id = PORT\_ASGMT.site\_id(+) And PORT\_ASGMT.site\_id = CUST\_ACCESS.site\_id(+) And PORT\_ASGMT.equip\_id = EQUIPMENT.equip\_id(+)~~  ~~And SiteZ.prem\_loc\_id = PremiseZ.loc\_id(+) And SiteA.prem\_loc\_id = PremiseA.loc\_id And SiteZ.site\_id = PVC.pvc\_rsite\_id(+) And SiteA.site\_id = PVC.pvc\_lsite\_id And PVC.pvc\_id = IpfrA.pvc\_id(+)~~  ~~And PVC.pvc\_id = IpfrZ.pvc\_id(+)~~  ~~And IpfrA.vpn\_id = VpnA.vpn\_id~~  ~~And IpfrZ.vpn\_id = VpnZ.vpn\_id~~  ~~And PVC.pvc\_rcust\_id = customerZ.cust\_id~~  ~~And PVC.pvc\_lcust\_id = customerA.cust\_id~~  ~~And Cust\_accessA.cust\_id = CustomerA.cust\_id(+)~~  ~~And cust\_accessZ.cust\_id (+) = customerZ.cust\_id <297409-workItem-251821-4>~~  ~~And cust\_accessZ.site\_id (+) = siteZ.site\_id <297409-workItem-251821-4>~~  **Step 4.A-30:** <297409-workItem-251821-4>  Additional logic retrieve Ethernet PVC records for the input vr1 device.  These PVCs do not have VLN, so IPFR and VPN tables are excluded from this logic, and all the fields populated using data from these two tables are left blank.  --<297409-ORT-Def>  This logic covers both 2-segment and 3-segment cases, with VR1(VPLS) router as gateway on either second or third segmenet.  One Segment instance (without specifying its sig\_no) is used to compare to the input equipment, and this segment can be either #2 or #3 segment.  Select  PVC.pvc\_id,  PVC.rSwitch, PVC.rSlot, PVC.rPort, PVC.pvc\_rdlci, PVC.pvc\_rontrcir, PVC.pvc\_rVci, PVC.pvc\_rVpi,  PVC.lSwitch, PVC.lSlot, PVC.lPort, PVC.pvc\_ldlci, PVC.pvc\_lontrcir, PVC.pvc\_lVci, PVC.pvc\_lVpi,  ~~VPNA.vpn\_name, VPNZ.vpn\_name,~~  ~~IPFRA.vpn\_id, IPFRZ.vpn\_id~~  SiteA.site\_id, SiteA.full\_port\_speed, SiteA.grc, SiteA.Clli  SiteZ.site\_id, SiteZ.full\_port\_speed, SiteZ.grc, SiteZ.Clli  CustomerA.cust\_id, CustomerA.cust\_name, CustomerA.cust\_mcn,  CustomerZ.cust\_id, CustomerZ.cust\_name, CustomerZ.cust\_mcn,  PremiseA.loc\_id, PremiseA.prem\_address, PremiseA.prem\_city, PremiseA.prem\_state, PremiseA.prem\_country,  PremiseZ.loc\_id, PremiseZ.prem\_address, PremiseZ.prem\_city, PremiseZ.prem\_state, PremiseZ.prem\_country,  Cust\_accessA.acc\_ckt,  Cust\_accessZ.acc\_ckt,  Decode (Cust\_accessA.acc\_ckt,  Instr (Cust\_accessA.acc\_ckt, ‘.’, 1, 1) = 3, ‘Y’,  Instr (Cust\_accessA.acc\_ckt, ‘.’, 1, 1) = 5, ‘Y’,  ‘N’ ) cktConversionFlagA, <288315> Used to mark whether the circuit format can be converted.  Decode (Cust\_accessZ.acc\_ckt,  Instr (Cust\_accessZ.acc\_ckt, ‘.’, 1, 1) = 3, ‘Y’,  Instr (Cust\_accessZ.acc\_ckt, ‘.’, 1, 1) = 5, ‘Y’,  ‘N’ ) cktConversionFlagZ, <288315> Used to mark whether the circuit format can be converted.  From  PVC,  ~~IPFR IpfrA, IPFR IpfrZ,~~  ~~VPN vpnA, VPN vpnZ,~~  Customer CustomerA, Customer CustomerZ,  Cust\_access Cust\_accessA, Cust\_access Cust\_accessZ,  SITE SiteA, SITE SiteZ,  PREMISE PremiseA, PRIMESE PremiseZ,  PORT\_ASGMT,  EQUIPMENT  segment Segment,  segment Segment1,  segment Segment2,  slot\_map <297409-CR173072>  Where 1=1  And ((Segment.lequip\_name = <input\_equip\_name>  And Segment.lshelf – Input\_shelf --only if Input shelf is populated <297409-upd5>  And Segment.lslot – Input\_slot --only if Input slot is populated  ~~And Segment.lslot – Input\_Subslot --only if Input subslot is populated~~  And Slot\_Map.Char\_Slot = Input\_Subslot --only if Input\_Subslot is populated <297409-CR173072>  And Segment.lslot = Slot\_Map.int\_slot (+) --only if Input\_Subslot is populated <297409-CR173072>  And Segment.lport = Input\_port --only if Input\_port is populated  )  Or (Segment.requip\_name = <input-equip\_name>  And Segment.rshelf – Input\_shelf --only if Input shelf is populated <297409-upd5>  And Segment.rslot – Input\_slot --only if Input slot is populated  ~~And Segment.rslot – Input\_Subslot --only if Input subslot is populated~~  And Slot\_Map.Char\_Slot = Input\_Subslot --only if Input\_Subslot is populated <297409-CR173072>  And Segment.rslot = Slot\_Map.int\_slot (+) --only if Input\_Subslot is populated <297409-CR173072>  And Segment.rport = Input\_port --only if Input\_port is populated  )  )  And Segment.pvc\_id = Segment1.pvc\_id  And Segment.pvc\_id = Segment2.pvc\_id  And Segment1.seg\_no = 1  And Segment2.seg\_no = 2  ~~And segment Segment3~~ <297409-ORT-Def>  And Segment.pvc\_id = pvc.pvc\_id  AND (segment1.lequip\_name = equipment.equip\_name  OR segment1.requip\_name = equipment.equip\_name  )  And SITE.site\_id = PORT\_ASGMT.site\_id(+) And PORT\_ASGMT.site\_id = CUST\_ACCESS.site\_id(+) And PORT\_ASGMT.equip\_id = EQUIPMENT.equip\_id(+)  And SiteZ.prem\_loc\_id = PremiseZ.loc\_id(+) And SiteA.prem\_loc\_id = PremiseA.loc\_id And SiteZ.site\_id = PVC.pvc\_rsite\_id(+) And SiteA.site\_id = PVC.pvc\_lsite\_id ~~And PVC.pvc\_id = IpfrA.pvc\_id(+)~~  ~~And PVC.pvc\_id = IpfrZ.pvc\_id(+)~~  ~~And IpfrA.vpn\_id = VpnA.vpn\_id~~  ~~And IpfrZ.vpn\_id = VpnZ.vpn\_id~~  And PVC.pvc\_rcust\_id = customerZ.cust\_id  And PVC.pvc\_lcust\_id = customerA.cust\_id  And Cust\_accessA.cust\_id = CustomerA.cust\_id(+)  And cust\_accessZ.cust\_id (+) = customerZ.cust\_id  And cust\_accessZ.site\_id (+) = siteZ.site\_id  And pvc.pvc\_type = ‘ETHERNET’  --<297409-ORT-Def> This section is REMOVED – it’s no longer needed. The above logic covers both 2-segment and 3-segment cases, with VR1(VPLS) router as gateway.  Union  Select  PVC.pvc\_id,  PVC.rSwitch, PVC.rSlot, PVC.rPort, PVC.pvc\_rdlci, PVC.pvc\_rontrcir, PVC.pvc\_rVci, PVC.pvc\_rVpi,  PVC.lSwitch, PVC.lSlot, PVC.lPort, PVC.pvc\_ldlci, PVC.pvc\_lontrcir, PVC.pvc\_lVci, PVC.pvc\_lVpi,  ~~VPNA.vpn\_name, VPNZ.vpn\_name,~~  ~~IPFRA.vpn\_id, IPFRZ.vpn\_id~~  SiteA.site\_id, SiteA.full\_port\_speed, SiteA.grc, SiteA.Clli  SiteZ.site\_id, SiteZ.full\_port\_speed, SiteZ.grc, SiteZ.Clli  CustomerA.cust\_id, CustomerA.cust\_name, CustomerA.cust\_mcn,  CustomerZ.cust\_id, CustomerZ.cust\_name, CustomerZ.cust\_mcn,  PremiseA.loc\_id, PremiseA.prem\_address, PremiseA.prem\_city, PremiseA.prem\_state, PremiseA.prem\_country,  PremiseZ.loc\_id, PremiseZ.prem\_address, PremiseZ.prem\_city, PremiseZ.prem\_state, PremiseZ.prem\_country,  Cust\_accessA.acc\_ckt,  Cust\_accessZ.acc\_ckt,  Decode (Cust\_accessA.acc\_ckt,  Instr (Cust\_accessA.acc\_ckt, ‘.’, 1, 1) = 3, ‘Y’,  Instr (Cust\_accessA.acc\_ckt, ‘.’, 1, 1) = 5, ‘Y’,  ‘N’ ) cktConversionFlagA, <288315> Used to mark whether the circuit format can be converted.  Decode (Cust\_accessZ.acc\_ckt,  Instr (Cust\_accessZ.acc\_ckt, ‘.’, 1, 1) = 3, ‘Y’,  Instr (Cust\_accessZ.acc\_ckt, ‘.’, 1, 1) = 5, ‘Y’,  ‘N’ ) cktConversionFlagZ, <288315> Used to mark whether the circuit format can be converted.  From  PVC,  ~~IPFR IpfrA, IPFR IpfrZ,~~  ~~VPN vpnA, VPN vpnZ,~~  Customer CustomerA, Customer CustomerZ,  Cust\_access Cust\_accessA, Cust\_access Cust\_accessZ,  SITE SiteA, SITE SiteZ,  PREMISE PremiseA, PRIMESE PremiseZ,  PORT\_ASGMT,  EQUIPMENT  segment Segment,  segment Segment1,  slot\_map <297409-CR173072>  Where 1=1  And ((Segment.lequip\_name = <input\_equip\_name>  And Segment.lshelf – Input\_shelf --only if Input shelf is populated <297409-upd5>  And Segment.lslot – Input\_slot --only if Input slot is populated  ~~And Segment.lslot – Input\_Subslot --only if Input subslot is populated~~  And Slot\_Map.Char\_Slot = Input\_Subslot --only if Input\_Subslot is populated <297409-CR173072>  And Segment.lslot = Slot\_Map.int\_slot (+) --only if Input\_Subslot is populated <297409-CR173072>  And Segment.lport = Input\_port --only if Input\_port is populated  )  Or (Segment.requip\_name = <input-equip\_name>  And Segment.rshelf – Input\_shelf --only if Input shelf is populated <297409-upd5>  And Segment.rslot – Input\_slot --only if Input slot is populated  ~~And Segment.rslot – Input\_Subslot --only if Input subslot is populated~~  And Slot\_Map.Char\_Slot = Input\_Subslot --only if Input\_Subslot is populated <297409-CR173072>  And Segment.rslot = Slot\_Map.int\_slot (+) --only if Input\_Subslot is populated <297409-CR173072>  And Segment.rport = Input\_port --only if Input\_port is populated  )  )  And Segment.pvc\_id = Segment1.pvc\_id  And Segment1.seg\_no = 1  And Segment.seg\_no = 2  And Segment.pvc\_id = pvc.pvc\_id  AND (segment1.lequip\_name = equipment.equip\_name  OR segment1.requip\_name = equipment.equip\_name  )  And SITE.site\_id = PORT\_ASGMT.site\_id(+) And PORT\_ASGMT.site\_id = CUST\_ACCESS.site\_id(+) And PORT\_ASGMT.equip\_id = EQUIPMENT.equip\_id(+)  And SiteZ.prem\_loc\_id = PremiseZ.loc\_id(+) And SiteA.prem\_loc\_id = PremiseA.loc\_id And SiteZ.site\_id = PVC.pvc\_rsite\_id(+) And SiteA.site\_id = PVC.pvc\_lsite\_id ~~And PVC.pvc\_id = IpfrA.pvc\_id(+)~~  ~~And PVC.pvc\_id = IpfrZ.pvc\_id(+)~~  ~~And IpfrA.vpn\_id = VpnA.vpn\_id~~  ~~And IpfrZ.vpn\_id = VpnZ.vpn\_id~~  And PVC.pvc\_rcust\_id = customerZ.cust\_id  And PVC.pvc\_lcust\_id = customerA.cust\_id  And Cust\_accessA.cust\_id = CustomerA.cust\_id(+)  And cust\_accessZ.cust\_id (+) = customerZ.cust\_id  And cust\_accessZ.site\_id (+) = siteZ.site\_id  And pvc.pvc\_type = ‘ETHERNET’  </297409-workItem-251821-4>  <288315>  Convert the acc\_ckt (in ICORE format) to normalized/BMP format, based on the cktConversionFlag value.  This is only for domestic circuit. For international circuit, leave it as is.  If cktConversionFlagA = Y, then  Call Common logic –CircuitFormatConversion\_ICORE\_to\_BMP (<A\_acc\_ckt>) -> A\_circuitIdBmpFormat  Else  Set A\_circuitIdBmpFormat = <A\_acc\_ckt>  End If  If cktConversionFlagZ = Y, then  Call Common logic –CircuitFormatConversion\_ICORE\_to\_BMP (<Z\_acc\_ckt>) -> Z\_circuitIdBmpFormat  Else  Set Z\_circuitIdBmpFormat = <Z\_acc\_ckt>  End If | |
|  | | **<288315-US681337-upd3>**  **Production SQL code retrieving PVC records: ServiceType=FR-ATM, FR-ATM-RPM**  **Updated to fix production issue – duplicate PVC records, and incorrect CER IP and PE IP:**  **SELECT DISTINCT**  vpn.cust\_id VPNCUSTID,  pvc.pvc\_id,  pvc.PVC\_TYPE,  pvc.rswitch,  pvc.rslot,  pvc.rport,  pvc.pvc\_rdlci,  pvc.pvc\_rcontrcir,  pvc.pvc\_rvci,  pvc.pvc\_rvpi,  pvc.lswitch,  pvc.lslot,  pvc.lport,  pvc.pvc\_ldlci,  pvc.pvc\_lcontrcir,  pvc.pvc\_lvci,  pvc.pvc\_lvpi,  customer.cust\_name acust\_name,  customer.cust\_id acust\_id,  customer.cust\_mcn acust\_mcn,  site.site\_id asite\_id,  site.site\_num asite\_num,  site.full\_port\_speed afull\_port\_speed,  site.grc agrc,  site.protocol aprotocol,  site.ta\_name ata\_name,  site.site\_glbl\_dlci asite\_glbl\_dlci,  site.clli aclli,  cust\_access.acc\_ckt aacc\_ckt,  port\_asgmt.port aport,  port\_asgmt.port\_speed aport\_speed,  premise.loc\_id aloc\_id,  premise.prem\_address aprem\_address,  premise.prem\_city aprem\_city,  premise.prem\_state aprem\_state,  premise.prem\_country aprem\_country,  --ipfr.cpe\_ip\_address a\_cust\_ip\_address,  --ipfr.per\_ip\_address z\_cust\_ip\_address,  --// cust\_id=395 is for ATT, PE side;  --// If A end has cust\_id=395, then the cust\_Ip is the PE IP; else, the Cust\_IP is the CE IP  Decode (customer.cust\_id, ‘395’, ipfr.per\_ip\_address, ipfr.cpe\_ip\_address) a\_cust\_ip\_address,  Decode (c2.cust\_id, ‘395’, ipfr.per\_ip\_address, ipfr.cpe\_ip\_address) z\_cust\_ip\_address,  ipfr.vpn\_id,  c2.cust\_name zcust\_name,  c2.cust\_id zcust\_id,  c2.cust\_mcn zcust\_mcn,  s2.site\_id zsite\_id,  s2.site\_num zsite\_num,  s2.full\_port\_speed zfull\_port\_speed,  s2.grc zgrc,  s2.protocol zprotocol,  s2.ta\_name zta\_name,  s2.site\_glbl\_dlci zsite\_glbl\_dlci,  s2.clli zclli,  ca2.acc\_ckt zacc\_ckt,  pa2.port zport,  pa2.port\_speed zport\_speed,  p2.loc\_id zloc\_id,  p2.prem\_address zprem\_address,  p2.prem\_city zprem\_city,  p2.prem\_state zprem\_state,  p2.prem\_country zprem\_country,  vpn.vpn\_name,  vpn.vpn\_id,  **--//A and Z ends use the same vpn customer data, okay when there’s only one VPN.**  cvpn.cust\_name avpn\_name,  cvpn.cust\_mcn avpn\_mcn,  cvpn.cust\_name zvpn\_name,  cvpn.cust\_mcn zvpn\_mcn,  cvpn.cust\_id a\_vpn\_cust\_id,  cvpn.cust\_id z\_vpn\_cust\_id  **FROM**  premise p2,  port\_asgmt pa2,  cust\_access ca2,  site s2,  customer c2,  ipfr,  premise,  port\_asgmt,  cust\_access,  site,  customer,  pvc,  equipment,  vpn,  customer cvpn  **WHERE** upper(equipment.equip\_name) = <equipName>  AND ((pvc.lequip\_id = equipment.equip\_id  AND (pvc.lport = <input port>  AND pvc.lslot = <input slot>  AND pvc.lshelf = <input shelf>)  )  **OR** (pvc.requip\_id = equipment.equip\_id  AND (pvc.rport = <input port>  AND pvc.rslot = <input slot>  AND pvc.rshelf = <input shelf>)  )  )  AND customer.cust\_id = pvc.pvc\_lcust\_id  AND site.site\_id = pvc.pvc\_lsite\_id  AND cust\_access.site\_id = site.site\_id  AND port\_asgmt.site\_id = site.site\_id  AND premise.loc\_id = site.prem\_loc\_id  AND ipfr.pvc\_id = pvc.pvc\_id  AND c2.cust\_id = pvc.pvc\_rcust\_id  AND s2.site\_id = pvc.pvc\_rsite\_id  AND ca2.site\_id = s2.site\_id  AND pa2.site\_id = s2.site\_id  AND p2.loc\_id = s2.prem\_loc\_id  AND ipfr.vpn\_id = vpn.vpn\_id  AND vpn.cust\_id = cvpn.cust\_id  AND (pvc.pvc\_type != 'IPVC' AND pvc.pvc\_type != 'PNPVC')  **UNION**  **--new logic for pvc.pvc\_type IN ( 'IPVC', 'PNPVC')**  **SELECT DISTINCT**  vpn.cust\_id VPNCUSTID,  pvc.pvc\_id,  pvc.PVC\_TYPE,  pvc.rswitch,  pvc.rslot,  pvc.rport,  pvc.pvc\_rdlci,  pvc.pvc\_rcontrcir,  pvc.pvc\_rvci,  pvc.pvc\_rvpi,  pvc.lswitch,  pvc.lslot,  pvc.lport,  pvc.pvc\_ldlci,  pvc.pvc\_lcontrcir,  pvc.pvc\_lvci,  pvc.pvc\_lvpi,  customer.cust\_name acust\_name,  customer.cust\_id acust\_id,  customer.cust\_mcn acust\_mcn,  site.site\_id asite\_id,  site.site\_num asite\_num,  site.full\_port\_speed afull\_port\_speed,  site.grc agrc,  site.protocol aprotocol,  site.ta\_name ata\_name,  site.site\_glbl\_dlci asite\_glbl\_dlci,  site.clli aclli,  cust\_access.acc\_ckt aacc\_ckt,  port\_asgmt.port aport,  port\_asgmt.port\_speed aport\_speed,  premise.loc\_id aloc\_id,  premise.prem\_address aprem\_address,  premise.prem\_city aprem\_city,  premise.prem\_state aprem\_state,  premise.prem\_country aprem\_country,  --ipfr.cpe\_ip\_address a\_cust\_ip\_address,  --ipfr.per\_ip\_address z\_cust\_ip\_address,  --// If A end has cust\_id=395, then the cust\_Ip is the PE IP; else, the Cust\_IP is the CE IP  Decode (customer.cust\_id, ‘395’, ipfr.per\_ip\_address, ipfr.cpe\_ip\_address) a\_cust\_ip\_address,  Decode (c2.cust\_id, ‘395’, ipfr.per\_ip\_address, ipfr.cpe\_ip\_address) z\_cust\_ip\_address,  ipfr.vpn\_id,  c2.cust\_name zcust\_name,  c2.cust\_id zcust\_id,  c2.cust\_mcn zcust\_mcn,  s2.site\_id zsite\_id,  s2.site\_num zsite\_num,  s2.full\_port\_speed zfull\_port\_speed,  s2.grc zgrc,  s2.protocol zprotocol,  s2.ta\_name zta\_name,  s2.site\_glbl\_dlci zsite\_glbl\_dlci,  s2.clli zclli,  ca2.acc\_ckt zacc\_ckt,  pa2.port zport,  pa2.port\_speed zport\_speed,  p2.loc\_id zloc\_id,  p2.prem\_address zprem\_address,  p2.prem\_city zprem\_city,  p2.prem\_state zprem\_state,  p2.prem\_country zprem\_country,  vpn.vpn\_name,  vpn.vpn\_id,  **--//A and Z ends VPNs are different when PVC type is IPVC, or PNPVC.**  **--//VPN1 go with PVC local side, VPN2 go with remote side.**  vpn1.vpn\_id AS aVpn\_id,  vpn1.vpn\_name AS aVpn\_name,  vpnCust1.cust\_id AS a\_Vpn\_Cust\_id  vpnCust1.mcn\_name AS aVpn\_mcn  vpn2.vpn\_id AS zVpn\_id,  vpn2.vpn\_name AS zVpn\_name,  vpnCust2.cust\_id AS z\_Vpn\_Cust\_id  vpnCust2.mcn\_name AS zVpn\_mcn  ~~cvpn.cust\_name avpn\_name,~~  ~~cvpn.cust\_mcn avpn\_mcn,~~  ~~cvpn.cust\_name zvpn\_name,~~  ~~cvpn.cust\_mcn zvpn\_mcn,~~  ~~cvpn.cust\_id a\_vpn\_cust\_id,~~  ~~cvpn.cust\_id z\_vpn\_cust\_id~~  **FROM**  premise p2,  port\_asgmt pa2,  cust\_access ca2,  site s2,  customer c2,  premise,  port\_asgmt,  cust\_access,  site,  customer,  pvc,  equipment,  ~~ipfr,~~  ~~vpn,~~  ~~customer cvpn~~  ipfr ipfe1,  ipfr ipfe2,  vpn vpn1,  vpn vpn2,  inter\_vpn\_pvc,  customer vpnCust1,  customer vpnCust2  **WHERE** upper(equipment.equip\_name) = <equipName>  AND ((pvc.lequip\_id = equipment.equip\_id  AND (pvc.lport = <input port>  AND pvc.lslot = <input slot>  AND pvc.lshelf = <input shelf>)  )  **OR** (pvc.requip\_id = equipment.equip\_id  AND (pvc.rport = <input port>  AND pvc.rslot = <input slot>  AND pvc.rshelf = <input shelf>)  )  )  AND customer.cust\_id = pvc.pvc\_lcust\_id  AND site.site\_id = pvc.pvc\_lsite\_id  AND cust\_access.site\_id = site.site\_id  AND port\_asgmt.site\_id = site.site\_id  AND premise.loc\_id = site.prem\_loc\_id  AND c2.cust\_id = pvc.pvc\_rcust\_id  AND s2.site\_id = pvc.pvc\_rsite\_id  AND ca2.site\_id = s2.site\_id  AND pa2.site\_id = s2.site\_id  AND p2.loc\_id = s2.prem\_loc\_id  ~~AND ipfr.pvc\_id = pvc.pvc\_id~~  ~~AND ipfr.vpn\_id = vpn.vpn\_id~~  ~~AND vpn.cust\_id = cvpn.cust\_id~~  And pvc.pvc\_type IN ( 'IPVC', 'PNPVC')  And pvc.pvc\_id = inter\_vpn\_pvc.pvc\_id  And inter\_vpn\_pvc.vpn\_id1 = vpn1.vpn\_id  And inter\_vpn\_pvc.vpn\_id2 = vpn2.vpn\_id  And ipfr1.pvc\_id = pvc.pvc\_id  And ipfr2.pvc\_id = pvc.pvc\_id  And ipfr1.vpn\_id = vpn1.vpn\_id  And ipfr2.vpn\_id = vpn2.vpn\_id  And vpn1.cust\_id = vpnCust1.cust\_id  And vpn2.cust\_id = vpnCust2.cust\_id | |
|  | | **For each PVC record, Derive TypeOfService data**  If SITEA.protocol is ‘FR’ (for a\_typeOfService) Or SITEZ.protocol is ‘FR’ (for z\_typeOfService)  If PREMISEA.prem\_country is ‘USA’ (for a\_typeOfService) Or PREMISEZ.prem\_country is ‘USA’ (for z\_typeOfService)  Set TypeOfService = ‘FR’  Else  Set TypeOfService = ‘IFR’  EndIf  EndIf  If SITEA.protocol is ‘ATM’ (for a\_typeOfService) Or SITEZ.protocol is ‘ATM’ (for z\_typeOfService)  If PREMISEA.prem\_country is ‘USA’ (for a\_typeOfService) Or PREMISEZ.prem\_country is ‘USA’ (for z\_typeOfService)  Set TypeOfService = ‘ATM’  Else  Set TypeOfService = ‘IATM’  EndIf  EndIf  If SITEA.protocol is not ‘ATM’ or ‘FR’, Then  Set ATypeOfService = SITEA.protocol  EndIf  If SITEZ.protocol is not ‘ATM’ or ‘FR’, Then  Set ZTypeOfService = SITEZ.protocol  EndIf  Retrieve all records from ICORE tables as described in “FR-ATM query response” | |
| 4.a) | | For PVC records, Retrieve from GCP SIDBOR database using ICORE acc\_ckt data from step 3.a  <288315> Adding new fields for both A and Z ends  managedIndicator  parentOrgGroup  Input: <acc\_ckt>  **For PVC A end:**  Select  FWTOPOLOGY.local\_router,  ~~ASSETA.src\_org\_id,~~  ASSETA.org\_cd, <288315>  ASSETA.managing\_org\_cd,  ASSETA.active\_org\_cd,  ASSETA.functional\_area,  ~~ASSETA.service,~~  Decode (Upper(ASSETA.service), <288315-US681337-upd4>  ‘IPEFX’, ‘BVOIP’,  ‘IPEFR’, ‘IPFR’,  ‘FRAME’, ‘FR’,  ASSETA.service) AS service,  CLIENT\_ORG.org\_cd\_prnt AS parentOrgGroup, <288315>  ‘Y’ managedIndicator, <288315>  From  FWTOPOLOGY,  ASSET ASSETA,  CLIENT\_ORG  Where FWTOPOLOGY.local\_access\_ckt = <acc\_ckt>  And FWTOPOLOGY.functional\_area = ASSETA.functional\_area  And FWTOPOLOGY.local\_router = ASSETA.asset\_nm  And ASSETA.org\_cd = CLIENT\_ORG.org\_cd  And ASSETA.functional\_area = CLIENT\_ORG.functional\_area  If data found  Set A CERName = FWTOPOLOGYA.local\_router  ~~Set A OrgGroup = ASSETA.src\_org\_id~~  Set A OrgGroup = ASSETA.org\_cd, <288315>  Set A ManageOrg = ASSETA. managing\_org\_cd  Set A ActiveOrg = ASSETA. active\_org\_cd  Set A FUNCTIONALAREA= ASSETA. functional\_area  Set A Service = ASSETA. Service  Set A parentOrgGroup = blank <288315>  Set A managedIndocator = managedIndicator <288315>  **For PVC Z end:**  Select  FWTOPOLOGY.local\_router,  ~~ASSETZ.src\_org\_id,~~  ASSETZ.org\_cd, <288315>  ASSETZ.managing\_org\_cd,  ASSETZ.active\_org\_cd,  ASSETZ.functional\_area,  ~~ASSETZ.service,~~  Decode (Upper(ASSETZ.service), <288315-US681337-upd4>  ‘IPEFX’, ‘BVOIP’,  ‘IPEFR’, ‘IPFR’,  ‘FRAME’, ‘FR’,  ASSETZ.service) AS service,  CLIENT\_ORG.org\_cd\_prnt AS parentOrgGroup, <288315>  ‘Y’ managedIndicator, <288315>  From  FWTOPOLOGY,  ASSET ASSETZ  CLIENT\_ORG  Where FWTOPOLOGY.local\_access\_ckt = <acc\_ckt>  And FWTOPOLOGY.functional\_area = ASSETZ.functional\_area  And FWTOPOLOGY.local\_router = ASSETZ.asset\_nm  And ASSETZ.org\_cd = CLIENT\_ORG.org\_cd  And ASSETZ.functional\_area = CLIENT\_ORG.functional\_area  If data found  Set A CERName = FWTOPOLOGYA.local\_router  ~~Set Z OrgGroup = ASSETZ.src\_org\_id~~  Set Z OrgGroup = ASSETZ.org\_cd, <288315>  Set Z ManageOrg = ASSETZ.managing\_org\_cd  Set Z ActiveOrg = ASSETZ.active\_org\_cd  Set Z FUNCTIONALAREA= ASSETZ.functional\_area  Set Z Service = ASSETZ. service  Set Z parentOrgGroup = blank <288315>  Set Z managedIndocator = managedIndicator <288315> | |
| 4.b) | | NOTE: this step is commented out until further development with CCI  Retrieve from GCP(DBOR) CCI database using ICORE customer\_mcn data from step 3.a  Match CUSTOMERA.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  Match CUSTOMERZ.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  and CUSTOMERA.cust\_mcn (pos. 7-9) =MCN\_ENTRPRS\_XREF.MCN\_SFX  and CUSTOMERZ.cust\_mcn (pos. 7-9) =MCN\_ENTRPRS\_XREF.MCN\_SFX    and XREF\_MCN\_ENTRPRS.ENT\_PTY\_ID = PARTY.ENTRPRS\_PTY\_ID  and PARTY.DUNS\_NB = DNB.DUNS\_NB  Retrieve DNB.(DUNS\_NB, PRNTHQ\_BIZ\_NM)  Set CCI = DUNS\_NB  Set PARENT NAME = PRNTHQ\_BIZ\_NM  Else  IF CUSTOMER1.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  CUSTOMER2.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  and XREF\_MCN\_ENTRPRS.ENT\_PTY\_ID = PARTY.ENTRPRS\_PTY\_ID  and PARTY.DUNS\_NB = DNB.DUNS\_NB  Retrieve DNB.(DUNS\_NB, PRNTHQ\_BIZ\_NM)  Set CCI = DUNS\_NB  Set PARENT NAME = PRNTHQ\_BIZ\_NM    Else  Set CCI = blank  Set PARENT NAME = blank  EndIf  EndIf | |
| 5. | | If AOTS.SERVICE TYPE is ‘FR-CNT’  Return record count of all data found in step 3.a and 3.b as described in “FR-ATM Record Count response”  Else  Return all records as described in “FR-ATM query response” .  EndIf | |
| 5.  a | | Create Record Selector for each of the records retrieved in steps 3.a,3.b,3.c, starting with number 1  Set Customer\_Record\_Selector = Record Selector | |
| 5.  b | | | Create return message to AOTS as described in “FRM-ATM QUERY RESPONSE” using data from steps 3.a,3.b,3.c.  If Input\_ change\_id & cmdc\_Trans\_ID are populated then  Insert retrieved FRM-ATM data from steps 3a, 3b, 3c into table “ACMR5\_CNC\_ALLCUSTOMERSAFFECTED”  Set input data as:   * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.REQUEST\_ID = Input\_change\_id * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.cmdc\_transid = Input\_cmdc\_TransID * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.customer\_record\_selector = customer\_record\_selector * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Customer\_ID = Customer ID * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Customer\_Name = CustomerName * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Customer\_MCN = CustomerMCN * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Location\_ID = LocationID * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Site\_ID = Site Number * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Site\_Protocol = SiteProtocol * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Site\_TA = Site TA * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Global\_DLCI = GlobalDLCI * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Circuit\_ID = Circuit ID * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.type\_of\_service = TypeOfService * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Port = Port * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Slot = Slot * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Shelf = Shelf—**Added for defect ID 230097 for PID 297409** * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Address = Address * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.City = City * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.State = State * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Country = Country * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.CLLI = clli * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Port\_Speed = PortSpeed * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.service\_line~~code~~ = Service~~Code~~ * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.CER\_name = CER name * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Org\_Group = OrgGroup * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Managing\_Org = ManageOrg * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Active\_Org = ActiveOrg * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Functional\_Area = FunctionalArea * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.CCI = CCI * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Parent\_Company = ParentCompanyName * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Circuit\_bmp\_format = circuitBmpFormat <288315> * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Managed\_indicator = managedIndicator <288315> * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Bvoip\_indicator = bvoidIndicator <288315> * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Parent\_org\_group = parentOrgGroup <288315> * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Request\_equip\_shelf = requestEquipShelf <297409.173072> * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Request\_equip\_slot = requestEquipSlot <297409.173072> * ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Request\_equip\_port = requestEquipPort <297409.173072>   **<288315>**  Insert retrieved CustomerAffected data, same attributes as shown above, to GCP\_cnc\_AllCustomersAffected  BAU:  After all the customer records inserted for this transaction, calculate the total number of records been inserted:  Set CustomerRecordCnt = last-record\_selector (Customer\_Record\_Selector)  **</288315>**  Else  Goto step 7  EndIf |
| ~~5.c~~ | | | ~~After the data is stored successfully, GCP call AOTS CM notification web service, sending them change\_id & transaction\_id as input parameters, for data that has been stored~~ |
| 6.A | | | Create a PVC\_Record\_Selector for each of the records retrieved from step 4, 4.a & 4.b for each change Request ID starting from 10000. |
| 6.b | | | Call ~~UC-DBOR-PB932-20~~ Common logic – Load\_Change\_PVC: To store retrieved PVC data in table Change\_PVC |
| 6.c | | | <288315> Generate new record and load or update the Summary table  Execute common logic: Load\_GCP\_AllCustomersAffected\_sum |
| 6.d | | | <288315> ETL:  Create CustomerRecord File  Create PVCRecord File |
| ~~5.~~ | | | ~~Provide all tables/columns described in “FR-ATM QUERY RESPONSE” in BO for reporting~~ |
| 7 | | | Use Case ends |

|  |  |
| --- | --- |
| POST-CONDITIONS | |
| Course | Description |
| 01 | DBOR ICORE data retrieved successfully |
| 02 | DBOR data returned to AOTS successfully |

|  |  |
| --- | --- |
| EXCEPTIONS | |
| Exception ID | Action(s) on Exception |
| 01 | Ideal Course Step 3 – GCP(DBOR) ICORE database not available for data retrieval. Create Error “DBOR\_001” |
| 02 | Ideal Course Step 4 – Retrieved data cannot be formatted to return to AOTS. Create Error “DBOR\_002” |

### InquireCustomerCircuitDetailsByNetworkElement for servicetype=FR-ATM-RPM – 288315

**Formally known as “GetCustInfoQuery for servicetype=FR-ATM-RPM”**

|  |  |
| --- | --- |
| **Query Name** | getCustInfoQuery for servicetype=FR-ATM-RPM |
| **PIDs** | 289116.140768, 288315 |
| **Service** | AVPN |
| **Data Source** | ICORE, SIDBOR, CCI |
| **WSDL File** | CmdcWebServicesContract.wsdl |
| **Client App** | AOTS CM |
| **Purpose, Usage** | FR-ATM-RPM Customer Inventory Retrieval .  This use case details the customer inventory request from AOTS for Frame ATM RPM data. FR-ATM-RPM query is to be created for project P9A78. GCP will retrieve data from GCP ICORE database. Satisfying P9A78 – Change Management Bundled features, requirement.  Input data combination can be: EQUIP\_NAME; EQUIP\_NAME + SLOT; |

**Query Change Summary**

|  |  |
| --- | --- |
| **Project /Ticket ID** | **Change Summary / Notes** |
| 6/12/2007  5.0 | Lan Tran-Vu:  Initial issue |
| 10/11/2007  5.3 | Lan Tran-Vu:  Take out CCI data. Change in step 3.c |
| 2/29/2008 | Gunjan Gupta, Rand Portus:  Adding PVC data |
| 289116.140768  July’2016 | US636605 - CR140768 - US GCP-AOTS-CM supporting uCPE for change management  Enhance for AOTS-CM to retrieve uCPE data when a PE is chosen as the input.  Dependency: The required data are available in the source systems --- A&AI, ~~GDB, CANOPI~~ |
| 288315  Oct’2016 | US681337 – Support AOTS CM ticket notification.  This enhancement includes  Add new fields to both Customer records and PVC records;  Change the process to Async, and the data retrieval logic is used by ETL to generate files. |
| 301033  June’2018 | US374866   * Re-engineer EDF logic to use the A&AI data sourced from DMaaP instead of the A&AI batch feed. * Support Vyatta uCPE. |

|  |  |
| --- | --- |
| USE CASE GENERAL INFORMATION | |
| Use Case Name | FR-ATM-RPM Customer Inventory Retrieval |
| Use Case ID | UC-DBOR-P9A78-018 |
| Description | This use case details the customer inventory request from AOTS for Frame ATM RPM data. FR-ATM-RPM query is to be created for project P9A78. DBOR will retrieve data from DBOR ICORE database. Satisfying P9A78 – Change Management Bundled features, requirement.  Input data combination can be: EQUIP\_NAME; EQUIP\_NAME + SLOT; |
| Responsible Analyst | Lan TranVu |
| Type of Execution | WebService and SQL to Oracle db. |

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE REVISION LOG | | | |
| Reason for Revision: | Take out CCI data. Change in step 3.c | Release Number: | 5.3 |
| Author: | Lan Tran-Vu | Revised: | 10/11/07 |
| Reason for Revision: | Initial Issue | Release Number: | 5.0 |
| Author: | Lan Tran-Vu | Revised: | 6/12/07 |
| Reason for Revision: | Add PVC Data | Release Number: |  |
| Author: | Gunjan Gupta, Rand Portus | Revised: | 02/29/08 |

**Output / Return Structure:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| FR-ATM-RPM Query Response | | | | | |
| Field | R/O/C | Data Type | Occur | Notes | ICORE/SIDBOR/CCI DB  TableName.elementName |
| Header | R | N/A | 1 | Begin Tag |  |
| Status\_Code | R | String | 1 | The status of the transaction.  Valid Values:  Success  Warning  Error |  |
| error Code | C | AN (10) | 1 | This is the error code.  This field is required if Status\_Code is Warning or Error. Refer to Error list in 4.3 section |  |
| Error Message | C | AN (250) | 1 | This is the error message. |  |
| Header | R | N/A | 1 | End Tag |  |
| Body | R | N/A | 1:N | Begin Tag |  |
| Customer Information Response | R | N/A | 0:N | Begin Tag |  |
| Recordcount | R | Int | 1 |  |  |
| Customer ID | R | Int | 1 | No trailing blanks | CUSTOMER.CUST\_ID |
| Customer Name | R | A24 | 1 | No trailing blanks | CUSTOMER.CUST\_NAME |
| Customer MCN | ~~R~~ O | A9 | 1 | No trailing blanks | CUSTOMER.CUST\_MCN |
| Location ID | ~~R~~ O | Int | 1 | No trailing blanks | PREMISE.LOC\_ID |
| Site Number | ~~R~~ O | Int | 1 | No trailing blanks | SITE.SITE\_NUM |
| Site Protocol | ~~R~~ O | A4 | 1 | No trailing blanks | SITE.PROTOCOL |
| Site TA | ~~R~~ O | A12 | 1 | No trailing blanks | SITE.TA\_NAME |
| Global DLCI | ~~R~~ O | Int | 1 | No trailing blanks | SITE.SITE\_GLBL\_DLCI |
| CircuitId | O | A50 | 1 |  | CUST\_ACCESS.ACC\_CKT  Note: This field has value directly retrieved from source DB, without format conversion.  seems BMP/normalized format being returned currently;  Check whether it’s converted already by GCP or ICORE.  Then determined whether new field is needed |
| circuitIdBmpFormat | O | A50 | 1 |  | <288315> Customer access circuit in BMP (normalized) format.  Deived value.  Note:  No conversion for International circuit. |
| TypeOfService | R | A7 | 1 |  | Derived |
| Port | O | Int | 1 | No trailing blanks | PORT\_ASGMT.PORT |
| Slot | O | Int | 1 | No trailing blanks | PORT\_ASGMT.SLOT |
| Address | ~~R~~ O | A50 | 1 | No trailing blanks | PREMISE.PREM\_ADDRESS |
| City | ~~R~~ O | A25 | 1 | No trailing blanks | PREMISE.PREM\_CITY |
| State | ~~R~~ O | A2 | 1 | No trailing blanks | PREMISE.PREM\_STATE |
| Country | ~~R~~ O | A20 | 1 | No trailing blanks | PREMIE.PREM\_COUNTRY |
| CLLI | O | A11 | 1 | No trailing blanks | SITE.CLLI |
| Port Speed | ~~R~~ O | Int | 1 | No trailing blanks | PORT\_ASGMT.PORT\_SPEED |
| CER Name | O | A50 | 1 | No trailing blanks | SIDBOR DB  FWTOPOLOGY.LOCAL\_ROUTER |
| managedIndicator | O | A5 | 1 | No trailing blanks | <288315> Values: Y, N, or null if no CPE exist |
| bvoipIndicator | O | A5 | 1 | No trailing blanks | <288315> Values: Y, N, or null |
| Service | O | A16 | 1 | No trailing blanks | SIDBOR DB  ASSET.SERVICE |
| OrgGroup | O | A11 | 1 | No trailing blanks | ~~ASSET.SRC\_ORG\_ID~~  ASSET.ORG\_CD |
| parentOrgGroup | O | A25 | 1 | No trailing blanks | <288315>  As of now after discussion, there is no current plan to use by CM for this project.  This may be used by EM. |
| ManageOrg | O | A11 | 1 | No trailing blanks | SIDBOR DB  ASSET.MANAGING\_ORG\_CD |
| ActiveOrg | O | A12 | 1 | No trailing blanks | SIDBOR DB  ASSET.ACTIVE\_ORG\_CD |
| FunctionalArea | O | A12 | 1 | No trailing blanks | SIDBOR DB  ASSET.functional\_area |
| RestrictionType | O | A30 | 1 |  | SIDBOR DB  CLIENT\_ORG.access\_type |
| CCI | O | A18 | 1 | Blanks | CCI DB.  DNB.DUNS\_NB |
| Parent Company Name | O | A90 |  | Blanks | CCI DB  DNB.PRNTHQ\_BIZ\_NM |
| <CpeDetailsList> | O | Complex  Array | 0:1 |  | <289116.140768-US636605>   * This list/array is to hold CPE records for this customer (customer access circuit). * For 1607 release, only uCPE data is supported. * See below for structure definition. |
| Customer Information Response | R | N/A |  | End Tag |  |
| Body | R | N/A | 1 | EndTag |  |

|  |  |
| --- | --- |
| IDEAL COURSE | |
| # | Step Description |
| 1. | Use Case begins |
| 2. | DBOR received query getCustInfoQuery request from AOTS with SERVICE TYPE = ‘FR-ATM-RPM’  For each record in <InputEquipmentList> <288315> Support list of input equipment names  Set Input\_Change ID = AOTS. Change ID  Set Input\_CmdcTransID = AOTS. CmdcTransID  Set Input\_ EquipName = AOTS.EquipName  If AOTS.PORT is populated and AOTS.SLOT is not populate  Set Input\_slot = SLOT from AOTS  EndIf |
| 3.a | Retrieve Customer records for input parameters, from ICORE (GCP replication) tables:  EQUIPMENT, CUSTOMER, CUST\_ACCESS, SITE, PREMISE, and PORT\_ASGMT  <288315> Add new fields  circuitIdBmpFormat  bvoipIndicator  **Note <288315>:**   * Access circuit in ICORE format has ‘.’ As delimiter to separate fields in the circuitID, as shown below. * The logic below checks these format and determine whether it’s ICORE format that can be convered to BMP format. For circuits other than these format will not get converted.   **DHEC.416129.200.ATI --‘.’ Delimiter starts at 5th position**  **23.HFFM.000433..SUV --‘.’ Delimiter starts at 3rd position**  Select  Decode (Cust\_access.acc\_ckt,  Instr (Cust\_access.acc\_ckt, ‘.’, 1, 1) = 3, ‘Y’,  Instr (Cust\_access.acc\_ckt, ‘.’, 1, 1) = 5, ‘Y’,  ‘N’ ) cktConversionFlag, <288315> Used to mark whether the circuit format can be converted.  CUSTOMER(cust\_id, cust\_name, cust\_mcn)  PREMISE(loc\_id, prem\_address, prem\_city, prem\_state, prem\_country)  SITE(clli, site\_num, protocol, ta\_name, site\_glbl\_dlci, site\_id)  PORT\_ASGMT(slot, port, port\_speed)  CUST\_ACCESS(acc\_ckt)  From  EQUIPMENT,  CUSTOMER,  CUST\_ACCESS,  SITE,  PREMISE,  PORT\_ASGMT  Match EQUIPMENT.equip\_id = PORT\_ASGMT.equip\_id  And PORT\_ASGMT.site\_id = SITE.site\_id  And CUSTOMER.cust\_id = CUST\_ACCESS.cust\_id  And CUST\_ACCESS.site\_id = SITE.site\_id  And CUST\_ACCESS.cust\_id = SITE.cust\_id  And SITE.prem\_loc\_id = PREMISE.loc\_id  And SITE.loc\_id = PREMISE.loc\_id  And EQUIPMENT.equip\_name = <Input\_EquipName>  And PORT\_ASGMT.slot = <Input\_slot>  Derive TypeOfService data  If SITE.protocol is ‘FR’  If PREMISE.prem\_country is ‘USA’  Set TypeOfService = ‘FR’  Else  Set TypeOfService = ‘IFR’  EndIf  EndIf  If SITE.protocol is ‘ATM’  If PREMISE.prem\_country is ‘USA’  Set TypeOfService = ‘ATM’  Else  Set TypeOfService = ‘IATM’  EndIf  EndIf  If SITE.protocol is not ‘ATM’ or ‘FR’  Set TypeOfService = SITE.protocol  EndIf  <289116.140768-US636605>  call GetUcpeByCircuitID (cust\_access.acc\_ckt) and populate CpeDetailsList  <288315> Derive bvoipIndiactor from ICORE data – For each record retrieved above:  Select service\_option.\*  From  service\_asgmt,  service\_option  Where service\_asgmt.site\_id = <site\_id>  And service\_asgmt.cust\_id = <cust\_id>  And service\_asgmt.serv\_opt\_id = service\_option.serv\_opt\_id  And upper(service\_option.serv\_opt) = ‘BVOIP’  If this returned data, then  Set bvoipIndicator = ‘Y’,  Else  Set bvoipIndicator = ‘N’,  End If  <288315>  Convert the acc\_ckt (in ICORE format) to normalized/BMP format.  This is don’t only for domestic circuit. For international circuit, leave it as is.  If cktConversionFlag = Y, then  Call Common logic –CircuitFormatConversion\_ICORE\_to\_BMP (<acc\_ckt>) -> circuitIdBmpFormat  Else  Set circuitIdBmpFormat = <acc\_ckt>  End If |
| 3.b | For each Customer record, retrieve data from GCP **SIDBOR** database using ICORE acc\_ckt data from step 3.a  **Note:** SiDBOR contains only managed devices. So if data is retrieved, then it’s managed. Otherwise (no CER Name), then leave this field blank.  <288315> Add new fields  managedIndicator  parentOrgGroup  Select  FWTOPOLOGY.local\_router AS cerName,  ~~ASSET.src\_org\_id,~~  ASSET.org\_cd AS orgGroup, <288315>  ASSET.managing\_org\_cd AS manageOrg,  ASSET.active\_org\_cd AS activeOrg,  ASSET.functional\_area AS functionalArea,  Decode (Upper(ASSET.service), <288315-US681337-upd3>  ‘IPEFX’, ‘BVOIP’,  ‘IPEFR’, ‘IPFR’,  ‘FRAME’, ‘FR’,  ASSET.service) AS service,  CLIENT\_ORG.access\_type,  CLIENT\_ORG.org\_cd\_prnt AS parentOrgGroup, <288315>  ‘Y’ managedIndicator <288315>  From  FWTOPOLOGY, --from FrameWorks (sourced from ICORE, NC3, …)  ASSET,  CLIENT\_ORG  Where 1=1  And FWTOPOLOGY.local\_access\_ckt = <acc\_ckt>  And FWTOPOLOGY.functional\_area = ASSET.functional\_area  And FWTOPOLOGY.local\_router = ASSET.asset\_nm  And ASSET.org\_cd = CLIENT\_ORG.org\_cd  And ASSET.functional\_area = CLIENT\_ORG.functional\_area  If no data found  Set CERName = blank  Set OrgGroup = blank  Set ManageOrg = blank  Set ActiveOrg = blank  Set FUNCTIONALAREA= blank  Set Service = blank  Set access\_type = blank  Set parentOrgGroup = blank <288315>  Set managedIndicator = blank <288315>  End If |
| 3.c | Document v 5.3 NOTE: this step needs to be commented out until further development with CCI  Retrieve from DBOR CCI database using ICORE customer\_mcn data from step 3.a  Match CUSTOMER.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  and CUSTOMER.cust\_mcn (pos. 7-9) =  MCN\_ENTRPRS\_XREF.MCN\_SFX  and XREF\_MCN\_ENTRPRS.ENT\_PTY\_ID = PARTY.ENTRPRS\_PTY\_ID  and PARTY.DUNS\_NB = DNB.DUNS\_NB  Retrieve DNB.(DUNS\_NB, PRNTHQ\_BIZ\_NM)  Set CCI = DUNS\_NB  Set PARENT NAME = PRNTHQ\_BIZ\_NM  Else  IF CUSTOMER.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  and XREF\_MCN\_ENTRPRS.ENT\_PTY\_ID = PARTY.ENTRPRS\_PTY\_ID  and PARTY.DUNS\_NB = DNB.DUNS\_NB  Retrieve DNB.(DUNS\_NB, PRNTHQ\_BIZ\_NM)  Set CCI = DUNS\_NB  Set PARENT NAME = PRNTHQ\_BIZ\_NM    Else  Set CCI = blank  Set PARENT NAME = blank  EndIf  EndIf |
| 4 | **Rtrieve PVC records from ICORE (GCP replication) tables**:  <288315> Adding new fields for both A and Z ends  managedIndicator  circuitIdBmpFormat  parentOrgGroup  Select  PVC.pvc\_id,  PVC.rSwitch, PVC.rSlot, PVC.rPort, PVC.pvc\_rdlci, PVC.pvc\_rontrcir, PVC.pvc\_rVci, PVC.pvc\_rVpi,  PVC.lSwitch, PVC.lSlot, PVC.lPort, PVC.pvc\_ldlci, PVC.pvc\_lontrcir, PVC.pvc\_lVci, PVC.pvc\_lVpi,  VPNA.vpn\_name, VPNZ.vpn\_name,  IPFRA.vpn\_id, IPFRZ.vpn\_id  SiteA.site\_id, SiteA.full\_port\_speed, SiteA.grc, SiteA.Clli  SiteZ.site\_id, SiteZ.full\_port\_speed, SiteZ.grc, SiteZ.Clli  CustomerA.cust\_id, CustomerA.cust\_name, CustomerA.cust\_mcn,  CustomerZ.cust\_id, CustomerZ.cust\_name, CustomerZ.cust\_mcn,  PremiseA.loc\_id, PremiseA.prem\_address, PremiseA.prem\_city, PremiseA.prem\_state, PremiseA.prem\_country,  PremiseZ.loc\_id, PremiseZ.prem\_address, PremiseZ.prem\_city, PremiseZ.prem\_state, PremiseZ.prem\_country,  Cust\_accessA.acc\_ckt,  Cust\_accessZ.acc\_ckt,  Decode (Cust\_accessA.acc\_ckt,  Instr (Cust\_accessA.acc\_ckt, ‘.’, 1, 1) = 3, ‘Y’,  Instr (Cust\_accessA.acc\_ckt, ‘.’, 1, 1) = 5, ‘Y’,  ‘N’ ) cktConversionFlagA, <288315> Used to mark whether the circuit format can be converted.  Decode (Cust\_accessZ.acc\_ckt,  Instr (Cust\_accessZ.acc\_ckt, ‘.’, 1, 1) = 3, ‘Y’,  Instr (Cust\_accessZ.acc\_ckt, ‘.’, 1, 1) = 5, ‘Y’,  ‘N’ ) cktConversionFlagZ, <288315> Used to mark whether the circuit format can be converted.  From  PVC,  IPFR IpfrA, IPFR IpfrZ,  VPN vpnA, VPN vpnZ,  Customer CustomerA, Customer CustomerZ,  Cust\_access Cust\_accessA, Cust\_access Cust\_accessZ,  SITE SiteA, SITE SiteZ,  PREMISE PremiseA, PRIMESE PremiseZ,  PORT\_ASGMT,  EQUIPMENT  Match  SITE.site\_id = PORT\_ASGMT.site\_id(+) And PORT\_ASGMT.site\_id = CUST\_ACCESS.site\_id(+) And PORT\_ASGMT.equip\_id = EQUIPMENT.equip\_id(+)  And SITEZ.prem\_loc\_id = PREMISEZ.loc\_id(+) And SITEA.prem\_loc\_id = PREMISEA.loc\_id And SITEZ.site\_id = PVC.pvc\_rsite\_id(+) And SITEA.site\_id = PVC.pvc\_lsite\_id And PVC.pvc\_id = IPFRA.pvc\_id(+)  And PVC.pvc\_id = IPFRZ.pvc\_id(+)  And IPFRA.vpn\_id = VPNA.vpn\_id  And IPFRZ.vpn\_id = VPNZ.vpn\_id  And PVC.pvc\_rcust\_id = customerZ.cust\_id  And PVC.pvc\_lcust\_id = customerA.cust\_id  And CUST\_ACCESSZ.cust\_id = CUSTOMERZ.cust\_id(+)  And CUST\_ACCESSA.cust\_id = CUSTOMERA.cust\_id(+)  ~~And SITEA.site\_id = CUST\_PREM\_EQUIPA.~~  ~~And SITEZ.site\_id = CUST\_PREM\_EQUIPZ.~~  And EQUIPMENT.equip\_name = Input\_EquipName  And PORT\_ASGMT.port = Input\_port (only if Input\_port is populated)  And PORT\_ASGMT.slot = Input\_slot (only if Input\_slot is populated)  <288315>  Convert the acc\_ckt (in ICORE format) to normalized/BMP format, based on the cktConversionFlag value.  This is only for domestic circuit. For international circuit, leave it as is.  If cktConversionFlagA = Y, then  Call Common logic –CircuitFormatConversion\_ICORE\_to\_BMP (<A\_acc\_ckt>) -> A\_circuitIdBmpFormat  Else  Set A\_circuitIdBmpFormat = <A\_acc\_ckt>  End If  If cktConversionFlagZ = Y, then  Call Common logic –CircuitFormatConversion\_ICORE\_to\_BMP (<Z\_acc\_ckt>) -> Z\_circuitIdBmpFormat  Else  Set Z\_circuitIdBmpFormat = <Z\_acc\_ckt>  End If |
|  | **For each PVC record, Derive TypeOfService data**  If SITEA.protocol is ‘FR’ (for a\_typeOfService) Or SITEZ.protocol is ‘FR’ (for z\_typeOfService)  If PREMISEA.prem\_country is ‘USA’ (for a\_typeOfService) Or PREMISEZ.prem\_country is ‘USA’ (for z\_typeOfService)  Set TypeOfService = ‘FR’  Else  Set TypeOfService = ‘IFR’  EndIf  EndIf  If SITEA.protocol is ‘ATM’ (for a\_typeOfService) Or SITEZ.protocol is ‘ATM’ (for z\_typeOfService)  If PREMISEA.prem\_country is ‘USA’ (for a\_typeOfService) Or PREMISEZ.prem\_country is ‘USA’ (for z\_typeOfService)  Set TypeOfService = ‘ATM’  Else  Set TypeOfService = ‘IATM’  EndIf  EndIf  If SITEA.protocol is not ‘ATM’ or ‘FR’, Then  Set ATypeOfService = SITEA.protocol  EndIf  If SITEZ.protocol is not ‘ATM’ or ‘FR’, Then  Set ZTypeOfService = SITEZ.protocol  EndIf  Retrieve all records from ICORE tables as described in “FR-ATM query response” |
| 4.a) | For PVC records, Retrieve from GCP SIDBOR database using ICORE acc\_ckt data from step 3.a  **Input: <acc\_ckt>**  <288315> Adding new fields for both A and Z ends  managedIndicator  parentOrgGroup  **For PVC A end:**  Select  FWTOPOLOGY.local\_router,  ~~ASSETA.src\_org\_id,~~  ASSETA.org\_cd, <288315>  ASSETA.managing\_org\_cd,  ASSETA.active\_org\_cd,  ASSETA.functional\_area,  ~~ASSETA.service,~~  Decode (Upper(ASSETA.service), <288315-US681337-upd4>  ‘IPEFX’, ‘BVOIP’,  ‘IPEFR’, ‘IPFR’,  ‘FRAME’, ‘FR’,  ASSETA.service) AS service,  CLIENT\_ORG.org\_cd\_prnt AS parentOrgGroup, <288315>  ‘Y’ managedIndicator, <288315>  From  FWTOPOLOGY,  ASSET ASSETA  CLIENT\_ORG  Where FWTOPOLOGY.local\_access\_ckt = <acc\_ckt>  And FWTOPOLOGY.functional\_area = ASSETA.functional\_area  And FWTOPOLOGY.local\_router = ASSETA.asset\_nm  And ASSETA.org\_cd = CLIENT\_ORG.org\_cd  And ASSETA.functional\_area = CLIENT\_ORG.functional\_area  If data found  Set A CERName = FWTOPOLOGYA.local\_router  ~~Set A OrgGroup = ASSETA.src\_org\_id~~  Set A OrgGroup = ASSETA.org\_cd, <288315>  Set A ManageOrg = ASSETA. managing\_org\_cd  Set A ActiveOrg = ASSETA. active\_org\_cd  Set A FUNCTIONALAREA= ASSETA. functional\_area  Set A Service = ASSETA. Service  Set A parentOrgGroup = blank <288315>  Set A managedIndocator = managedIndicator <288315>  **For PVC Z end:**  Select  FWTOPOLOGY.local\_router,  ~~ASSETZ.src\_org\_id,~~  ASSETZ.org\_cd, <288315>  ASSETZ.managing\_org\_cd,  ASSETZ.active\_org\_cd,  ASSETZ.functional\_area,  ~~ASSETZ.service,~~  Decode (Upper(ASSETZ.service), <288315-US681337-upd4>  ‘IPEFX’, ‘BVOIP’,  ‘IPEFR’, ‘IPFR’,  ‘FRAME’, ‘FR’,  ASSETZ.service) AS service,  CLIENT\_ORG.org\_cd\_prnt AS parentOrgGroup, <288315>  ‘Y’ managedIndicator, <288315>  From  FWTOPOLOGY,  ASSET ASSETZ  CLIENT\_ORG  Where FWTOPOLOGY.local\_access\_ckt = <acc\_ckt>  And FWTOPOLOGY.functional\_area = ASSETZ.functional\_area  And FWTOPOLOGY.local\_router = ASSETZ.asset\_nm  And ASSETZ.org\_cd = CLIENT\_ORG.org\_cd  And ASSETZ.functional\_area = CLIENT\_ORG.functional\_area  If data found  Set A CERName = FWTOPOLOGYA.local\_router  ~~Set Z OrgGroup = ASSETZ.src\_org\_id~~  Set Z OrgGroup = ASSETZ.org\_cd, <288315>  Set Z ManageOrg = ASSETZ.managing\_org\_cd  Set Z ActiveOrg = ASSETZ.active\_org\_cd  Set Z FUNCTIONALAREA= ASSETZ.functional\_area  Set Z Service = ASSETZ. service  Set Z parentOrgGroup = blank <288315>  Set Z managedIndocator = managedIndicator <288315> |
| 4.b) | NOTE: this step is commented out until further development with CCI  Retrieve from GCP(DBOR) CCI database using ICORE customer\_mcn data from step 3.a  Match CUSTOMERA.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  Match CUSTOMERZ.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  and CUSTOMERA.cust\_mcn (pos. 7-9) =MCN\_ENTRPRS\_XREF.MCN\_SFX  and CUSTOMERZ.cust\_mcn (pos. 7-9) =MCN\_ENTRPRS\_XREF.MCN\_SFX    and XREF\_MCN\_ENTRPRS.ENT\_PTY\_ID = PARTY.ENTRPRS\_PTY\_ID  and PARTY.DUNS\_NB = DNB.DUNS\_NB  Retrieve DNB.(DUNS\_NB, PRNTHQ\_BIZ\_NM)  Set CCI = DUNS\_NB  Set PARENT NAME = PRNTHQ\_BIZ\_NM  Else  IF CUSTOMER1.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  CUSTOMER2.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  and XREF\_MCN\_ENTRPRS.ENT\_PTY\_ID = PARTY.ENTRPRS\_PTY\_ID  and PARTY.DUNS\_NB = DNB.DUNS\_NB  Retrieve DNB.(DUNS\_NB, PRNTHQ\_BIZ\_NM)  Set CCI = DUNS\_NB  Set PARENT NAME = PRNTHQ\_BIZ\_NM    Else  Set CCI = blank  Set PARENT NAME = blank  EndIf  EndIf |
| 5 | Create Record Selector for each of the records retrieved in steps 3.a,3.b,3.c, starting with number 1  Set Customer\_Record\_Selector = Record Selector |
| 6. | Create return message to AOTS for all records found in steps 3,4 as described in “FR-ATM query response”  Refer to AOTS-DBOR IAD document “WPID\_C3 13\_R14\_AOTSCM-DBOR\_IAD\_V14.04”  If Input\_ change\_id & cmdc\_Trans\_ID are populated then  Insert retrieved FR-ATM-RPM data from steps 3a, 3b, 3c into table “ACMR5\_CNC\_ALLCUSTOMERSAFFECTED”  Set input data as:  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.REQUEST\_ID = Input\_change\_id  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.cmdc\_transid = Input\_cmdc\_TransID  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.customer\_record\_selector = customer\_record\_selector  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Customer\_ID = Customer ID  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Customer\_Name = CustomerName  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Customer\_MCN = Customer MCN  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Location\_ID = LocationID  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Site\_ID = Site Number  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Site\_Protocol = Site Protocol  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Site\_TA = Site TA  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Global\_DLCI = Global DLCI  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Circuit\_ID = Circuit ID  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.type\_of\_service = TypeOfService  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Port = Port  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Slot = Slot  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Address = Address  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.City = City  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.State = State  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Country = Country  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.CLLI = clli  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Port\_Speed = PortSpeed  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.service\_line~~code~~ = Service~~Code~~  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.CER\_name = CER name  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Org\_Group = OrgGroup  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Managing\_Org = ManageOrg  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Active\_Org = ActiveOrg  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Functional\_Area = Functional Area  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.CCI = CCI  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Parent\_Company = ParentCompanyName  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Circuit\_bmp\_format = circuitBmpFormat  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Managed\_indicator = managedIndicator  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Bvoip\_indicator = bvoidIndicator  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Parent\_org\_group = parentOrgGroup <288315>  Else  Goto step 10  EndIf  **<288315>**  Insert retrieved CustomerAffected data, same attributes as shown above, to GCP\_cnc\_AllCustomersAffected  After all the customer records inserted for this transaction, calculate the total number of records been inserted:  Set CustomerRecordCnt = last-record\_selector (Customer\_Record\_Selector)  **</288315>** |
| 6-a | Create a PVC\_Record\_Selector for each of the records retrieved from step 4, 4.a & 4.b for each change Request ID starting from 10000.  <288315>  Call ~~UC-DBOR-PB932-20~~ Common logic – Load\_Change\_PVC: To store retrieved PVC data in table Change\_PVC |
| 6-b | <288315> Generate new record and load or update the Summary table  Execute common logic: Load\_GCP\_AllCustomersAffected\_sum |
| 6.c | <288315> ETL:  Create CustomerRecord File  Create PVCRecord File |
| 7 | After the data is stored successfully, GCP call AOTS CM notification web service, sending them change\_id & transaction\_id as input parameters, for data that has been stored |
| 8 | Create a PVC\_Record\_Selector for each of the records retrieved from step 4, 4.a & 4.b for each change Request ID starting from 10000. |
| ~~9~~ | ~~Call UC-DBOR-PB932-19: To store retrieved PVC data in table Change\_PVC~~ |
| 10 | Use Case ends |

|  |  |
| --- | --- |
| POST-CONDITIONS | |
| Course | Description |
| 01 | DBOR ICORE data retrieved successfully |
| 02 | DBOR data returned to AOTS successfully |

|  |  |
| --- | --- |
| EXCEPTIONS | |
| Exception ID | Action(s) on Exception |
| 01 | Ideal Course Step 3a – DBOR ICORE database not available for data retrieval. |
| 02 | Ideal Course Step 4 – Retrieved data cannot be formatted to return to AOTS. |

### InquireCustomerCircuitDetailsByNetworkElement for servicetype=AGN – 288315

**Formally known as “GetCustInfoQuery for servicetype=AGN”**

|  |  |
| --- | --- |
| **Query Name** | getCustInfoQuery for servicetype=AGN |
| **PIDs** | PA569, 289116.140768, 288315 |
| **Service** | AVPN |
| **Data Source** | ICORE, SIDBOR |
| **WSDL File** | CmdcWebServicesContract.wsdl |
| **Client App** | AOTS CM |
| **Purpose, Usage** | AGN Customer Inventory Retrieval.  This use case details the customer inventory request from AOTS for AGN data. GCP will retrieve data from GCP ICORE database. Satisfying P8635 Customer Notification Phase 3 BRD, requirement BRD\_Command\_and\_Control\_Ph3\_060.  Input data combination can be: EQUIP\_NAME; EQUIP\_NAME + SLOT; EQUIP\_NAME + SLOT + PORT.  Input data combination can be: EQUIP\_NAME; EQUIP\_NAME + SLOT; EQUIP\_NAME + SLOT + PORT. |

**Query Change Summary**

|  |  |
| --- | --- |
| **Project /Ticket ID** | **Change Summary / Notes** |
| 1/15/2007  3.0 | Lan Tran-Vu:  Initial Issue |
| 2/22/2007  4.0 | Lan Tran-Vu:  PA569 – add government restriction type in step 3.b |
| 2/27/2008 | Rand Potrus, Lan Tran-Vu:  Store Data in ACMR5\_CNC\_ALLCUSTOMERSAFFECTED Table |
| 2/27/2008 | Gunjan Gupta,Lan Tran-Vu:  Adding PVC data |
| 04/25/2012 | Pete Oswald :  AOTS Ticket 000000144028424  AOTS CM IBM IE 146879198  AGN service using getCustInfoQuery does not return end customers when Ethernet cards/ports are used in the device  (using shared Ethernet access arrangements)  Adds use of IPFR and VPN table for end customer name identification |
| 05/15/2012 | Pete Oswald :  AOTS Ticket 154474025  Certain routers (Juniper and Cisco) using FPC/PIC and SIP/Spa card arrangements return 5 digit slot value that needs to be converted for output |
| 289116.140768  July’2016 | US636605 – CR140768 – US GCP-AOTS-CM supporting uCPE for change management  Enhance for AOTS-CM to retrieve uCPE data when a PE is chosen as the input.  Dependency: The required data are available in the source systems --- A&AI, ~~GDB, CANOPI~~ |
| 288315  Oct’2016 | US681337 – Support AOTS CM ticket notification.  This enhancement includes  Add new fields to both Customer records and PVC records;  Change the process to Async, and the data retrieval logic is used by ETL to generate files. |
| 301033  June’2018 | US374866   * Re-engineer EDF logic to use the A&AI data sourced from DMaaP instead of the A&AI batch feed. * Support Vyatta uCPE. |

|  |  |
| --- | --- |
| USE CASE GENERAL INFORMATION | |
| Use Case Name | AGN Customer Inventory Retrieval |
| Use Case ID | UC-GCP-PB932-006 |
| Description | This use case details the customer inventory request from AOTS for AGN data. GCP(DBOR) will retrieve data from GCP(DBOR) ICORE database. Satisfying P8635 Customer Notification Phase 3 BRD, requirement BRD\_Command\_and\_Control\_Ph3\_060.  Input data combination can be: EQUIP\_NAME; EQUIP\_NAME + SLOT; EQUIP\_NAME + SLOT + PORT.  Data Source: ICORE, SIDBOR |
| Responsible Analyst | Lan TranVu |
| Type of Execution | WebService and SQL to Oracle db. |

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE REVISION LOG | | | |
| Reason for Revision: | PA569 – add government restriction type in step 3.b | Release Number | 4.0 |
| Author: | Lan Tran-Vu | Revised: | 2/22/07 |
| Reason for Revision: | P8A30 add Service | Release Number: | 3.0 |
| Author: | Lan Tran-Vu | Revised: | 1/15/07 |
| Reason for Revision: | Initial Issue | Release Number: |  |
| Author: | Lan Tran-Vu | Revised: |  |
| Reason for Revision: | Store Data in ACMR5\_CNC\_ALLCUSTOMERSAFFECTED Table | Release Number: | 02/27/08 |
| Author: | Rand Potrus, Lan Tran-Vu | Revised: |  |
| Reason for Revision: | Adding PVC data | Release Number: |  |
| Author: | Gunjan Gupta, Lan Tran-Vu | Revised: | 02/27/08 |
| Reason for Revision | AOTS ticket 144028424  Inquiry Failing when router appears in ICORE.slot\_map added logic to link slot\_map table and use value to query ICORE.PORT\_ASGMT.slot |  |  |
| Author | Pete Oswald | Revised | 11/22/11 |
| Reason for  Revision  V17 | AOTS Ticket 000000144028424  AOTS CM IBM IE 146879198  AGN service using getCustInfoQuery does not return end customers when Ethernet cards/ports are used in the device  (using shared Ethernet access arrangements)  Adds use of IPFR and VPN table for end customer name identification | Revised | 04/25/12 |
| Author | Pete Oswald | Revised | 04/25/12 |
| Reason for Revision  V18 | AOTS Ticket 154474025  Certain routers (Juniper and Cisco) using FPC/PIC and SIP/Spa card arrangements return 5 digit slot value that needs to be converted for output |  |  |
| Author | Pete Oswald | Revised | 05/15/12 |

**Output / Return Structure:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AGN Query Response | | | | | |
| Field | R/O/C | Data Type | Occur | Notes | ICORE database  queryName.elementName |
| Header | R | N/A | 1 | Begin Tag |  |
| Status | R | AN (3) | 1 | The status of the transaction.  Valid Values:  ‘0’ for Success  ‘1’ for Warning  2 for Backend system is down |  |
| error Code | C | AN (10) | 1 | This is the error code.  This field is required if statusCode =0 ‘1’, ‘2’ or ‘-1’.  Valid Values:  Refer to Appendix B |  |
| Error Message | C | AN (250) | 1 | This is the error message.  This field is required if statusCode =0 ‘1’, ‘2’  Valid Values:  Refer to Section 4.3 |  |
| Header | R | N/A | 1 | End Tag |  |
| Body | R | N/A | 1:N | Begin Tag |  |
| Customer Information Response | R | N/A | 0:N | Begin Tag |  |
| Customer\_Record\_Selector | R | Int | 3 | No trailing blanks | Record Selector |
| Customer ID | R | Int | 1 | No trailing blanks | CUSTOMER.CUST\_ID |
| Customer Name | R | A24 | 1 | No trailing blanks | CUSTOMER.CUST\_NAME |
| Customer MCN | ~~R~~ O | A9 | 1 | No trailing blanks | CUSTOMER.CUST\_MCN |
| Location ID | ~~R~~ O | Int | 1 | No trailing blanks | PREMISE.LOC\_ID |
| Site Number | ~~R~~ O | Int | 1 | No trailing blanks | SITE.SITE\_NUM |
| Site Protocol | ~~R~~ O | A4 | 1 | No trailing blanks | SITE.PROTOCOL |
| Site TA | ~~R~~ O | A12 | 1 | No trailing blanks | SITE.TA\_NAME |
| Global DLCI | ~~R~~ O | Int | 1 | No trailing blanks | SITE.SITE\_GLBL\_DLCI |
| Circuit ID | O | A50 | 1 |  | Note: This field has value directly retrieved from source DB, without format conversion.  seems BMP/normalized format being returned currently;  Check whether it’s converted already by GCP or ICORE.  Then determined whether new field is needed |
| circuitIdBmpFormat | O | A50 | 1 |  | <288315> Customer access circuit in BMP (normalized) format.  Deived value.  Note:  No conversion for International circuit. |
| TypeOfService | R | A7 | 1 | Fixed value | “AGN” - Derived |
| Port | O | Int | 1 | No trailing blanks | SITE.SITE\_ID |
| Router Slot | O | Int | 1 | No trailing blanks | PORT\_ASGMT.slot |
| A Location ID | ~~R~~ O | Int | 1 | No trailing blanks | PREMISE.LOC\_ID |
| A Address | ~~R~~ O | A50 | 1 | No trailing blanks | PREMISE.PREM\_ADDRESS |
| A City | ~~R~~ O | A25 | 1 | No trailing blanks | PREMISE.PREM\_CITY |
| A State | ~~R~~ O | A2 | 1 | No trailing blanks | PREMISE.PREM\_STATE |
| A Country | ~~R~~ O | A20 | 1 | No trailing blanks | PREMIE.PREM\_COUNTRY |
| A CLLI | O | A11 | 1 | No trailing blanks | SITE.CLLI |
| Port Speed | ~~R~~ O | Int | 1 | No trailing blanks | PORT\_ASGMT.PORT\_SPEED |
| OrgGroup | O | A12 | 1 | No trailing blanks | ~~ASSET.SRC\_ORG\_ID~~  ASSET.ORG\_CD |
| parentOrgGroup | O | A125 | 1 | No trailing blanks | <288315>  As of now after discussion, there is no current plan to use by CM for this project.  This may be used by EM. |
| ManageOrg | O | A11 | 1 | No trailing blanks | SIDBOR DB  ASSET.MANAGING\_ORG\_CD |
| ActiveOrg | O | A12 | 1 | No trailing blanks | SIDBOR DB  ASSET.ACTIVE\_ORG\_CD |
| FunctionalArea | O | A12 |  | No trailing blanks | SIDBOR DB  ASSET.FUNCTIONAL\_AREA |
| RestrictionType | O | A30 | 1 |  | SIDBOR DB  CLIENT\_ORG.access\_type |
| CER Name | O | A50 | 1 | No trailing blanks | SIDBOR DB  FWTOPOLOGY.LOCAL\_ROUTER |
| managedIndicator | O | A5 | 1 | Derived | <288315> Values: Y, N, or null if no CPE exist |
| bvoipIndicator | O | A5 | 1 | Derived | <288315> Values: Y, N, or null |
| CCI | O | A18 | 1 | No trailing blanks | CCI DB.  DNB.DUNS\_NB |
| Parent Company Name | O | A90 |  | No trailing blanks | CCI DB  DNB.PRNTHQ\_BIZ\_NM |
| <CpeDetailsList> | O | Complex  Array | 0:1 |  | <289116.140768-US636605>   * This list/array is to hold CPE records for this customer (customer access circuit). * For 1607 release, only uCPE data is supported. * See below for structure definition. |
| Customer Information Response | R | N/A |  | End Tag |  |
| Body | R | N/A | 1 | EndTag |  |

|  |  |
| --- | --- |
| IDEAL COURSE | |
| # | Step Description |
| 1. | Use Case begins |
| 2. | GCP/DBOR received query getCustInfoQuery request from AOTS CM with SERVICE TYPE = ‘AGN’ or ‘AGN-CNT’  For each record in <InputEquipmentList> <288315> Support list of input equipment names  If AOTS.PORT is populated and AOTS.SLOT is not populate  Set Input\_Change ID = AOTS. Change ID  Set Input\_CmdcTransID = AOTS. CmdcTransID  Set Input\_EquipName = AOTS.EQUIP\_NAME  If AOTS.SLOT and/or AOTS.PORT are populated  Set Input\_Slot = AOTS.SLOT  Set Input\_Port = AOTS.PORT  Else  EndIf |
| 3.a | Retrieve Customer records for input parameters, from ICORE (GCP replication) tables:  <288315> Add new fields  circuitIdBmpFormat  bvoipIndicator  **Note <288315>:**   * Access circuit in ICORE format has ‘.’ As delimiter to separate fields in the circuitID, as shown below. * The logic below checks these format and determine whether it’s ICORE format that can be convered to BMP format. For circuits other than these format will not get converted.   **DHEC.416129.200.ATI --‘.’ Delimiter starts at 5th position**  **23.HFFM.000433..SUV --‘.’ Delimiter starts at 3rd position**  Retrieve all records from ICORE tables as described in “AGN query response”  <288315> Add below element in Select clause:  Decode (Cust\_access.acc\_ckt,  Instr (Cust\_access.acc\_ckt, ‘.’, 1, 1) = 3, ‘Y’,  Instr (Cust\_access.acc\_ckt, ‘.’, 1, 1) = 5, ‘Y’,  ‘N’ ) cktConversionFlag, <288315> Used to mark whether the circuit format can be converted.  Retrieve data From  EQUIPMENT,  CUSTOMER,  CUST\_ACCESS,  SITE,  PREMISE,  PVC,  IPFR,  VPN  PORT\_ASGMT,  EQUIP\_MODEL,  SLOT\_MAP  Match EQUIPMENT.equip\_id = PVC.requip\_id  And PVC.lequip\_id = PORT\_ASGMT.equip\_id  And PORT\_ASGMT.site\_id = CUST\_ACCESS.site\_id  And CUSTOMER.cust\_id = CUST\_ACCESS.cust\_id  And PORT\_ASGMT.site\_id = SITE.site\_id  And SITE.prem\_loc\_id = PREMISE.loc\_id  And EQUIPMENT.equip\_name = <Input\_EquipName>  And PORT\_ASGMT.port = Input\_port (only if Input\_port is populated)  And PORT\_ASGMT.slot = Input\_slot (only if Input\_slot is populated)  <AOTS ticket 144028424>  # adds logic to get correct end customer name for Ethernet Cards  Join IPFR PVC  If PORT\_ASGMT.protocol = ‘ETH’  And PORT\_ASGMT.site\_id = PVC.pvc\_rsite\_id  And PVC.pvc\_id = IPFR.pvc\_id  And IPFR.vpn\_id = VPN.vpn\_id, VPN.cust\_id  And VPN.cust\_id = Customer.cust\_id gets CUSTOMER.cust\_name  </AOTS ticket 144028424>  <AOTS ticket AOTS ticket 154474025>  Added logic to change output slot value when ICORE.PORT\_ASGMT.slot > 10000  Match ICORE.PORT\_ASGMT.slot = ICORE.SLOT\_MAP.int\_slot  And ICORE.SLOT\_MAP.char\_slot = Output Router Slot  Example PORT\_ASGMT.slot = 10501  SLOT\_MAP.int\_slot = 10501  SLOT\_MAP.char\_slot = 5-1  5-1 returned to output  </AOTS ticket AOTS ticket 154474025>  <288315> Derive bvoipIndiactor from ICORE data – For each record retrieved above:  Select service\_option.\*  From  service\_asgmt,  service\_option  Where service\_asgmt.site\_id = <site\_id>  And service\_asgmt.cust\_id = <cust\_id>  And service\_asgmt.serv\_opt\_id = service\_option.serv\_opt\_id  And upper(service\_option.serv\_opt) = ‘BVOIP’  If this returned data, then  Set bvoipIndicator = ‘Y’,  Else  Set bvoipIndicator = ‘N’,  End If  <288315>  Convert the acc\_ckt (in ICORE format) to normalized/BMP format.  This is don’t only for domestic circuit. For international circuit, leave it as is.  If cktConversionFlag = Y, then  Call Common logic –CircuitFormatConversion\_ICORE\_to\_BMP (<acc\_ckt>) -> circuitIdBmpFormat  Else  Set circuitIdBmpFormat = <acc\_ckt>  End If  <289116.140768-US636605>  **For each Customer record retrieved above:**  call GetUcpeByCircuitID (cust\_access.acc\_ckt) and populate CpeDetailsList  ~~If no data found return error code ‘0000’~~ |
| 3.b | For each Customer record, retrieve data from GCP **SIDBOR** database using ICORE acc\_ckt data from step 3.a  **Note:** SiDBOR contains only managed devices. So if data is retrieved, then it’s managed. Otherwise (no CER Name), then leave this field blank.  <288315> Add new fields  managedIndicator  parentOrgGroup  **Input: <acc\_ckt> from ICORE, Step 3.a**  Select  FWTOPOLOGY.local\_router AS cerName,  ~~ASSET.src\_org\_id,~~  ASSET.org\_cd AS orgGroup, <288315>  ASSET.managing\_org\_cd AS manageOrg,  ASSET.active\_org\_cd AS activeOrg,  ASSET.functional\_area AS functionalArea,  Decode (Upper(ASSET.service), <288315-US681337-upd3>  ‘IPEFX’, ‘BVOIP’,  ‘IPEFR’, ‘IPFR’,  ‘FRAME’, ‘FR’,  ASSET.service) AS service,  CLIENT\_ORG.org\_cd\_prnt AS parentOrgGroup, <288315>  CLIENT\_ORG.access\_type,  ‘Y’ managedIndicator <288315>  From  FWTOPOLOGY, --from FrameWorks (sourced from ICORE, NC3, …)  ASSET,  CLIENT\_ORG  Where 1=1  And FWTOPOLOGY.local\_access\_ckt = <acc\_ckt>  And FWTOPOLOGY.functional\_area = ASSET.functional\_area  And FWTOPOLOGY.local\_router = ASSET.asset\_nm  And ASSET.org\_cd = CLIENT\_ORG.org\_cd  And ASSET.functional\_area = CLIENT\_ORG.functional\_area  If no data found  Set CERName = blank  Set OrgGroup = blank  Set ManageOrg = blank  Set ActiveOrg = blank  Set FunctionalArea = blank  Set Service = blank  Set RestrictionType = blank  Set managedIndicator = blank <288315>  End If |
| 3.c | Document v 5.3 NOTE: this step needs to be commented out until further development with CCI  **Input: <customer\_mcn> from ICORE, Step 3.a**  Retrieve from GCP(DBOR) CCI database using ICORE customer\_mcn data from step 3.a  Match CUSTOMER.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  and CUSTOMER.cust\_mcn (pos. 7-9) = MCN\_ENTRPRS\_XREF.MCN\_SFX  and XREF\_MCN\_ENTRPRS.ENT\_PTY\_ID = PARTY.ENTRPRS\_PTY\_ID  and PARTY.DUNS\_NB = DNB.DUNS\_NB  Retrieve DNB.(DUNS\_NB, PRNTHQ\_BIZ\_NM)  Set CCI = DUNS\_NB  Set PARENT NAME = PRNTHQ\_BIZ\_NM  Else  IF CUSTOMER.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  and XREF\_MCN\_ENTRPRS.ENT\_PTY\_ID = PARTY.ENTRPRS\_PTY\_ID  and PARTY.DUNS\_NB = DNB.DUNS\_NB  Retrieve DNB.(DUNS\_NB, PRNTHQ\_BIZ\_NM)  Set CCI = DUNS\_NB  Set PARENT NAME = PRNTHQ\_BIZ\_NM    Else  Set CCI = blank  Set PARENT NAME = blank  EndIf  EndIf |
| 4 | **Rtrieve PVC records from ICORE (GCP replication) tables**:  <288315> Adding new fields for both A and Z ends  circuitIdBmpFormat  Retrieve From **PVC**  Select  PVC.pvc\_id,  PVC.rSwitch, PVC.rSlot, PVC.rPort, PVC.pvc\_rdlci, PVC.pvc\_rontrcir, PVC.pvc\_rVci, PVC.pvc\_rVpi,  PVC.lSwitch, PVC.lSlot, PVC.lPort, PVC.pvc\_ldlci, PVC.pvc\_lontrcir, PVC.pvc\_lVci, PVC.pvc\_lVpi,  VPNA.vpn\_name, VPNZ.vpn\_name,  IPFRA.vpn\_id, IPFRZ.vpn\_id  SiteA.site\_id, SiteA.full\_port\_speed, SiteA.grc, SiteA.Clli  SiteZ.site\_id, SiteZ.full\_port\_speed, SiteZ.grc, SiteZ.Clli  CustomerA.cust\_id, CustomerA.cust\_name, CustomerA.cust\_mcn,  CustomerZ.cust\_id, CustomerZ.cust\_name, CustomerZ.cust\_mcn,  PremiseA.loc\_id, PremiseA.prem\_address, PremiseA.prem\_city, PremiseA.prem\_state, PremiseA.prem\_country,  PremiseZ.loc\_id, PremiseZ.prem\_address, PremiseZ.prem\_city, PremiseZ.prem\_state, PremiseZ.prem\_country,  Cust\_accessA.acc\_ckt,  Cust\_accessZ.acc\_ckt,  Decode (Cust\_accessA.acc\_ckt, <288315>  Instr (Cust\_accessA.acc\_ckt, ‘.’, 1, 1) = 3, ‘Y’,  Instr (Cust\_accessA.acc\_ckt, ‘.’, 1, 1) = 5, ‘Y’,  ‘N’ ) cktConversionFlagA, <288315> Used to mark whether the circuit format can be converted.  Decode (Cust\_accessZ.acc\_ckt,  Instr (Cust\_accessZ.acc\_ckt, ‘.’, 1, 1) = 3, ‘Y’,  Instr (Cust\_accessZ.acc\_ckt, ‘.’, 1, 1) = 5, ‘Y’,  ‘N’ ) cktConversionFlagZ, <288315> Used to mark whether the circuit format can be converted.  From  PVC,  IPFR IpfrA, IPFR IpfrZ,  VPN vpnA, VPN vpnZ,  Customer CustomerA, Customer CustomerZ,  Cust\_access Cust\_accessA, Cust\_access Cust\_accessZ,  SITE SiteA, SITE SiteZ,  PREMISE PremiseA, PRIMESE PremiseZ,  PORT\_ASGMT,  EQUIPMENT  Match  EQUIPMENT.equip\_id = PVC.requip\_id  And PORT\_ASGMT.equip\_id = PVC.lequip\_id  And SITEZ.prem\_loc\_id = PREMISEZ.loc\_id(+) And SITEA.prem\_loc\_id = PREMISEA.loc\_id And SITEZ.site\_id = PVC.pvc\_rsite\_id(+) And SITEA.site\_id = PVC.pvc\_lsite\_id And PVC.pvc\_id = IPFRA.pvc\_id(+)  And PVC.pvc\_id = IPFRZ.pvc\_id(+)  And IPFRA.vpn\_id = VPNA.vpn\_id  And IPFRZ.vpn\_id = VPNZ.vpn\_id  And PVC.pvc\_rcust\_id = customerZ.cust\_id  And PVC.pvc\_lcust\_id = customerA.cust\_id  And CUST\_ACCESSZ.cust\_id = CUSTOMERZ.cust\_id(+)  And CUST\_ACCESSA.cust\_id = CUSTOMERA.cust\_id(+)  ~~And SITEA.site\_id = CUST\_PREM\_EQUIPA.~~  ~~And SITEZ.site\_id = CUST\_PREM\_EQUIPZ.~~  And EQUIPMENT.equip\_name = <Input\_EquipName>  And PORT\_ASGMT.port = Input\_port (only if Input\_port is populated)  And PORT\_ASGMT.slot = Input\_slot (only if Input\_slot is populated)  <288315>  Convert the acc\_ckt (in ICORE format) to normalized/BMP format, based on the cktConversionFlag value.  This is only for domestic circuit. For international circuit, leave it as is.  If cktConversionFlagA = Y, then  Call Common logic –CircuitFormatConversion\_ICORE\_to\_BMP (<A\_acc\_ckt>) -> A\_circuitIdBmpFormat  Else  Set A\_circuitIdBmpFormat = <A\_acc\_ckt>  End If  If cktConversionFlagZ = Y, then  Call Common logic –CircuitFormatConversion\_ICORE\_to\_BMP (<Z\_acc\_ckt>) -> Z\_circuitIdBmpFormat  Else  Set Z\_circuitIdBmpFormat = <Z\_acc\_ckt>  End If |
|  | **<288315-US681337-upd3>**  **Production SQL code retrieving PVC records: ServiceType=AGN**  **Updated to fix production issue – duplicate PVC records, and incorrect CER IP and PE IP:**  SELECT DISTINCT vpn.cust\_id VPNCUSTID,  pvc.pvc\_id,  pvc.rswitch,  pvc.rslot,  pvc.rport,  pvc.pvc\_rdlci,  pvc.pvc\_rcontrcir,  pvc.pvc\_rvci,  pvc.pvc\_rvpi,  pvc.lswitch,  pvc.lslot,  pvc.lport,  pvc.pvc\_ldlci,  pvc.pvc\_lcontrcir,  pvc.pvc\_lvci,  pvc.pvc\_lvpi,  pvc.PVC\_TYPE,  customer.cust\_name ACUST\_NAME,  customer.cust\_id ACUST\_ID,  customer.cust\_mcn ACUST\_MCN,  site.site\_id ASITE\_ID,  site.site\_num ASITE\_NUM,  site.full\_port\_speed AFULL\_PORT\_SPEED,  site.grc AGRC,  site.protocol APROTOCOL,  site.ta\_name ATA\_NAME,  site.site\_glbl\_dlci ASITE\_GLBL\_DLCI,  site.clli ACLLI,  cust\_access.acc\_ckt AACC\_CKT,  port\_asgmt.port APORT,  port\_asgmt.port\_speed APORT\_SPEED,  premise.loc\_id ALOC\_ID,  premise.prem\_address APREM\_ADDRESS,  premise.prem\_city APREM\_CITY,  premise.prem\_state APREM\_STATE,  premise.prem\_country APREM\_COUNTRY,  --ipfr.cpe\_ip\_address A\_CUST\_IP\_ADDRESS,  --ipfr.per\_ip\_address Z\_CUST\_IP\_ADDRESS,  --// If A end has cust\_id=395, then the cust\_Ip is the PE IP; else, the Cust\_IP is the CE IP  Decode (customer.cust\_id, ‘395’, ipfr.per\_ip\_address, ipfr.cpe\_ip\_address) a\_cust\_ip\_address,  Decode (c2.cust\_id, ‘395’, ipfr.per\_ip\_address, ipfr.cpe\_ip\_address) z\_cust\_ip\_address,  ipfr.vpn\_id,  c2.cust\_name ZCUST\_NAME,  c2.cust\_id ZCUST\_ID,  c2.cust\_mcn ZCUST\_MCN,  s2.site\_id ZSITE\_ID,  s2.site\_num ZSITE\_NUM,  s2.full\_port\_speed ZFULL\_PORT\_SPEED,  s2.grc ZGRC,  s2.protocol ZPROTOCOL,  s2.ta\_name ZTA\_NAME,  s2.site\_glbl\_dlci ZSITE\_GLBL\_DLCI,  s2.clli ZCLLI,  ca2.acc\_ckt ZACC\_CKT,  pa2.port ZPORT,  pa2.port\_speed ZPORT\_SPEED,  p2.loc\_id ZLOC\_ID,  p2.prem\_address ZPREM\_ADDRESS,  p2.prem\_city ZPREM\_CITY,  p2.prem\_state ZPREM\_STATE,  p2.prem\_country ZPREM\_COUNTRY,  VPN.VPN\_NAME ,  VPN.VPN\_ID,  **--//A and Z ends use the same vpn customer data, okay when there’s only one VPN.**  cvpn.cust\_name avpn\_name,  cvpn.cust\_mcn avpn\_mcn,  cvpn.cust\_name zvpn\_name,  cvpn.cust\_mcn zvpn\_mcn,  cvpn.cust\_id a\_vpn\_cust\_id,  cvpn.cust\_id z\_vpn\_cust\_id  FROM premise p2,  port\_asgmt pa2,  cust\_access ca2,  site s2,  customer c2,  ipfr,  premise,  port\_asgmt,  cust\_access,  site,  customer,  pvc,  equipment,  VPN,  CUSTOMER CVPN  WHERE pvc.requip\_id = equipment.equip\_id  AND customer.cust\_id = pvc.pvc\_lcust\_id  **AND pvc.pvc\_lsite\_id = site.site\_id**  AND cust\_access.site\_id = site.site\_id  AND port\_asgmt.site\_id = site.site\_id  AND premise.loc\_id = site.prem\_loc\_id  AND ipfr.pvc\_id = pvc.pvc\_id  AND c2.cust\_id = pvc.pvc\_rcust\_id  AND s2.site\_id = pvc.pvc\_rsite\_id  AND ca2.site\_id = s2.site\_id  AND pa2.site\_id = s2.site\_id  AND p2.loc\_id = s2.prem\_loc\_id  AND IPFR.VPN\_ID = VPN.VPN\_ID  AND VPN.CUST\_ID = CVPN.CUST\_ID  AND (pvc.pvc\_type != 'IPVC' AND pvc.pvc\_type != 'PNPVC')  AND upper(EQUIPMENT.equip\_name) = <equipName>  AND ((pvc.lport = <input port>  AND pvc.lslot = <input slot>  AND pvc.lshelf = <input shelf>)  OR (pvc.rport = <input port>  AND pvc.rslot = <input slot>  AND pvc.rshelf = <input shelf>))  **UNION ALL**  SELECT DISTINCT vpn.cust\_id VPNCUSTID,  pvc.pvc\_id,  pvc.rswitch,  pvc.rslot,  pvc.rport,  pvc.pvc\_rdlci,  pvc.pvc\_rcontrcir,  pvc.pvc\_rvci,  pvc.pvc\_rvpi,  pvc.lswitch,  pvc.lslot,  pvc.lport,  pvc.pvc\_ldlci,  pvc.pvc\_lcontrcir,  pvc.pvc\_lvci,  pvc.pvc\_lvpi,  pvc.PVC\_TYPE,  customer.cust\_name ACUST\_NAME,  customer.cust\_id ACUST\_ID,  customer.cust\_mcn ACUST\_MCN,  site.site\_id ASITE\_ID,  site.site\_num ASITE\_NUM,  site.full\_port\_speed AFULL\_PORT\_SPEED,  site.grc AGRC,  site.protocol APROTOCOL,  site.ta\_name ATA\_NAME,  site.site\_glbl\_dlci ASITE\_GLBL\_DLCI,  site.clli ACLLI,  cust\_access.acc\_ckt AACC\_CKT,  port\_asgmt.port APORT,  port\_asgmt.port\_speed APORT\_SPEED,  premise.loc\_id ALOC\_ID,  premise.prem\_address APREM\_ADDRESS,  premise.prem\_city APREM\_CITY,  premise.prem\_state APREM\_STATE,  premise.prem\_country APREM\_COUNTRY,  --ipfr.cpe\_ip\_address A\_CUST\_IP\_ADDRESS,  --ipfr.per\_ip\_address Z\_CUST\_IP\_ADDRESS,  --// If A end has cust\_id=395, then the cust\_Ip is the PE IP; else, the Cust\_IP is the CE IP  Decode (customer.cust\_id, ‘395’, ipfr.per\_ip\_address, ipfr.cpe\_ip\_address) a\_cust\_ip\_address,  Decode (c2.cust\_id, ‘395’, ipfr.per\_ip\_address, ipfr.cpe\_ip\_address) z\_cust\_ip\_address,  ipfr.vpn\_id,  c2.cust\_name ZCUST\_NAME,  c2.cust\_id ZCUST\_ID,  c2.cust\_mcn ZCUST\_MCN,  s2.site\_id ZSITE\_ID,  s2.site\_num ZSITE\_NUM,  s2.full\_port\_speed ZFULL\_PORT\_SPEED,  s2.grc ZGRC,  s2.protocol ZPROTOCOL,  s2.ta\_name ZTA\_NAME,  s2.site\_glbl\_dlci ZSITE\_GLBL\_DLCI,  s2.clli ZCLLI,  ca2.acc\_ckt ZACC\_CKT,  pa2.port ZPORT,  pa2.port\_speed ZPORT\_SPEED,  p2.loc\_id ZLOC\_ID,  p2.prem\_address ZPREM\_ADDRESS,  p2.prem\_city ZPREM\_CITY,  p2.prem\_state ZPREM\_STATE,  p2.prem\_country ZPREM\_COUNTRY,  VPN.VPN\_NAME ,  VPN.VPN\_ID,  **--//A and Z ends use the same vpn customer data, okay when there’s only one VPN.**  cvpn.cust\_name avpn\_name,  cvpn.cust\_mcn avpn\_mcn,  cvpn.cust\_name zvpn\_name,  cvpn.cust\_mcn zvpn\_mcn,  cvpn.cust\_id a\_vpn\_cust\_id,  cvpn.cust\_id z\_vpn\_cust\_id  FROM premise p2,  port\_asgmt pa2,  cust\_access ca2,  site s2,  customer c2,  ipfr,  premise,  port\_asgmt,  cust\_access,  site,  customer,  pvc,  equipment,  VPN,  CUSTOMER CVPN  WHERE pvc.lequip\_id = equipment.equip\_id  **AND pvc.pvc\_rsite\_id = site.site\_id**  AND site.prem\_loc\_id = premise.loc\_id  AND customer.cust\_id = pvc.pvc\_rcust\_id  AND cust\_access.site\_id = site.site\_id  AND port\_asgmt.site\_id = site.site\_id  AND ipfr.pvc\_id = pvc.pvc\_id  AND c2.cust\_id = pvc.pvc\_lcust\_id  AND s2.site\_id = pvc.pvc\_lsite\_id  AND ca2.site\_id = s2.site\_id  AND pa2.site\_id = s2.site\_id  AND p2.loc\_id = s2.prem\_loc\_id  AND IPFR.VPN\_ID = VPN.VPN\_ID  AND VPN.CUST\_ID = CVPN.CUST\_ID  AND (pvc.pvc\_type != 'IPVC' AND pvc.pvc\_type != 'PNPVC')  AND upper(EQUIPMENT.equip\_name) = <equipName>  AND ((pvc.lport = <input port>  AND pvc.lslot = <input slot>  AND pvc.lshelf = <input shelf>)  OR (pvc.rport = <input port>  AND pvc.rslot = <input slot>  AND pvc.rshelf = <input shelf>)  )  **UNION**  **--new logic for pvc.pvc\_type IN ( 'IPVC', 'PNPVC'), PVC.LSite\_id at A end**  **SELECT DISTINCT**  vpn.cust\_id VPNCUSTID,  pvc.pvc\_id,  pvc.PVC\_TYPE,  pvc.rswitch,  pvc.rslot,  pvc.rport,  pvc.pvc\_rdlci,  pvc.pvc\_rcontrcir,  pvc.pvc\_rvci,  pvc.pvc\_rvpi,  pvc.lswitch,  pvc.lslot,  pvc.lport,  pvc.pvc\_ldlci,  pvc.pvc\_lcontrcir,  pvc.pvc\_lvci,  pvc.pvc\_lvpi,  customer.cust\_name acust\_name,  customer.cust\_id acust\_id,  customer.cust\_mcn acust\_mcn,  site.site\_id asite\_id,  site.site\_num asite\_num,  site.full\_port\_speed afull\_port\_speed,  site.grc agrc,  site.protocol aprotocol,  site.ta\_name ata\_name,  site.site\_glbl\_dlci asite\_glbl\_dlci,  site.clli aclli,  cust\_access.acc\_ckt aacc\_ckt,  port\_asgmt.port aport,  port\_asgmt.port\_speed aport\_speed,  premise.loc\_id aloc\_id,  premise.prem\_address aprem\_address,  premise.prem\_city aprem\_city,  premise.prem\_state aprem\_state,  premise.prem\_country aprem\_country,  --ipfr.cpe\_ip\_address a\_cust\_ip\_address,  --ipfr.per\_ip\_address z\_cust\_ip\_address,  --// If A end has cust\_id=395, then the cust\_Ip is the PE IP; else, the Cust\_IP is the CE IP  Decode (customer.cust\_id, ‘395’, ipfr.per\_ip\_address, ipfr.cpe\_ip\_address) a\_cust\_ip\_address,  Decode (c2.cust\_id, ‘395’, ipfr.per\_ip\_address, ipfr.cpe\_ip\_address) z\_cust\_ip\_address,  ipfr.vpn\_id,  c2.cust\_name zcust\_name,  c2.cust\_id zcust\_id,  c2.cust\_mcn zcust\_mcn,  s2.site\_id zsite\_id,  s2.site\_num zsite\_num,  s2.full\_port\_speed zfull\_port\_speed,  s2.grc zgrc,  s2.protocol zprotocol,  s2.ta\_name zta\_name,  s2.site\_glbl\_dlci zsite\_glbl\_dlci,  s2.clli zclli,  ca2.acc\_ckt zacc\_ckt,  pa2.port zport,  pa2.port\_speed zport\_speed,  p2.loc\_id zloc\_id,  p2.prem\_address zprem\_address,  p2.prem\_city zprem\_city,  p2.prem\_state zprem\_state,  p2.prem\_country zprem\_country,  vpn.vpn\_name,  vpn.vpn\_id,  **--//A and Z end VPNs are different when PVC type is IPVC, or PNPVC.**  **--//VPN1 go with PVC local side, VPN2 go with remote side.**  vpn1.vpn\_id AS aVpn\_id,  vpn1.vpn\_name AS aVpn\_name,  vpnCust1.cust\_id AS a\_Vpn\_Cust\_id  vpnCust1.mcn\_name AS aVpn\_mcn  vpn2.vpn\_id AS zVpn\_id,  vpn2.vpn\_name AS zVpn\_name,  vpnCust2.cust\_id AS z\_Vpn\_Cust\_id  vpnCust2.mcn\_name AS zVpn\_mcn  **FROM**  premise p2,  port\_asgmt pa2,  cust\_access ca2,  site s2,  customer c2,  premise,  port\_asgmt,  cust\_access,  site,  customer,  pvc,  equipment,  ipfr ipfe1,  ipfr ipfe2,  vpn vpn1,  vpn vpn2,  inter\_vpn\_pvc,  customer vpnCust1,  customer vpnCust2  **WHERE** upper(equipment.equip\_name) = <equipName>  AND ((pvc.lequip\_id = equipment.equip\_id  AND (pvc.lport = <input port>  AND pvc.lslot = <input slot>  AND pvc.lshelf = <input shelf>)  )  **OR** (pvc.requip\_id = equipment.equip\_id  AND (pvc.rport = <input port>  AND pvc.rslot = <input slot>  AND pvc.rshelf = <input shelf>)  )  )  AND customer.cust\_id = pvc.pvc\_lcust\_id  **AND pvc.pvc\_lsite\_id = site.site\_id**  AND cust\_access.site\_id = site.site\_id  AND port\_asgmt.site\_id = site.site\_id  AND premise.loc\_id = site.prem\_loc\_id  AND c2.cust\_id = pvc.pvc\_rcust\_id  AND s2.site\_id = pvc.pvc\_rsite\_id  AND ca2.site\_id = s2.site\_id  AND pa2.site\_id = s2.site\_id  AND p2.loc\_id = s2.prem\_loc\_id  And pvc.pvc\_type IN ( 'IPVC', 'PNPVC')  And pvc.pvc\_id = inter\_vpn\_pvc.pvc\_id  And inter\_vpn\_pvc.vpn\_id1 = vpn1.vpn\_id  And inter\_vpn\_pvc.vpn\_id2 = vpn2.vpn\_id  And ipfr1.pvc\_id = pvc.pvc\_id  And ipfr2.pvc\_id = pvc.pvc\_id  And ipfr1.vpn\_id = vpn1.vpn\_id  And ipfr2.vpn\_id = vpn2.vpn\_id  And vpn1.cust\_id = vpnCust1.cust\_id  And vpn2.cust\_id = vpnCust2.cust\_id  **UNION**  **--new logic for pvc.pvc\_type IN ( 'IPVC', 'PNPVC'), PVC.RSite\_id at A end**  **SELECT DISTINCT**  vpn.cust\_id VPNCUSTID,  pvc.pvc\_id,  pvc.PVC\_TYPE,  pvc.rswitch,  pvc.rslot,  pvc.rport,  pvc.pvc\_rdlci,  pvc.pvc\_rcontrcir,  pvc.pvc\_rvci,  pvc.pvc\_rvpi,  pvc.lswitch,  pvc.lslot,  pvc.lport,  pvc.pvc\_ldlci,  pvc.pvc\_lcontrcir,  pvc.pvc\_lvci,  pvc.pvc\_lvpi,  customer.cust\_name acust\_name,  customer.cust\_id acust\_id,  customer.cust\_mcn acust\_mcn,  site.site\_id asite\_id,  site.site\_num asite\_num,  site.full\_port\_speed afull\_port\_speed,  site.grc agrc,  site.protocol aprotocol,  site.ta\_name ata\_name,  site.site\_glbl\_dlci asite\_glbl\_dlci,  site.clli aclli,  cust\_access.acc\_ckt aacc\_ckt,  port\_asgmt.port aport,  port\_asgmt.port\_speed aport\_speed,  premise.loc\_id aloc\_id,  premise.prem\_address aprem\_address,  premise.prem\_city aprem\_city,  premise.prem\_state aprem\_state,  premise.prem\_country aprem\_country,  --ipfr.cpe\_ip\_address a\_cust\_ip\_address,  --ipfr.per\_ip\_address z\_cust\_ip\_address,  --// If A end has cust\_id=395, then the cust\_Ip is the PE IP; else, the Cust\_IP is the CE IP  Decode (customer.cust\_id, ‘395’, ipfr.per\_ip\_address, ipfr.cpe\_ip\_address) a\_cust\_ip\_address,  Decode (c2.cust\_id, ‘395’, ipfr.per\_ip\_address, ipfr.cpe\_ip\_address) z\_cust\_ip\_address,  ipfr.vpn\_id,  c2.cust\_name zcust\_name,  c2.cust\_id zcust\_id,  c2.cust\_mcn zcust\_mcn,  s2.site\_id zsite\_id,  s2.site\_num zsite\_num,  s2.full\_port\_speed zfull\_port\_speed,  s2.grc zgrc,  s2.protocol zprotocol,  s2.ta\_name zta\_name,  s2.site\_glbl\_dlci zsite\_glbl\_dlci,  s2.clli zclli,  ca2.acc\_ckt zacc\_ckt,  pa2.port zport,  pa2.port\_speed zport\_speed,  p2.loc\_id zloc\_id,  p2.prem\_address zprem\_address,  p2.prem\_city zprem\_city,  p2.prem\_state zprem\_state,  p2.prem\_country zprem\_country,  vpn.vpn\_name,  vpn.vpn\_id,  **--//A and Z end VPNs are different when PVC type is IPVC, or PNPVC.**  **--//VPN1 go with PVC local side, VPN2 go with remote side.**  Vpn2.vpn\_id AS aVpn\_id,  Vpn2.vpn\_name AS aVpn\_name,  vpnCust2.cust\_id AS a\_Vpn\_Cust\_id  vpnCust2.mcn\_name AS aVpn\_mcn  vpn1.vpn\_id AS zVpn\_id,  vpn1.vpn\_name AS zVpn\_name,  vpnCust1.cust\_id AS z\_Vpn\_Cust\_id  vpnCust1.mcn\_name AS zVpn\_mcn  **FROM**  premise p2,  port\_asgmt pa2,  cust\_access ca2,  site s2,  customer c2,  premise,  port\_asgmt,  cust\_access,  site,  customer,  pvc,  equipment,  ipfr ipfe1,  ipfr ipfe2,  vpn vpn1,  vpn vpn2,  inter\_vpn\_pvc,  customer vpnCust1,  customer vpnCust2  **WHERE** upper(equipment.equip\_name) = <equipName>  AND ((pvc.lequip\_id = equipment.equip\_id  AND (pvc.lport = <input port>  AND pvc.lslot = <input slot>  AND pvc.lshelf = <input shelf>)  )  **OR** (pvc.requip\_id = equipment.equip\_id  AND (pvc.rport = <input port>  AND pvc.rslot = <input slot>  AND pvc.rshelf = <input shelf>)  )  )  AND customer.cust\_id = pvc.pvc\_lcust\_id  **AND pvc.pvc\_rsite\_id = site.site\_id**  AND cust\_access.site\_id = site.site\_id  AND port\_asgmt.site\_id = site.site\_id  AND premise.loc\_id = site.prem\_loc\_id  AND c2.cust\_id = pvc.pvc\_rcust\_id  AND s2.site\_id = pvc.pvc\_rsite\_id  AND ca2.site\_id = s2.site\_id  AND pa2.site\_id = s2.site\_id  AND p2.loc\_id = s2.prem\_loc\_id  And pvc.pvc\_type IN ( 'IPVC', 'PNPVC')  And pvc.pvc\_id = inter\_vpn\_pvc.pvc\_id  And inter\_vpn\_pvc.vpn\_id1 = vpn1.vpn\_id  And inter\_vpn\_pvc.vpn\_id2 = vpn2.vpn\_id  And ipfr1.pvc\_id = pvc.pvc\_id  And ipfr2.pvc\_id = pvc.pvc\_id  And ipfr1.vpn\_id = vpn1.vpn\_id  And ipfr2.vpn\_id = vpn2.vpn\_id  And vpn1.cust\_id = vpnCust1.cust\_id  And vpn2.cust\_id = vpnCust2.cust\_id |
|  | **For each PVC record, Derive TypeOfService data**  If SITEA.protocol is ‘FR’ (for a\_typeOfService) Or SITEZ.protocol is ‘FR’ (for z\_typeOfService)  If PREMISEA.prem\_country is ‘USA’ (for a\_typeOfService) Or PREMISEZ.prem\_country is ‘USA’ (for z\_typeOfService)  Set TypeOfService = ‘FR’  Else  Set TypeOfService = ‘IFR’  EndIf  EndIf  If SITEA.protocol is ‘ATM’ (for a\_typeOfService) Or SITEZ.protocol is ‘ATM’ (for z\_typeOfService)  If PREMISEA.prem\_country is ‘USA’ (for a\_typeOfService) Or PREMISEZ.prem\_country is ‘USA’ (for z\_typeOfService)  Set TypeOfService = ‘ATM’  Else  Set TypeOfService = ‘IATM’  EndIf  EndIf  If SITEA.protocol is not ‘ATM’ or ‘FR’, Then  Set AtypeOfService = SITEA.protocol  EndIf  If SITEZ.protocol is not ‘ATM’ or ‘FR’, Then  Set ZtypeOfService = SITEZ.protocol  EndIf  Retrieve all records from ICORE tables as described in “FR-ATM query response” |
| 4.a) | For PVC records, Retrieve from GCP SIDBOR database using ICORE acc\_ckt data from step 3.a  **Input: <acc\_ckt>**  <288315> Adding new fields for both A and Z ends  managedIndicator  parentOrgGroup  **For PVC A end:**  Select  FWTOPOLOGY.local\_router,  ~~ASSETA.src\_org\_id,~~  ASSETA.org\_cd, <288315>  ASSETA.managing\_org\_cd,  ASSETA.active\_org\_cd,  ASSETA.functional\_area,  ~~ASSETA.service,~~  Decode (Upper(ASSETA.service), <288315-US681337-upd4>  ‘IPEFX’, ‘BVOIP’,  ‘IPEFR’, ‘IPFR’,  ‘FRAME’, ‘FR’,  ASSETA.service) AS service,  CLIENT\_ORG.org\_cd\_prnt AS parentOrgGroup, <288315>  ‘Y’ managedIndicator, <288315>  From  FWTOPOLOGY,  ASSET ASSETA  CLIENT\_ORG  Where FWTOPOLOGY.local\_access\_ckt = <acc\_ckt>  And FWTOPOLOGY.functional\_area = ASSETA.functional\_area  And FWTOPOLOGY.local\_router = ASSETA.asset\_nm  And ASSETA.org\_cd = CLIENT\_ORG.org\_cd  And ASSETA.functional\_area = CLIENT\_ORG.functional\_area  If data found  Set A CERName = FWTOPOLOGYA.local\_router  ~~Set A OrgGroup = ASSETA.src\_org\_id~~  Set A OrgGroup = ASSETA.org\_cd, <288315>  Set A ManageOrg = ASSETA. Managing\_org\_cd  Set A ActiveOrg = ASSETA. Active\_org\_cd  Set A FUNCTIONALAREA= ASSETA. Functional\_area  Set A Service = ASSETA. Service  Set A parentOrgGroup = blank <288315>  Set A managedIndocator = managedIndicator <288315>  End If  **For PVC Z end:**  Select  FWTOPOLOGY.local\_router,  ~~ASSETZ.src\_org\_id,~~  ASSETZ.org\_cd, <288315>  ASSETZ.managing\_org\_cd,  ASSETZ.active\_org\_cd,  ASSETZ.functional\_area,  ~~ASSETZ.service,~~  Decode (Upper(ASSETZ.service), <288315-US681337-upd4>  ‘IPEFX’, ‘BVOIP’,  ‘IPEFR’, ‘IPFR’,  ‘FRAME’, ‘FR’,  ASSETZ.service) AS service,  CLIENT\_ORG.org\_cd\_prnt AS parentOrgGroup, <288315>  ‘Y’ managedIndicator, <288315>  From  FWTOPOLOGY,  ASSET ASSETZ  CLIENT\_ORG  Where FWTOPOLOGY.local\_access\_ckt = <acc\_ckt>  And FWTOPOLOGY.functional\_area = ASSETZ.functional\_area  And FWTOPOLOGY.local\_router = ASSETZ.asset\_nm  And ASSETZ.org\_cd = CLIENT\_ORG.org\_cd  And ASSETZ.functional\_area = CLIENT\_ORG.functional\_area  If data found  Set A CERName = FWTOPOLOGYA.local\_router  ~~Set Z OrgGroup = ASSETZ.src\_org\_id~~  Set Z OrgGroup = ASSETZ.org\_cd, <288315>  Set Z ManageOrg = ASSETZ.managing\_org\_cd  Set Z ActiveOrg = ASSETZ.active\_org\_cd  Set Z FUNCTIONALAREA= ASSETZ.functional\_area  Set Z Service = ASSETZ. Service  Set Z parentOrgGroup = blank <288315>  Set Z managedIndocator = managedIndicator <288315>  End If |
| 4.b) | NOTE: this step is commented out until further development with CCI  Retrieve from GCP(DBOR) CCI database using ICORE customer\_mcn data from step 3.a  Match CUSTOMERA.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  Match CUSTOMERZ.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  and CUSTOMERA.cust\_mcn (pos. 7-9) =MCN\_ENTRPRS\_XREF.MCN\_SFX  and CUSTOMERZ.cust\_mcn (pos. 7-9) =MCN\_ENTRPRS\_XREF.MCN\_SFX    and XREF\_MCN\_ENTRPRS.ENT\_PTY\_ID = PARTY.ENTRPRS\_PTY\_ID  and PARTY.DUNS\_NB = DNB.DUNS\_NB  Retrieve DNB.(DUNS\_NB, PRNTHQ\_BIZ\_NM)  Set CCI = DUNS\_NB  Set PARENT NAME = PRNTHQ\_BIZ\_NM  Else  IF CUSTOMERA.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  CUSTOMERZ.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  and XREF\_MCN\_ENTRPRS.ENT\_PTY\_ID = PARTY.ENTRPRS\_PTY\_ID  and PARTY.DUNS\_NB = DNB.DUNS\_NB  Retrieve DNB.(DUNS\_NB, PRNTHQ\_BIZ\_NM)  Set CCI = DUNS\_NB  Set PARENT NAME = PRNTHQ\_BIZ\_NM    Else  Set CCI = blank  Set PARENT NAME = blank  EndIf  EndIf |
| 5. | If AOTS.SERVICE TYPE is ‘AGN-CNT’  Return record count of all data found in step 3.a and 3.b as described in “AGN Record Count response”  Else  Return all records as described in “AGN query response”  EndIf |
| 6. | Provide all tables/columns described in “AGN QUERY RESPONSE” in BO for reporting |
| 7. | Create Record Selector for each of the records retrieved in steps 3 & 4 starting with number 1  Set Customer\_Record\_Selector = Record Selector |
| 8. | Create return message to AOTS as described in “AGN QUERY RESPONSE” using data from steps 3 & 4  If Input\_ change\_id & cmdc\_Trans\_ID are populated then  Insert retrieved AGN data from steps 3, 4 into table “ACMR5\_CNC\_ALLCUSTOMERSAFFECTED”  Set input data as:  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.REQUEST\_ID = Input\_change\_id  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.cmdc\_transid = Input\_cmdc\_TransID  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.customer\_record\_selector = customer\_record\_selector  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Customer\_ID = Customer ID  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.customer\_name = cust\_name  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.customer\_mcn = cust\_mcn  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Location\_ID = LocationID  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Site\_ID = Site Number  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Site\_Protocol = Site Protocol  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Site\_TA = Site TA  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Global\_DLCI = Global DLCI  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Circuit\_ID = Circuit ID  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.type\_of\_service = TypeOfService  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Port = Port  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Slot = Slot  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.LOCATION\_ID = A Location ID  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.a\_address = A address  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.a\_city = A city  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.a\_state = A state  ACMR5\_CNC\_ALLCUSTOMERSAFFECTEDa\_country = A country  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.a\_clli = A clli  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Port\_Speed = PortSpeed  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Org\_Group = OrgGroup  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Managing\_Org = ManageOrg  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Active\_Org = ActiveOrg  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Functional\_Area = Functional Area  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Restriction\_Type = RestrictionType  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.CER\_name = CER name  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.CCI = CCI  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Parent\_Company = ParentCompanyName  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Circuit\_bmp\_format = circuitBmpFormat  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Managed\_indicator = managedIndicator  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Bvoip\_indicator = bvoidIndicator  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Parent\_org\_group = parentOrgGroup <288315>  Else  Goto step 12  EndIf  **<288315>**  Insert retrieved CustomerAffected data, same attributes as shown above, to GCP\_cnc\_AllCustomersAffected  After all the customer records inserted for this transaction, calculate the total number of records been inserted:  Set CustomerRecordCnt = last-record\_selector (Customer\_Record\_Selector)  **</288315>** |
| 9 | After the data is stored successfully, GCP call AOTS CM notification web service, sending them change\_id & transaction\_id as input parameters, for data that has been stored |
| 10 | Create a Customer\_Record\_Selector for each of the records retrieved from step 3.d, 3.e & 3.f for each change Request ID starting from 10000. |
| 11 | <288315>  Call ~~UC-DBOR-PB932-19~~ Common logic – Load\_Change\_PVC: To store retrieved PVC data in table Change\_PVC |
| 11.b | <288315> Generate new record and load or update the Summary table  Execute common logic: Load\_GCP\_AllCustomersAffected\_sum |
| 11.c | <288315> ETL  Create CustomerRecord File  Create PVCRecord File |
| 12 | Use Case ends |

|  |  |
| --- | --- |
| POST-CONDITIONS | |
| Course | Description |
| 01 | GCP(DBOR) ICORE data retrieved successfully |
| 02 | GCP(DBOR) data returned to AOTS successfully |

|  |  |
| --- | --- |
| EXCEPTIONS | |
| Exception ID | Action(s) on Exception |
| 01 | Ideal Course Step 3 – GCP(DBOR) ICORE database not available for data retrieval. Create Error “GCP(DBOR)\_001” |
| 02 | Ideal Course Step 4 – Retrieved data cannot be formatted to return to AOTS. Create Error “GCP(DBOR)\_002” |

### InquireCustomerCircuitDetailsByNetworkElement for servicetype=AGN-RPM – 288315

**Formally known as “GetCustInfoQuery for servicetype=AGN-RPM”**

|  |  |
| --- | --- |
| **Query Name** | getCustInfoQuery for servicetype=AGN-RPM |
| **PIDs** | 289116.140768, 288315 |
| **Service** | AVPN |
| **Data Source** | ICORE, SIDBOR |
| **WSDL File** | CmdcWebServicesContract.wsdl |
| **Client App** | AOTS CM |
| **Purpose, Usage** | AGN-RPM Customer Inventory Retrieval.  This use case details the customer inventory request from AOTS for AGN RPM data. AGN-RPM query is to be created for project P9A78. GCP will retrieve data from GCP ICORE database. This is the same as AGN query but user will input the information as router in AOTS and AOTS will parse the information as:  EQUIP\_NAME or EQUIP\_NAME + SLOT; |

**Query Change Summary**

|  |  |
| --- | --- |
| **Project /Ticket ID** | **Change Summary / Notes** |
| 8/22/2007  5.1 | Lan Tran-Vu:  Initial Issue |
| 289116.140768  July’2016 | US636605 - CR140768 - US GCP-AOTS-CM supporting uCPE for change management  Enhance for AOTS-CM to retrieve uCPE data when a PE is chosen as the input.  Dependency: The required data are available in the source systems --- A&AI, ~~GDB, CANOPI~~ |
| 288315  Oct’2016 | US681337 – Support AOTS CM ticket notification.  This enhancement includes  Add new fields to both Customer records and PVC records;  Change the process to Async, and the data retrieval logic is used by ETL to generate files. |
| 301033  June’2018 | US374866   * Re-engineer EDF logic to use the A&AI data sourced from DMaaP instead of the A&AI batch feed. * Support Vyatta uCPE. |

|  |  |
| --- | --- |
| USE CASE GENERAL INFORMATION | |
| Use Case Name | AGN-RPM Customer Inventory Retrieval |
| Use Case ID | UC-DBOR-P9A78-021 |
| Description | This use case details the customer inventory request from AOTS for AGN RPM data. AGN-RPM query is to be created for project P9A78. DBOR will retrieve data from DBOR ICORE database. This is the same as AGN query but user will input the information as router in AOTS and AOTS will parse the information as:  EQUIP\_NAME or EQUIP\_NAME + SLOT;  Data Source: ICORE, SIDBOR |
| Responsible Analyst | Lan TranVu |
| Type of Execution | WebService and SQL to Oracle db. |

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE REVISION LOG | | | |
| Reason for Revision: | Initial Issue | Release Number: | 5.1 |
| Author: | Lan Tran-Vu | Revised: | 8/22/07 |

|  |  |
| --- | --- |
| ACTORS | |
| Primary Actors | AOTS |
| Secondary Actors | DBOR |

|  |  |
| --- | --- |
| PRE-CONDITIONS | |
| # | Description |
| 1 | AOTS sent request in WebService getCustInfoQuery with ServiceType “AGN-RPM” |

|  |  |  |
| --- | --- | --- |
| IDEAL COURSE | | |
| # | Step Description | |
| 1. | Use Case begins | |
| 2. | DBOR received query getCustInfoQuery request from AOTS with ServiceType = ‘AGN-RPM’  Set Input\_Change ID = AOTS. Change ID  Set Input\_CmdcTransID = AOTS. CmdcTransID  Set Input\_ EquipName = AOTS.EquipName  If AOTS.PORT is populated and AOTS.SLOT is not populate  If AOTS.Input is Slot  Set Input\_slot = SLOT from AOTS  EndIf | |
| 3 | Retrieve data from ICORE, SIDBOR and CCI schema as existing query with ServiceType = ‘AGN’  <289116.140768-US636605>  call GetUcpeByCircuitID (cust\_access.acc\_ckt) and populate CpeDetailsList | |
| 4 | **Rtrieve PVC records from ICORE (GCP replication) tables**:  <288315> Adding new fields for both A and Z ends  circuitIdBmpFormat  Retrieve From **PVC**  (PVC\_ID, RSWITCH, RSLOT, RPORT, PVC\_RDLCI, PVC\_RCONTRCIR, PVC\_rVCI, PVC\_rVPI,  LSWITCH, LSLOT, LPORT, PVC\_LDLCI, PVC\_LCONTRCIR, PVC\_lVPI, PVC\_lVCI )  VPNA.VPN\_NAME VPNZ.VPN\_NAME  IPFRA.VPN\_ID  IPFRZ.VPN\_ID  SITEA  (SITE\_ID, FULL\_PORT\_SPEED , GRC, CLLI)  SITEZ  (SITE\_ID, FULL\_PORT\_SPEED , GRC, CLLI)  CUSTOMERA  (CUST\_NAME, CUST\_MCN,CUST\_ID)  CUSTOMERZ  (CUST\_NAME, CUST\_MCN,CUST\_ID)  PREMISEA  (LOC\_ID, PREM\_ADDRESS, PREM\_CITY, PREM\_STATE, PREM\_COUNTRY)  PREMISEZ  (LOC\_ID, PREM\_ADDRESS, PREM\_CITY, PREM\_STATE, PREM\_COUNTRY)  CUST\_ACCESS1  (ACC\_CKT)  CUST\_ACCESS2  (ACC\_CKT )  Decode (Cust\_accessA.acc\_ckt, <288315>  Instr (Cust\_accessA.acc\_ckt, ‘.’, 1, 1) = 3, ‘Y’,  Instr (Cust\_accessA.acc\_ckt, ‘.’, 1, 1) = 5, ‘Y’,  ‘N’ ) cktConversionFlagA, <288315> Used to mark whether the circuit format can be converted.  Decode (Cust\_accessZ.acc\_ckt,  Instr (Cust\_accessZ.acc\_ckt, ‘.’, 1, 1) = 3, ‘Y’,  Instr (Cust\_accessZ.acc\_ckt, ‘.’, 1, 1) = 5, ‘Y’,  ‘N’ ) cktConversionFlagZ, <288315> Used to mark whether the circuit format can be converted.  From DBOR ICORE tables: PVC, IPFRA,IPFRZ, VPNA,VPNZ,CUSTOMERA,CUSTOMERZ,CUST\_ACCESSA, CUST\_ACCESSZ, SITEA, SITEZ, PREMISEA, PREMISEZ, PORT\_ASGMT, EQUIPMENT  Match  EQUIPMENT.equip\_id = PVC.requip\_id  And PORT\_ASGMT.equip\_id = PVC.lequip\_id  And SITEZ.prem\_loc\_id = PREMISEZ.loc\_id(+) AND SITEA.prem\_loc\_id = PREMISEA.loc\_id And SITEZ.site\_id = PVC.pvc\_rsite\_id(+) AND SITEA.site\_id = PVC.pvc\_lsite\_id And PVC.pvc\_id = IPFRA.pvc\_id(+)  And PVC.pvc\_id = IPFRZ.pvc\_id(+)  And IPFRA.vpn\_id = VPNA.vpn\_id  And IPFRZ.vpn\_id = VPNZ.vpn\_id  And PVC.pvc\_rcust\_id = customerZ.cust\_id  And PVC.pvc\_lcust\_id = customerA.cust\_id  And CUST\_ACCESSZ.cust\_id = CUSTOMERZ.cust\_id(+)  And CUST\_ACCESSA.cust\_id = CUSTOMERA.cust\_id(+)  ~~AND SITEA.site\_id = CUST\_PREM\_EQUIPA.~~  ~~And SITEZ.site\_id = CUST\_PREM\_EQUIPZ.~~  And EQUIPMENT.equip\_name = Input\_EquipName  And PORT\_ASGMT.port = Input\_port (only if Input\_port is populated)  And PORT\_ASGMT.slot = Input\_slot (only if Input\_slot is populated)  <288315>  Convert the acc\_ckt (in ICORE format) to normalized/BMP format, based on the cktConversionFlag value.  This is only for domestic circuit. For international circuit, leave it as is.  If cktConversionFlagA = Y, then  Call Common logic –CircuitFormatConversion\_ICORE\_to\_BMP (<A\_acc\_ckt>) -> A\_circuitIdBmpFormat  Else  Set A\_circuitIdBmpFormat = <A\_acc\_ckt>  End If  If cktConversionFlagZ = Y, then  Call Common logic –CircuitFormatConversion\_ICORE\_to\_BMP (<Z\_acc\_ckt>) -> Z\_circuitIdBmpFormat  Else  Set Z\_circuitIdBmpFormat = <Z\_acc\_ckt>  End If | |
|  | **For each PVC record, Derive TypeOfService data**  If SITEA.protocol is ‘FR’ (for a\_typeOfService) Or SITEZ.protocol is ‘FR’ (for z\_typeOfService)  If PREMISEA.prem\_country is ‘USA’ (for a\_typeOfService) Or PREMISEZ.prem\_country is ‘USA’ (for z\_typeOfService)  Set TypeOfService = ‘FR’  Else  Set TypeOfService = ‘IFR’  EndIf  EndIf  If SITEA.protocol is ‘ATM’ (for a\_typeOfService) Or SITEZ.protocol is ‘ATM’ (for z\_typeOfService)  If PREMISEA.prem\_country is ‘USA’ (for a\_typeOfService) Or PREMISEZ.prem\_country is ‘USA’ (for z\_typeOfService)  Set TypeOfService = ‘ATM’  Else  Set TypeOfService = ‘IATM’  EndIf  EndIf  If SITEA.protocol is not ‘ATM’ or ‘FR’, Then  Set ATypeOfService = SITEA.protocol  EndIf  If SITEZ.protocol is not ‘ATM’ or ‘FR’, Then  Set ZTypeOfService = SITEZ.protocol  EndIf  Retrieve all records from ICORE tables as described in “FR-ATM query response” | |
| 4.a) | For PVC records, Retrieve from GCP SIDBOR database using ICORE acc\_ckt data from step 3.a  **Input: <acc\_ckt>**  <288315> Adding new fields for both A and Z ends  managedIndicator  parentOrgGroup  **For PVC A end:**  Select  FWTOPOLOGY.local\_router,  ~~ASSETA.src\_org\_id,~~  ASSETA.org\_cd, <288315>  ASSETA.managing\_org\_cd,  ASSETA.active\_org\_cd,  ASSETA.functional\_area,  ~~ASSETA.service,~~  Decode (Upper(ASSETA.service), <288315-US681337-upd4>  ‘IPEFX’, ‘BVOIP’,  ‘IPEFR’, ‘IPFR’,  ‘FRAME’, ‘FR’,  ASSETA.service) AS service,  CLIENT\_ORG.org\_cd\_prnt AS parentOrgGroup, <288315>  ‘Y’ managedIndicator, <288315>  From  FWTOPOLOGY,  ASSET ASSETA  CLIENT\_ORG  Where FWTOPOLOGY.local\_access\_ckt = <acc\_ckt>  And FWTOPOLOGY.functional\_area = ASSETA.functional\_area  And FWTOPOLOGY.local\_router = ASSETA.asset\_nm  And ASSETA.org\_cd = CLIENT\_ORG.org\_cd  And ASSETA.functional\_area = CLIENT\_ORG.functional\_area  If data found  Set A CERName = FWTOPOLOGYA.local\_router  ~~Set A OrgGroup = ASSETA.src\_org\_id~~  Set A OrgGroup = ASSETA.org\_cd, <288315>  Set A ManageOrg = ASSETA. managing\_org\_cd  Set A ActiveOrg = ASSETA. active\_org\_cd  Set A FUNCTIONALAREA= ASSETA. functional\_area  Set A Service = ASSETA. Service  Set A parentOrgGroup = blank <288315>  Set A managedIndocator = managedIndicator <288315>  **For PVC Z end:**  Select  FWTOPOLOGY.local\_router,  ~~ASSETZ.src\_org\_id,~~  ASSETZ.org\_cd, <288315>  ASSETZ.managing\_org\_cd,  ASSETZ.active\_org\_cd,  ASSETZ.functional\_area,  ~~ASSETZ.service,~~  Decode (Upper(ASSETZ.service), <288315-US681337-upd4>  ‘IPEFX’, ‘BVOIP’,  ‘IPEFR’, ‘IPFR’,  ‘FRAME’, ‘FR’,  ASSETZ.service) AS service,  CLIENT\_ORG.org\_cd\_prnt AS parentOrgGroup, <288315>  ‘Y’ managedIndicator, <288315>  From  FWTOPOLOGY,  ASSET ASSETZ  CLIENT\_ORG  Where FWTOPOLOGY.local\_access\_ckt = <acc\_ckt>  And FWTOPOLOGY.functional\_area = ASSETZ.functional\_area  And FWTOPOLOGY.local\_router = ASSETZ.asset\_nm  And ASSETZ.org\_cd = CLIENT\_ORG.org\_cd  And ASSETZ.functional\_area = CLIENT\_ORG.functional\_area  If data found  Set A CERName = FWTOPOLOGYA.local\_router  ~~Set Z OrgGroup = ASSETZ.src\_org\_id~~  Set Z OrgGroup = ASSETZ.org\_cd, <288315>  Set Z ManageOrg = ASSETZ.managing\_org\_cd  Set Z ActiveOrg = ASSETZ.active\_org\_cd  Set Z FUNCTIONALAREA= ASSETZ.functional\_area  Set Z Service = ASSETZ. service  Set Z parentOrgGroup = blank <288315>  Set Z managedIndocator = managedIndicator <288315> | |
| 4.b) | NOTE: this step is commented out until further development with CCI  Retrieve from GCP(DBOR) CCI database using ICORE customer\_mcn data from step 3.a  Match CUSTOMERA.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  Match CUSTOMERZ.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  and CUSTOMERA.cust\_mcn (pos. 7-9) =MCN\_ENTRPRS\_XREF.MCN\_SFX  and CUSTOMERZ.cust\_mcn (pos. 7-9) =MCN\_ENTRPRS\_XREF.MCN\_SFX    and XREF\_MCN\_ENTRPRS.ENT\_PTY\_ID = PARTY.ENTRPRS\_PTY\_ID  and PARTY.DUNS\_NB = DNB.DUNS\_NB  Retrieve DNB.(DUNS\_NB, PRNTHQ\_BIZ\_NM)  Set CCI = DUNS\_NB  Set PARENT NAME = PRNTHQ\_BIZ\_NM  Else  IF CUSTOMERA.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  CUSTOMERZ.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  and XREF\_MCN\_ENTRPRS.ENT\_PTY\_ID = PARTY.ENTRPRS\_PTY\_ID  and PARTY.DUNS\_NB = DNB.DUNS\_NB  Retrieve DNB.(DUNS\_NB, PRNTHQ\_BIZ\_NM)  Set CCI = DUNS\_NB  Set PARENT NAME = PRNTHQ\_BIZ\_NM    Else  Set CCI = blank  Set PARENT NAME = blank  EndIf  EndIf | |
| 5. | Create Record Selector for each of the records retrieved in step 3 starting with number 1  Set Customer\_Record\_Selector = Record Selector | |
| 6. | | Create return message to AOTS for all records as existing ServiceType = ‘AGN’  Refer to AOTS-DBOR IAD document “WPID\_C3 13\_R14\_AOTSCM-DBOR\_IAD\_V14.04”  If no data found return error ‘0000- No Data Found’  If Input\_ change\_id & cmdc\_Trans\_ID are populated then  Insert retrieved AGN-RPM data from steps 3a, 3b, 3c into table “ACMR5\_CNC\_ALLCUSTOMERSAFFECTED”  Set input data as:  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.REQUEST\_ID = Input\_change\_id  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.cmdc\_transid = Input\_cmdc\_TransID  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.customer\_record\_selector = customer\_record\_selector  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Customer\_ID = Customer ID  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Customer\_Name = CustomerName  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Customer\_MCN = Customer MCN  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.GRC = grc  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Location\_ID = LocationID  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Site\_ID = Site Number  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.IP\_Address = IPAddress  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Equipment\_Name = EquipmentName  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Port = Port  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Slot = Slot  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Circuit\_ID = Circuit ID  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.type\_of\_service = TypeOfService  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Address = Address  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.City = City  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.State = State  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Country = Country  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.CLLI = clli  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.CER\_name = CER name  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Org\_Group = OrgGroup  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Managing\_Org = ManageOrg  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Active\_Org = ActiveOrg  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Functional\_Area = Functional Area  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Cust\_IP\_Address = CustIPAddress  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Provider\_Circuit = ProviderCircuit  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.PTTProvider = PTTProvider  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.CCI = CCI  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Parent\_Company = ParentCompany  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Circuit\_bmp\_format = circuitBmpFormat  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Managed\_indicator = managedIndicator  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Bvoip\_indicator = bvoidIndicator  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Parent\_org\_group = parentOrgGroup <288315>  Else  Goto step 10  EndIf  **<288315>**  Insert retrieved CustomerAffected data, same attributes as shown above, to GCP\_cnc\_AllCustomersAffected  After all the customer records inserted for this transaction, calculate the total number of records been inserted:  Set CustomerRecordCnt = last-record\_selector (Customer\_Record\_Selector)  **</288315>** |
| 7 | | After the data is stored successfully, GCP call AOTS CM notification web service, sending them change\_id & transaction\_id as input parameters, for data that has been stored |
| 8 | | Create a Customer\_Record\_Selector for each of the records retrieved from step 3.a , 3.b & 3.c for each change Request ID starting from 10000. |
| 9 | | <288315>  Call ~~UC-DBOR-PB932-19~~ Common logic – Load\_Change\_PVC: To store retrieved PVC data in table Change\_PVC |
| 9.b | | <288315> Generate new record and load or update the Summary table  Execute common logic: Load\_GCP\_AllCustomersAffected\_sum |
| 9.c | | <288315> ETL:  Create CustomerRecord File  Create PVCRecord File |
| 10 | | Use Case ends |

|  |  |
| --- | --- |
| POST-CONDITIONS | |
| Course | Description |
| 01 | DBOR ICORE data retrieved successfully |
| 02 | DBOR data returned to AOTS successfully |

|  |  |
| --- | --- |
| EXCEPTIONS | |
| Exception ID | Action(s) on Exception |
| 01 | Ideal Course Step 3a – DBOR ICORE database not available for data retrieval. |
| 02 | Ideal Course Step 4 – Retrieved data cannot be formatted to return to AOTS. |

### InquireCustomerCircuitDetailsByNetworkElement for servicetype=GMIS – 297409-CT173072

**Formally known as “GetCustInfoQuery for servicetype=GMIS”**

|  |  |
| --- | --- |
| **Query Name** | getCustInfoQuery for servicetype=GMIS  InquireCustomerCircuitDetailsByNetworkElement for servicetype=GMIS  ETL backend data retrieval logic |
| **PIDs** | 289116.140768, 288315, 297409, 297409 CR173072, 297100 |
| **Service** | AVPN,MIS |
| **Data Source** | INSTAR |
| **WSDL File** | CmdcWebServicesContract.wsdl |
| **Client App** | AOTS CM |
| **Purpose, Usage** | GMIS Customer Inventory Retrieval.  This use case details the customer inventory request from AOTS for GMIS data. GMIS query is to be created for project P9A78. GCP will retrieve data from GCP GRDB database. Satisfying P9A78 – Change Management Bundled features, requirement.  Input data combination can be:  EQUIP\_NAME;  EQUIP\_NAME + SLOT;  EQUIP\_NAME + PORT;  EQUIP\_NAME + SLOT + PORT; |

**Query Change Summary**

|  |  |
| --- | --- |
| **Project /Ticket ID** | **Change Summary / Notes** |
| 6/13/2007  5.0 | Lan Tran-Vu:  Initial Issue |
| 8/28/2007  5.2 | Lan Tran-Vu:  Update to add Slot and Port as optional inputs |
| 2/7/2008 | Karen Figurelli:  Change data source (except for two telco fields) from GRDB to INSTAR |
| 289116.140768  July’2016 | US636605 – CR140768 – US GCP-AOTS-CM supporting uCPE for change management  Enhance for AOTS-CM to retrieve uCPE data when a PE is chosen as the input.  Dependency: The required data are available in the source systems --- A&AI, ~~GDB, CANOPI~~ |
| 288315  Oct’2016 | US681337 – Support AOTS CM ticket notification.  This enhancement includes  Add new fields to Customer records;  Change the process to Async, and the data retrieval logic is used by ETL to generate files. |
| 297409  Oct’2017 | <297409-workitem-281399>  For serviceType=VPLS, Include logic to cover services for MIS with the VPLS as dedicated PE for MIS. |
| 297409 CR173072  Dec’2017 | <297409-CR173072>   * Add shelf, slot, port fields to Customer record structure in the response, for the input equipment (PE or Gateway). * These new fields are only populated for service line of AVPN, OEW, and MIS (in logic flow for servive type of FR/ATM, GMIS, VPLS) * For 10/40/100 Gige MIS, VPLS-PE as dedicated device to customer, same as PE. Using GMIS logic flow. |
| 301033  June’2018 | US374866   * Re-engineer EDF logic to use the A&AI data sourced from DMaaP instead of the A&AI batch feed. * Support Vyatta uCPE. |

|  |  |
| --- | --- |
| USE CASE GENERAL INFORMATION | |
| Use Case Name | GMIS Customer Inventory Retrieval |
| Use Case ID | UC-DBOR-P9A78-020 |
| Description | This use case details the customer inventory request from AOTS for GMIS data. GMIS query is to be created for project P9A78. DBOR will retrieve data from DBOR GRDB database. Satisfying P9A78 – Change Management Bundled features, requirement.  Input data combination can be:  EQUIP\_NAME;  EQUIP\_NAME + SLOT;  EQUIP\_NAME + PORT;  EQUIP\_NAME + SLOT + PORT;  Data Source: INSTAR, GRDB |
| Responsible Analyst | Lan TranVu |
| Type of Execution | WebService and SQL to Oracle db. |

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE REVISION LOG | | | |
| Reason for Revision: | Change data source (except for two telco fields) from GRDB to INSTAR | Production Ticket#  106842400 | 02/07/2008 |
| Author: | Karen Figurelli |  |  |
| Reason for Revision: | Update to add Slot and Port as optional inputs | Release Number: | 5.2 |
| Author: | Lan Tran-Vu | Revised: | 08/28/07 |
| Reason for Revision: | Initial Issue | Release Number: | 5.0 |
| Author: | Lan Tran-Vu | Revised: | 06/13/07 |

**Output / Return Structures:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| GMIS Query Response | | | | | |
| Field | R/O/C | Data Type | Occur | Notes | GMIS |
| Header | R | N/A | 1 | Begin Tag |  |
| Status | R | AN (3) | 1 | The status of the transaction.  Valid Values:  Success  Warning  Error |  |
| error Code | C | AN (10) | 1 | This is the error code.  This field is required if Status\_Code is Warning or Error. Refer to Error list in 4.3 section |  |
| Error Message | C | AN (250) | 1 | This is the error message. |  |
| Header | R | N/A | 1 | End Tag |  |
| Body | R | N/A | 1:N | Begin Tag |  |
| Customer Information Response | R | N/A | 0:N | Begin Tag |  |
| CUSTOMER\_RECORD\_SELECTOR | R |  |  |  |  |
| Customer ID | R | Int | 1 | No trailing blanks | INSTAR  FACILITY\_ACCOUNT.cust\_id |
| Customer Name | R | A50 | 1 | No trailing blanks | INSTAR  BIDS\_CUSTOMER.cust\_name |
| Customer MCN | ~~R~~ O | A9 | 1 | No trailing blanks | INSTAR  FACILITY\_ACCOUNT.mcn |
| GRC | ~~R~~ O | A3 | 1 | No trailing blanks | INSTAR  FACILITY\_ACCOUNT.grc |
| Location ID | ~~R~~ O | Int | 1 | No trailing blanks | INSTAR  SITE.loc\_id  <297100>Not applicable for Arista/Drivenet equipment. |
| Site ID | ~~R~~ O | Int | 1 | No trailing blanks | INSTAR  SITE.site\_id  <297100>Not applicable for Arista/Drivenet equipment. |
| IP Address | ~~R~~ O | A20 | 1 | No trailing blanks | INSTAR  SERIAL\_IP\_ADDRESS(2).ip\_address  <297100>Not applicable for Arista/Drivenet equipment. |
| Equipment Name | ~~R~~ O | A20 | 1 | No trailing blanks | INSTAR  EQUIPMENT.ptnii\_equip\_name |
| Port | O | Int | 1 | No trailing blanks | INSTAR  COMPONENT.port – derived as follows:  mod(component1.port,1000)  /\*Returns the remainder \*/  <297100>Not applicable for Arista/Drivenet equipment. |
| Slot | O | Int | 1 | No trailing blanks | INSTAR  COMPONENT.beg\_slot  <297100>Not applicable for Arista/Drivenet equipment. |
| Port Speed | O | Int | 1 | No trailing blanks | INSTAR  IP\_SERV\_ACC\_SPEED.speed\_value  <297100>Not applicable for Arista/Drivenet equipment. |
| CircuitId | O | A57 | 1 |  | INSTAR  CUST\_ACCESS.acc\_ckt  Note: This field has value directly retrieved from source DB, without format conversion.  seems BMP/normalized format being returned currently;  Check whether it’s converted already by GCP or ICORE.  Then determined whether new field is needed |
| circuitIdBmpFormat | O | A50 | 1 |  | <288315> Customer access circuit in BMP (normalized) format.  Deived value.  Note:  No conversion for International circuit. |
| Address | ~~R~~ O | A75 | 1 | No trailing blanks | INSTAR  PREMISE.address |
| City | ~~R~~ O | A26 | 1 | No trailing blanks | INSTAR  PREMISE.city |
| State | ~~R~~ O | A2 | 1 | No trailing blanks | INSTAR  PREMISE.state\_abbr |
| Country | ~~R~~ O | A128 | 1 | No trailing blanks | INSTAR  PREMISE.country\_name |
| CLLI | O | A11 | 1 | No trailing blanks | GRDB  GRDB\_ROUTER.clli\_code |
| CER Name | O | A50 | 1 | No trailing blanks | GRDB  GRDB\_ROUTER.common\_narme |
| managedIndicator | O | A5 | 1 | No trailing blanks | <288315> Values: Y, N, or null if no CPE exist  <297100>Not applicable for Arista/Drivenet equipment. |
| bvoipIndicator | O | A5 | 1 | No trailing blanks | <288315> Value  s: Y, N, or null  <297100>Not applicable for Arista/Drivenet equipment. |
| OrgGroup | O | A11 | 1 | No trailing blanks | GRDB  GRDB\_CMP\_ORG.org\_acct  Remove the first character if it is a dash ‘-‘.  For example:  for “-EV\_ACEIC” return “EV\_ACEIC”  <297100>Not applicable for Arista/Drivenet equipment. |
| OrgGroup | O | A25 | 1 | No trailing blanks | <288315>  As of now after discussion, there is no current plan to use by CM for this project.  This may be used by EM.  <297100>Not applicable for Arista/Drivenet equipment. |
| ManageOrg | O | A11 | 1 | No trailing blanks | GRDB  GRDB\_ROUTER.local\_support\_id  Derive the ManageOrg as follows:  Use first 7 characters of local\_support\_id  <297100>Not applicable for Arista/Drivenet equipment. |
| ActiveOrg | O | A12 | 1 | No trailing blanks | GRDB  GRDB\_ROUTER.local\_support\_id |
| FunctionalArea | O | A12 | 1 | No trailing blanks | GRDB  GRDB\_CMP\_ORD.gems\_schema\_id |
| CustIPAddress | O | A15 | 1 | No trailing blanks | INSTAR  SERIAL\_IP\_ADDR(1).ip\_address |
| ProviderCircuit | O | A32 | 1 | No trailing blanks | GRDB  GRDB\_CIRCUIT.telco\_ckt\_num  <297100>Not applicable for Arista/Drivenet equipment. |
| PTTProvide | O | A30 | 1 | Blanks | GRDB  GRDB\_CIRCUIT.telco\_provider  <297100>Not applicable for Arista/Drivenet equipment. |
| requestEquipShelf | O | Varchar2(10) | 1 | No trailing blanks | <297409-CR173072> Shelf for input device (which can be PE or gateway) related to this customer record. |
| requestEquipSlot | O | Varchar2(5) | 1 | No trailing blanks | <297409-CR173072> Slot for input device (which can be PE or gateway) related to this customer record. |
| requestEquipPort | O | Varchar2(5) | 1 | No trailing blanks | <297409-CR173072> Port for input device (which can be PE or gateway) related to this customer record. |
| <CpeDetailsList> | O | Complex  Array | 0:1 |  | <289116.140768-US636605>   * This list/array is to hold CPE records for this customer (customer access circuit). * For 1607 release, only uCPE data is supported. * See below for structure definition. |
| Customer Information Response | R | N/A |  | End Tag |  |
| Body | R | N/A | 1 | EndTag |  |

|  |  |
| --- | --- |
| IDEAL COURSE | |
| # | Step Description |
| 1. | Use Case begins |
| 2. | DBOR received query getCustInfoQuery request from AOTS with SERVICE TYPE = ‘GMIS’  Set Input\_Change ID = AOTS. Change ID  Set Input\_CmdcTransID = AOTS. CmdcTransID  Set Input\_ EquipName = AOTS.EquipName  If AOTS.SLOT and/or AOTS.PORT are populated:  Set Input\_Slot = SLOT from AOTS  Set Input\_Port = PORT from AOTS  <297409-workitem-281399>  For serviceType=VPLS, this GMIS logic flow is executed to support VPLS as dedicated PE for MIS.  Before this step, the logic flow for serviceType=FR-ATM should have been executed, which   * May have created a <ACC-CKT-List>. If this list is not blank, then it needs be used here for the MIS data retrieval. * Set the **CustomerRecordCnt** already. This value is used in this flow to set the Customer\_Record\_Selector |
| 2A | <Step added for 297100>  Check input.EquipmentName using INSTAR db  select equip\_model.ptnii\_equip\_code from equipment equip,equip\_model  where  equip.equip\_mod\_id = equip\_model.equip\_mod\_id  and equipment.ptnii\_equip\_name=<input\_EquipmentName>  If the result return contains the equip\_model.ptnii\_equip\_code in ('mee','sw8’) then return the customer record and return only fields that are applicable for 297100.Refer to AID doc for fields applicable.  Please see <2971001-Logic to get Customer details for Arista Leaf/Drivenet VPE)  Else  Follow the BAU logic and return the customer record. |
| 3 | Retrieve Customer records for input parameters, from INSTAR (GCP replication) tables and views:  <288315> Add new fields  circuitIdBmpFormat  bvoipIndicator  <297409-CR173072>  Add new shelf/slot/port fields to the Select clause.  For 10/40/100 Gige MIS, VPLS-PE is dedicated device to customer, same as PE (follow existing logic).  Set TypeOfService = ‘GMIS’  **Retrieve from GCP INSTAR tables and views:**  **if the input device had an associated IPv6 ipaddress**  Select  BIDS\_CUSTOMER.cust\_name AS customerName,  PREMISE.address  PREMISE.city  PREMISE.state\_abbr  PREMISE.country\_name  CUST\_ACCESS.acc\_ckt AS circuitID,  Decode (Cust\_access.acc\_ckt,  Instr (Cust\_access.acc\_ckt, ‘.’, 1, 1) = 3, ‘Y’,  Instr (Cust\_access.acc\_ckt, ‘.’, 1, 1) = 5, ‘Y’,  ‘N’ ) cktConversionFlag, <288315> Used to mark whether the circuit format can be converted.  SITE.loc\_id AS locationID, ---<297100 not applicable for Arista leaf/Drivenet VPE>  SITE.site\_id AS siteID, ---<297100 not applicable for Arista leaf/Drivenet VPE>  IP\_SERV\_ACC\_TYPE.acc\_type---<297100 not applicable for Arista leaf/Drivenet VPE>  IP\_SERV\_ACC\_SPEED.speed\_value AS portSpeed, ---<297100 not applicable for Arista leaf/Drivenet VPE>  FACILITY\_ACCOUNT.cust\_id AS customerID,  FACILITY\_ACCOUNT.mcn AS customerMcn,  FACILITY\_ACCOUNT.grc,  NVL(SERIAL\_IP\_ADDR(1).ip\_address, IPV6\_ASSIGNED\_LINK\_IPS(1).ipv6\_address\_compress)  AS customerIpAddress, /\* customer router ip address) \*/ <288315>  NVL(SERIAL\_IP\_ADDR(2).ip\_address, IPV6\_ASSIGNED\_LINK\_IPS(2).ipv6\_address\_compress)  AS ipAddress /\* access router ip address \*/ <288315>---<297100 not applicable for Arista leaf/Drivenet VPE>  COMPONENT.mod(component1.port,1000) AS port, /\*Returns the remainder \*/---<297100 not applicable for Arista leaf/Drivenet VPE>  COMPONENT.beg\_slot AS slot---<297100 not applicable for Arista leaf/Drivenet VPE>  EQUIPMENT.ptnii\_equip\_name AS equipmentName,  Decode (NVL(IP\_SERVICE\_OPTIONS.option\_name, ‘-‘) <288315-upd4)  ‘Plus’, ‘Y’,  ‘N’) AS managedIndicator---<297100 not applicable for Arista leaf/Drivenet VPE>  <297409-CR173072>  COMPONENT.shelf AS requestEquipShelf --<297100 not applicable for Arista leaf/Drivenet VPE>  COMPONENT.beg\_slot AS requestEquipSlot --<297100 not applicable for Arista leaf/Drivenet VPE>  COMPONENT.mod(component1.port,1000) AS requestEquipPort, ---<297100 not applicable for Arista leaf/Drivenet VPE>  </297409-CR173072>  From  BIDS\_CUSTOMER  PREMISE  CUST\_ACCESS  SITE  IP\_SERV\_ACC\_TYPE  IP\_SERV\_ACC\_SPEED  FACILITY\_ACCOUNT  SERIAL\_IP\_ADDR(1)  SERIAL\_IP\_ADDR(2)  IP\_PORT\_ASGMT  COMPONENT  IP\_ASSIGNED\_ENDPT  EQUIPMENT  IPV6\_PORT\_ASGMT\_MAP(1),  IPV6\_ASSIGNED\_LINK\_IPS(1)  IPV6\_PORT\_ASGMT\_MAP(2),  IPV6\_ASSIGNED\_LINK\_IPS(2),  SELECTED\_OPTS\_REL, <288315-upd4>  IP\_SELECTED\_OPTS,  IP\_SERVICE\_OPTIONS  Match  EQUIPMENT.ptnii\_equip\_name = <Input\_EquipName>  and equipment.equip\_id = ip\_assigned\_endpt.equip\_id  and ip\_assigned\_endpt.equip\_id = component.equip\_id  and COMPONENT.beg\_slot = <Input\_Slot > (use only when AOTS.Slot is populated)  and mod(COMPONENT.port,1000) = <input\_port> (use only when AOTS.PORT is populated)  and ip\_assigned\_endpt.endpoint\_id = ip\_port\_asgmt.ip\_endpt  and ip\_port\_asgmt.cr\_addr\_id = serial\_ip\_addr(1).serial\_ip\_addr\_id (+) /\* cust rout ip\*/  and ip\_port\_asgmt.ar\_addr\_id = serial\_ip\_addr(2).serial\_ip\_addr\_id (+) /\* access rout ip \*/  and ip\_port\_asgmt.serv\_acc\_pt\_id = ip\_serv\_acc\_pt.serv\_acc\_pt\_id  and ip\_serv\_acc\_pt.account\_id = facility\_account.account\_id (+) /\* may not have facility data \*/  and ip\_serv\_acc\_speed.serv\_acc\_speed\_id = ip\_serv\_acc\_pt.serv\_acc\_speed\_id  and site.site\_id = ip\_serv\_acc\_pt.site\_id  and cust\_access.site\_id (+) = site.site\_id /\* customer may not require access \*/  and premise.loc\_id = site.loc\_id =  and bids\_customer.cust\_id = premise.cust\_id  <288315> Add Ipv6 if Ipv4 doesn’t exist  And IP\_PORT\_ASGMT.sdid = IPV6\_PORT\_ASGMT\_MAP(1).sdid (+)  And IPV6\_PORT\_ASGMT\_MAP(1). Ipv6\_link\_ip\_id = IPV6\_ASSIGNED\_LINK\_IPS(1).ipv6\_link\_ip\_id  And IPV6\_PORT\_ASGMT\_MAP(1). IPV6\_ADDRESS\_TYPE\_ID=’10’ (Ipv6\_CR Address)  And IP\_PORT\_ASGMT.sdid = IPV6\_PORT\_ASGMT\_MAP(2).sdid (+)  And IPV6\_PORT\_ASGMT\_MAP(2). Ipv6\_link\_ip\_id = IPV6\_ASSIGNED\_LINK\_IPS(2).ipv6\_link\_ip\_id  And IPV6\_PORT\_ASGMT\_MAP(2). IPV6\_ADDRESS\_TYPE\_ID=’9’ (Ipv6\_AR Address)  And IP\_PORT\_ASGMT.serv\_acc\_pt\_id = SELECTED\_OPTS\_REL.serv\_acc\_pt\_id (+) <288315-upd4>  And SELECTED\_OPTS\_REL.selected\_opt\_id = IP\_SELECTED\_OPTS.selected\_opt\_id (+)  And IP\_SELECTED\_OPTS.service\_option\_id = IP\_SERVICE\_OPTIONS.service\_option\_id (+)  And IP\_SERVICE\_OPTIONS.option\_name = ‘Plus’ (+)  <297409-workitem-281399>   * Add below condition to exclude Access circuits already retrieved in FR-ATM flow. * Meadure needs to be taken to avoid performance degradation.   And CUST\_ACCESS.acc\_ckt Not IN (<ACC-CKT-List>)  **288315 - QC 64083 and 65990 - The original query for GMIS service - returns data if the input device had an associated IPv4 address**  **Union**  Retrieve from GCP INSTAR tables and views:  Select  BIDS\_CUSTOMER.cust\_name AS customerName,  PREMISE.address  PREMISE.city  PREMISE.state\_abbr  PREMISE.country\_name  CUST\_ACCESS.acc\_ckt AS circuitID,  Decode (Cust\_access.acc\_ckt,  Instr (Cust\_access.acc\_ckt, ‘.’, 1, 1) = 3, ‘Y’,  Instr (Cust\_access.acc\_ckt, ‘.’, 1, 1) = 5, ‘Y’,  ‘N’ ) cktConversionFlag, <288315> Used to mark whether the circuit format can be converted.  SITE.loc\_id AS locationID, ---<297100 not applicable for Arista leaf/Drivenet VPE>  SITE.site\_id AS siteID, ---<297100 not applicable for Arista leaf/Drivenet VPE>  IP\_SERV\_ACC\_TYPE.acc\_type---<297100 not applicable for Arista leaf/Drivenet VPE>  IP\_SERV\_ACC\_SPEED.speed\_value AS portSpeed, ---<297100 not applicable for Arista leaf/Drivenet VPE>  FACILITY\_ACCOUNT.cust\_id AS customerID,  FACILITY\_ACCOUNT.mcn AS customerMcn,  FACILITY\_ACCOUNT.grc,  SERIAL\_IP\_ADDR(1).ip\_address AS customerIpAddress, /\* customer router ip address) \*/  SERIAL\_IP\_ADDR(2).ip\_address AS ipAddress /\* access router ip address \*/ ---<297100 not applicable for Arista leaf/Drivenet VPE>COMPONENT.mod(component1.port,1000) AS port, /\*Returns the remainder \*/---<297100 not applicable for Arista leaf/Drivenet VPE>  COMPONENT.beg\_slot AS slot---<297100 not applicable for Arista leaf/Drivenet VPE>  EQUIPMENT.ptnii\_equip\_name AS equipmentName,  Decode (NVL(IP\_SERVICE\_OPTIONS.option\_name, ‘-‘) <288315-upd4)  ‘Plus’, ‘Y’,  ‘N’) AS managedIndicator---<297100 not applicable for Arista leaf/Drivenet VPE>  <297409-CR173072>  COMPONENT.shelf AS requestEquipShelf --<297100 not applicable for Arista leaf/Drivenet VPE>  COMPONENT.beg\_slot AS requestEquipSlot --<297100 not applicable for Arista leaf/Drivenet VPE>  COMPONENT.mod(component1.port,1000) AS requestEquipPort, ---<297100 not applicable for Arista leaf/Drivenet VPE>  </297409-CR173072>  From  BIDS\_CUSTOMER  PREMISE  CUST\_ACCESS  SITE  IP\_SERV\_ACC\_TYPE  IP\_SERV\_ACC\_SPEED  FACILITY\_ACCOUNT  SERIAL\_IP\_ADDR(1)  SERIAL\_IP\_ADDR(2)  IP\_PORT\_ASGMT  COMPONENT  IP\_ASSIGNED\_ENDPT  EQUIPMENT  Match  EQUIPMENT.ptnii\_equip\_name = <Input\_EquipName>  and equipment.equip\_id = ip\_assigned\_endpt.equip\_id  and ip\_assigned\_endpt.equip\_id = component.equip\_id  and COMPONENT.beg\_slot = <Input\_Slot > (use only when AOTS.Slot is populated)  and mod(COMPONENT.port,1000) = <input\_port> (use only when AOTS.PORT is populated)  and ip\_assigned\_endpt.endpoint\_id = ip\_port\_asgmt.ip\_endpt  and ip\_port\_asgmt.cr\_addr\_id = serial\_ip\_addr(1).serial\_ip\_addr\_id (+) /\* cust rout ip\*/  and ip\_port\_asgmt.ar\_addr\_id = serial\_ip\_addr(2).serial\_ip\_addr\_id (+) /\* access rout ip \*/  and ip\_port\_asgmt.serv\_acc\_pt\_id = ip\_serv\_acc\_pt.serv\_acc\_pt\_id  and ip\_serv\_acc\_pt.account\_id = facility\_account.account\_id (+) /\* may not have facility data \*/  and ip\_serv\_acc\_speed.serv\_acc\_speed\_id = ip\_serv\_acc\_pt.serv\_acc\_speed\_id  and site.site\_id = ip\_serv\_acc\_pt.site\_id  and cust\_access.site\_id (+) = site.site\_id /\* customer may not require access \*/  and premise.loc\_id = site.loc\_id =  and bids\_customer.cust\_id = premise.cust\_id  <2971001-Logic to get Customer details for Arista Leaf/Drivenet VPE customer info)—  Changes added based on the information from INSTAR.    If the inut equipment\_ptnii\_code in (sw8,mee)  SELECT DISTINCT upper(E.ptnii\_equip\_name),  B.cust\_name,  P.address,  P.city,  P.state\_abbr,  P.country\_name,  C.acc\_ckt circuitID,  S.loc\_id,  S.site\_id,  FA.cust\_id,  FA.mcn,  FA.grc,  FA.soc,  '' AS ip\_address\_1,  '' AS ip\_address\_2,  E.ptnii\_equip\_name,  DECODE (Instr (c.acc\_ckt,'.', 1, 1) , 3, 'Y', 5, 'Y', 'N' ) cktConversionFlag  FROM  BIDS\_CUSTOMER B,  PREMISE P,  CUST\_ACCESS Ca,  SITE S,  IP\_SERV\_ACC\_SPEED IPSA,  FACILITY\_ACCOUNT FA,  IP\_PORT\_ASGMT IPP,  IP\_ASSIGNED\_ENDPT IPAE,  IP\_SERV\_ACC\_PT IIPS,  EQUIPMENT E,  equip\_model em  WHERE  upper(E.ptnii\_equip\_name) =<input.equipment\_ptnii\_name>  AND E.equip\_id = IPAE.equip\_id  AND IPAE.endpoint\_id = IPP.ip\_endpt  AND IPP.serv\_acc\_pt\_id = IIPS.serv\_acc\_pt\_id  AND IIPS.account\_id = FA.account\_id (+)  AND IPSA.serv\_acc\_speed\_id = IIPS.serv\_acc\_speed\_id  AND S.site\_id = IIPS.site\_id  AND Ca.site\_id (+) = S.site\_id  AND P.loc\_id = S.loc\_id  AND B.cust\_id =P.cust\_id  AND em.ptnii\_equip\_code in ('mee','sw8')  </297100>  <297409-workitem-281399>   * Add below condition to exclude Access circuits already retrieved in FR-ATM flow. * Meadure needs to be taken to avoid performance degradation.   And CUST\_ACCESS.acc\_ckt Not IN (<ACC-CKT-List>)  <289116.140768-US636605>  **For each Customer record retrieved above:**  call GetUcpeByCircuitID (cust\_access.acc\_ckt) and populate CpeDetailsList  <288315>  Convert the acc\_ckt (in ICORE format) to normalized/BMP format.  This is don’t only for domestic circuit. For international circuit, leave it as is.  If cktConversionFlag = Y, then  Call Common logic –CircuitFormatConversion\_ICORE\_to\_BMP (<acc\_ckt>) -> circuitIdBmpFormat  Else  Set circuitIdBmpFormat = <acc\_ckt>  End If  ~~<288315> Derive bvoipIndiactor from ICORE data – For each record retrieved above:~~  ~~Select service\_option.\*~~  ~~From~~  ~~service\_asgmt,~~  ~~service\_option~~  ~~Where service\_asgmt.site\_id = <site\_id>~~  ~~And service\_asgmt.cust\_id = <cust\_id>~~  ~~And service\_asgmt.serv\_opt\_id = service\_option.serv\_opt\_id~~  ~~And upper(service\_option.serv\_opt) = ‘BVOIP’~~  ~~If this returned data, then~~  ~~Set bvoipIndicator = ‘Y’,~~  ~~Else~~  ~~Set bvoipIndicator = ‘N’,~~  ~~End If~~  If data found:  \*Retrieve all records as described in ‘GMIS query response” below  \*Perform Step 3a to get data from GRDB  <289116.140768-US636605>  call GetUcpeByCircuitID (cust\_access.acc\_ckt) and populate CpeDetailsList  Else if NO data found:  Return “No data found” |
|  | Sample code from Production:  SELECT DISTINCT B.cust\_name                                      ,   P.address                                                      ,   P.city                                                         ,   P.state\_abbr                                                   ,   P.country\_name                                                 ,   C.acc\_ckt                                                      ,   S.loc\_id                                                       ,   S.site\_id                                                      ,   IPSA.speed\_value                                               ,   FA.cust\_id                                                     ,   FA.mcn                                                         ,   FA.grc                                                         ,   NVL(SI.ip\_address, iali1.ipv6\_address\_compress)  AS ip\_address\_1,   NVL(SIA.ip\_address, iali2.ipv6\_address\_compress) AS ip\_address\_2,   mod(C.port,1000) MOD\_PORT                                       ,   C.beg\_slot                                                      ,   E.ptnii\_equip\_name                                              ,   DECODE (Instr (c.acc\_ckt,'.', 1, 1) , 3, 'Y', 5, 'Y', 'N' ) cktConversionFlag    FROM BIDS\_CUSTOMER B       ,   PREMISE P                   ,   CUST\_ACCESS C               ,   SITE S                      ,   IP\_SERV\_ACC\_SPEED IPSA      ,   FACILITY\_ACCOUNT FA         ,   SERIAL\_IP\_ADDR SI           ,   SERIAL\_IP\_ADDR SIA          ,   IP\_PORT\_ASGMT IPP           ,   COMPONENT C                 ,   IP\_ASSIGNED\_ENDPT IPAE      ,   IP\_SERV\_ACC\_PT IIPS         ,   EQUIPMENT E                 ,   IPV6\_PORT\_ASGMT\_MAP ipam1   ,   IPV6\_ASSIGNED\_LINK\_IPS iali1,   IPV6\_PORT\_ASGMT\_MAP ipam2   ,   IPV6\_ASSIGNED\_LINK\_IPS iali2   WHERE upper(E.ptnii\_equip\_name) = 'KC9MO310ME7' AND C.comp\_id                     = IPAE.comp\_id AND E.equip\_id                    = IPAE.equip\_id AND IPAE.equip\_id                 = C.equip\_id AND IPAE.endpoint\_id              = IPP.ip\_endpt AND IPP.cr\_addr\_id                = SI.serial\_ip\_addr\_id(+) AND IPP.ar\_addr\_id                = SIA.serial\_ip\_addr\_id(+) AND IPP.serv\_acc\_pt\_id            = IIPS.serv\_acc\_pt\_id AND IIPS.account\_id               = FA.account\_id (+) AND IPSA.serv\_acc\_speed\_id        = IIPS.serv\_acc\_speed\_id AND S.site\_id                     = IIPS.site\_id AND C.site\_id (+)                 = S.site\_id AND P.loc\_id                      = S.loc\_id AND B.cust\_id                     =P.cust\_id AND ipp.sdid                      = ipam1.sdid(+) AND ipam1.ipv6\_link\_ip\_id         = iali1.ipv6\_link\_ip\_id AND ipam1.ipv6\_address\_type\_id    = '10' AND ipp.sdid                      = ipam2.sdid(+) AND ipam2.ipv6\_link\_ip\_id         = iali2.ipv6\_link\_ip\_id AND ipam2.ipv6\_address\_type\_id    = '9';  Proposed change is to add a union with this query (the pre-October version of the GMIS query) with the above query:  SELECT DISTINCT B.cust\_name                                      ,   P.address                                                      ,   P.city                                                         ,   P.state\_abbr                                                   ,   P.country\_name                                                 ,   C.acc\_ckt                                                      ,   S.loc\_id                                                       ,   S.site\_id                                                      ,   IPSA.speed\_value                                               ,   FA.cust\_id                                                     ,   FA.mcn                                                         ,   FA.grc                                                         ,   SI.ip\_address  AS ip\_address\_1,   SIA.ip\_address AS ip\_address\_2,   mod(C.port,1000) MOD\_PORT                                       ,   C.beg\_slot                                                      ,   E.ptnii\_equip\_name                                              ,   DECODE (Instr (c.acc\_ckt,'.', 1, 1) , 3, 'Y', 5, 'Y', 'N' ) cktConversionFlag    FROM BIDS\_CUSTOMER B       ,   PREMISE P                   ,   CUST\_ACCESS C               ,   SITE S                      ,   IP\_SERV\_ACC\_SPEED IPSA      ,   FACILITY\_ACCOUNT FA         ,   SERIAL\_IP\_ADDR SI           ,   SERIAL\_IP\_ADDR SIA          ,   IP\_PORT\_ASGMT IPP           ,   COMPONENT C                 ,   IP\_ASSIGNED\_ENDPT IPAE      ,   IP\_SERV\_ACC\_PT IIPS         ,   EQUIPMENT E                    WHERE upper(E.ptnii\_equip\_name) = 'KC9MO310ME7' AND C.comp\_id                     = IPAE.comp\_id AND E.equip\_id                    = IPAE.equip\_id AND IPAE.equip\_id                 = C.equip\_id AND IPAE.endpoint\_id              = IPP.ip\_endpt AND IPP.cr\_addr\_id                = SI.serial\_ip\_addr\_id AND IPP.ar\_addr\_id                = SIA.serial\_ip\_addr\_id(+) AND IPP.serv\_acc\_pt\_id            = IIPS.serv\_acc\_pt\_id AND IIPS.account\_id               = FA.account\_id (+) AND IPSA.serv\_acc\_speed\_id        = IIPS.serv\_acc\_speed\_id AND S.site\_id                     = IIPS.site\_id AND C.site\_id (+)                 = S.site\_id AND P.loc\_id                      = S.loc\_id AND B.cust\_id                     =P.cust\_id; |
| 3 a | **For each Customer record retrieved above, Retrieve from DBOR GRDB table and view**:  <288315> Add new fields  bvoipIndicator  managedIndicator  parentOrgGroup ???  Select  Ce.ip\_address  Ce.clli\_code AS clli,  Ce.common\_name AS cerName,  Substr(Ce.local\_support\_id, 1,7) AS manageOrg, --take first 7-char as manageOrg <288315-upd1>  Ce.local\_support\_id AS activeOrg,  ~~Ce.managed\_by AS managedIndicator, <288315>~~ <288315-upd4>  Ce.bvoip\_on\_router AS bvoipIndicator, <288315>---<297100 not applicable for Arista leaf/Drivenet VPE>  ~~Org.org\_acct AS orgGroup,~~ --Remove the first character if it’s a ‘-‘  Decode (substr(Org.org\_acct,1,1),  '-', substr (Org.org\_acct,2),  Org.org\_acct) orgGroup, <288315-upd1>  Org.gems\_schema\_id AS functionalArea,  Ckt.telco\_ckt\_num AS providerCircuit---<297100 not applicable for Arista leaf/Drivenet VPE>  Ckt.telco\_provider AS provider---<297100 not applicable for Arista leaf/Drivenet VPE>  From  grdb\_router ce  grdb\_cmp\_org org  grdb\_accesslink ACC (used to join tables)  grdb\_circuit ckt  Match  and (CE.ip\_address =  <customerIpAddress> -- INSTAR.SERIAL\_IP\_ADDR.ip\_address FROM STEP 3>  Or CE.ipv6\_address =  <customerIpAddress> -- INSTAR.SERIAL\_IP\_ADDR.ip\_address FROM STEP 3> This could be Ipv6  ) <288315>  and CE.odbid\_eqp = ORG.odbid\_cmponent  and ORG.role = 'GEMS ORG'  and CE.odbid\_eqp = ACC.odbid\_base\_eqp1  and ACC.odbid\_config = CKT.odbid\_config |
| 3-a-tkt | **<288315> Note on Ticket 219773863** – GMIS CUSTOMERS ARE MISSING FUNCTIONAL AREA AND ORG GROUP  Currently in AOTS\_CM production some GMIS customers are missing data in the following fields: Functional Area, Managing Org and Active Org Group in AOTS\_CM. (See screens shots from production below). GCP to investigate and identify the mapping issue and solution so that the following fields are populated in AOTS-CM by September 15th, 2016.   * Functional Area = ATT\_IT * Managing Org = IT-ENTRPRS * Active Org = IT-ENTRPRS-SDEL.   Note: The CER Name is also blank. All these fields are from GRDB, with the Customer IP Address as key. |
| 4. | Create Record Selector for each of the records retrieved in step 3. ~~starting with number 1~~  <297409-workitem-281399>  The start value is defined below:  Use the **CustomerRecordCnt** value determined from the FR-ATM flow to set the Customer\_Record\_Selector  If **CustomerRecordCnt** value is not blak/empty, Then  Set Customer\_Record\_Selector = CustomerRecordCnt + 1  Else  ~~Set Customer\_Record\_Selector = Record Selector~~  Set Customer\_Record\_Selector = 1  End If |
| 5 | Create return message with data from step 3/3a as described in GMIS table  Refer to AOTS-DBOR IAD document “WPID\_C3 13\_R14\_AOTSCM-DBOR\_IAD\_V14.04”  If no GRDB data found then return the INSTAR data.  If Input\_ change\_id & cmdc\_Trans\_ID are populated then  Insert retrieved GMISdata from steps 3a, 3b, 3c into table “ACMR5\_CNC\_ALLCUSTOMERSAFFECTED”  Set input data as:  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.REQUEST\_ID = Input\_change\_id  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.cmdc\_transid = Input\_cmdc\_TransID  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.customer\_record\_selector = customer\_record\_selector  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Customer\_ID = Customer ID  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Customer\_Name = CustomerName  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Customer\_MCN = Customer MCN  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.GRC = grc  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Location\_ID = LocationID  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Site\_ID = Site Number  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.IP\_Address = IPAddress  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Equipment\_Name = EquipmentName  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Port = Port  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Slot = Slot  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Circuit\_ID = Circuit ID  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.type\_of\_service = TypeOfService  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Address = Address  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.City = City  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.State = State  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Country = Country  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.CLLI = clli  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.CER\_name = CER name  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Org\_Group = OrgGroup  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Managing\_Org = ManageOrg  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Active\_Org = ActiveOrg  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Functional\_Area = Functional Area  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Cust\_IP\_Address = CustIPAddress  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Provider\_Circuit = ProviderCircuit  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.PTTProvider = PTTProvider  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Circuit\_bmp\_format = circuitBmpFormat <288315>  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Managed\_indicator = managedIndicator <288315>  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Bvoip\_indicator = bvoidIndicator <288315>  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Request\_equip\_shelf = requestEquipShelf <297409.173072>  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Request\_equip\_slot = requestEquipSlot <297409.173072>  ACMR5\_CNC\_ALLCUSTOMERSAFFECTED.Request\_equip\_port = requestEquipPort <297409.173072>  **<288315>**  Insert retrieved CustomerAffected data, same attributes as shown above, to GCP\_cnc\_AllCustomersAffected.  After all the customer records inserted for this transaction, calculate the total number of records been inserted:  Set CustomerRecordCnt = last-record\_selector (Customer\_Record\_Selector)  Update GCP\_cnc\_AllCustomersAffected\_sum table.  **</288315>**  Else  Goto step 7  EndIf |
| 6 | After the data is stored successfully, GCP call AOTS CM notification web service, sending them change\_id & transaction\_id as input parameters, for data that has been stored |
| 7. | Use Case ends |

|  |  |
| --- | --- |
| POST-CONDITIONS | |
| Course | Description |
| 01 | DBOR INSTAR data retrieved successfully |
| 02 | DBOR GRDB data retrieved successfully |
| 03 | DBOR data returned to AOTS successfully |

|  |  |
| --- | --- |
| EXCEPTIONS | |
| Exception ID | Action(s) on Exception |
| 01 | DBOR INSTAR database not available for data retrieval. Create Error “DBOR\_001” |
| 02 | Retrieved data cannot be formatted to return to AOTS. Create Error “DBOR\_002” |

### InquireCustomerCircuitDetailsByNetworkElement for servicetype=PVC-ID – 288315

**Formally known as “GetCustInfoQuery for servicetype=PVC-ID”**

|  |  |
| --- | --- |
| **USE CASE GENERAL INFORMATION** | |
| Use Case Name | PVC-ID Customer Inventory Retrieval |
| Use Case ID | **UC-DBOR-PB932-028** |
| Description | This use case details the customer inventory request from AOTS for PVC data. DBOR will retrieve data from DBOR ICORE database  Input data can be: PVC\_ID and REQUEST\_ID |
| Responsible Analyst | Lan TranVu,Gunjan Gupta |
| Type of Execution | WebService and SQL to Oracle db. |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE REVISION LOG** | | | |
| Reason for Revision: | Adding PVC data | Release Number: |  |
| Author: | Gunjan Gupta,Lan Tran-Vu | Revised: | 02/27/08 |

|  |  |
| --- | --- |
| **ACTORS** | |
| Primary Actors | AOTS |
| Secondary Actors | DBOR |

|  |  |
| --- | --- |
| **PRE-CONDITIONS** | |
| **#** | **Description** |
| 1 | AOTS query |

|  |  |
| --- | --- |
| **IDEAL COURSE** | |
| # | **Step Description** |
| 1. | Use Case begins |
| 2. | DBOR received query getCustInfoQuery request from AOTS with **SERVICE TYPE = PVC\_ID**  Set Input\_Change ID = <AOTS. Change ID>  Set Input\_CmdcTransID = <AOTS. CmdcTransID>  Set Input\_ EquipName = <AOTS.EquipName> This is PVC ID |
| 3.a | **Rtrieve PVC records from ICORE (GCP replication) tables**:  <288315> Adding new fields for both A and Z ends  circuitIdBmpFormat  Retrieve PVC records from GCP(DBOR) ICORE tables: PVC, IPFRA,IPFRZ, VPNA,VPNZ,CUSTOMERA,CUSTOMERZ,CUST\_ACCESSA, CUST\_ACCESSZ, SITEA, SITEZ, PREMISEA, PREMISEZ, PORT\_ASGMT, EQUIPMENT  Match  SITEZ.prem\_loc\_id = PREMISEZ.loc\_id(+) AND SITEA.prem\_loc\_id = PREMISEA.loc\_id And SITEZ.site\_id = PVC.pvc\_rsite\_id(+) AND SITEA.site\_id = PVC.pvc\_lsite\_id And PVC.pvc\_id = IPFRA.pvc\_id(+)  And PVC.pvc\_id = IPFRZ.pvc\_id(+)  And IPFRA.vpn\_id = VPNA.vpn\_id  And IPFRZ.vpn\_id = VPNZ.vpn\_id  And PVC.pvc\_rcust\_id = customerZ.cust\_id  And PVC.pvc\_lcust\_id = customerA.cust\_id  And CUST\_ACCESSZ.cust\_id = CUSTOMERZ.cust\_id(+)  And CUST\_ACCESSA.cust\_id = CUSTOMERA.cust\_id(+)  And PVC.PVC\_ID = Input\_EquipName (this is the input PVC ID)  Retrieve From  **PVC (**PVC\_ID, RSWITCH, RSLOT, RPORT, PVC\_RDLCI, PVC\_RCONTRCIR, PVC\_rVCI, PVC\_rVPI,  LSWITCH, LSLOT, LPORT, PVC\_LDLCI, PVC\_LCONTRCIR, PVC\_lVPI, PVC\_lVCI )  **VPNA**.VPN\_NAME  **VPNZ**.VPN\_NAME  **IPFRA**.VPN\_ID  **IPFRZ**.VPN\_ID  **SITEA** (SITE\_ID, FULL\_PORT\_SPEED , GRC, CLLI)  **SITEZ** (SITE\_ID, FULL\_PORT\_SPEED , GRC, CLLI)  **CUSTOMERA** (CUST\_NAME, CUST\_MCN,CUST\_ID)  **CUSTOMERZ** (CUST\_NAME, CUST\_MCN,CUST\_ID)  **PREMISEA** (LOC\_ID, PREM\_ADDRESS, PREM\_CITY, PREM\_STATE, PREM\_COUNTRY)  **PREMISEZ** (LOC\_ID, PREM\_ADDRESS, PREM\_CITY, PREM\_STATE, PREM\_COUNTRY)  **CUST\_ACCESSA**(ACC\_CKT)  **CUST\_ACCESSZ**(ACC\_CKT )  Decode (Cust\_accessA.acc\_ckt, <288315>  Instr (Cust\_accessA.acc\_ckt, ‘.’, 1, 1) = 3, ‘Y’,  Instr (Cust\_accessA.acc\_ckt, ‘.’, 1, 1) = 5, ‘Y’,  ‘N’ ) cktConversionFlagA, <288315> Used to mark whether the circuit format can be converted.  Decode (Cust\_accessZ.acc\_ckt,  Instr (Cust\_accessZ.acc\_ckt, ‘.’, 1, 1) = 3, ‘Y’,  Instr (Cust\_accessZ.acc\_ckt, ‘.’, 1, 1) = 5, ‘Y’,  ‘N’ ) cktConversionFlagZ, <288315> Used to mark whether the circuit format can be converted.  <288315>  Convert the acc\_ckt (in ICORE format) to normalized/BMP format, based on the cktConversionFlag value.  This is only for domestic circuit. For international circuit, leave it as is.  If cktConversionFlagA = Y, then  Call Common logic –CircuitFormatConversion\_ICORE\_to\_BMP (<A\_acc\_ckt>) -> A\_circuitIdBmpFormat  Else  Set A\_circuitIdBmpFormat = <A\_acc\_ckt>  End If  If cktConversionFlagZ = Y, then  Call Common logic –CircuitFormatConversion\_ICORE\_to\_BMP (<Z\_acc\_ckt>) -> Z\_circuitIdBmpFormat  Else  Set Z\_circuitIdBmpFormat = <Z\_acc\_ckt>  End If |
| 3.b | **For each PVC record, Derive TypeOfService data**  If SITEA.protocol is ‘FR’ (for a\_typeOfService) Or SITEZ.protocol is ‘FR’ (for z\_typeOfService)  If PREMISEA.prem\_country is ‘USA’ (for a\_typeOfService) Or PREMISEZ.prem\_country is ‘USA’ (for z\_typeOfService)  Set TypeOfService = ‘FR’  Else  Set TypeOfService = ‘IFR’  EndIf  EndIf  If SITEA.protocol is ‘ATM’ (for a\_typeOfService) Or SITEZ.protocol is ‘ATM’ (for z\_typeOfService)  If PREMISEA.prem\_country is ‘USA’ (for a\_typeOfService) Or PREMISEZ.prem\_country is ‘USA’ (for z\_typeOfService)  Set TypeOfService = ‘ATM’  Else  Set TypeOfService = ‘IATM’  EndIf  EndIf  If SITEA.protocol is not ‘ATM’ or ‘FR’, Then  Set AtypeOfService = SITEA.protocol  EndIf  If SITEZ.protocol is not ‘ATM’ or ‘FR’, Then  Set ZtypeOfService = SITEZ.protocol  EndIf |
| 3.c | Retrieve from GCP(DBOR) SIDBOR database using ICORE acc\_ckt data from step 3.a  **Input: <acc\_ckt>**  <288315> Adding new fields for both A and Z ends  managedIndicator  parentOrgGroup  **For PVC A end:**  Select  FWTOPOLOGY.local\_router,  ~~ASSETA.src\_org\_id,~~  ASSETA.org\_cd, <288315>  ASSETA.managing\_org\_cd,  ASSETA.active\_org\_cd,  ASSETA.functional\_area,  ~~ASSETA.service,~~  Decode (Upper(ASSETA.service), <288315-US681337-upd4>  ‘IPEFX’, ‘BVOIP’,  ‘IPEFR’, ‘IPFR’,  ‘FRAME’, ‘FR’,  ASSETA.service) AS service,  CLIENT\_ORG.org\_cd\_prnt AS parentOrgGroup, <288315>  ‘Y’ managedIndicator, <288315>  From  FWTOPOLOGY,  ASSET ASSETA  CLIENT\_ORG  Where FWTOPOLOGY.local\_access\_ckt = <acc\_ckt>  And FWTOPOLOGY.functional\_area = ASSETA.functional\_area  And FWTOPOLOGY.local\_router = ASSETA.asset\_nm  And ASSETA.org\_cd = CLIENT\_ORG.org\_cd  And ASSETA.functional\_area = CLIENT\_ORG.functional\_area  If data found  Set A CERName = FWTOPOLOGYA.local\_router  ~~Set A OrgGroup = ASSETA.src\_org\_id~~  Set A OrgGroup = ASSETA.org\_cd, <288315>  Set A ManageOrg = ASSETA. Managing\_org\_cd  Set A ActiveOrg = ASSETA. Active\_org\_cd  Set A FUNCTIONALAREA= ASSETA. Functional\_area  Set A Service = ASSETA. Service  Set A parentOrgGroup = blank <288315>  Set A managedIndocator = managedIndicator <288315>  End If  **For PVC Z end:**  Select  FWTOPOLOGY.local\_router,  ~~ASSETZ.src\_org\_id,~~  ASSETZ.org\_cd, <288315>  ASSETZ.managing\_org\_cd,  ASSETZ.active\_org\_cd,  ASSETZ.functional\_area,  ~~ASSETZ.service,~~  Decode (Upper(ASSETZ.service), <288315-US681337-upd4>  ‘IPEFX’, ‘BVOIP’,  ‘IPEFR’, ‘IPFR’,  ‘FRAME’, ‘FR’,  ASSETZ.service) AS service,  CLIENT\_ORG.org\_cd\_prnt AS parentOrgGroup, <288315>  ‘Y’ managedIndicator, <288315>  From  FWTOPOLOGY,  ASSET ASSETZ  CLIENT\_ORG  Where FWTOPOLOGY.local\_access\_ckt = <acc\_ckt>  And FWTOPOLOGY.functional\_area = ASSETZ.functional\_area  And FWTOPOLOGY.local\_router = ASSETZ.asset\_nm  And ASSETZ.org\_cd = CLIENT\_ORG.org\_cd  And ASSETZ.functional\_area = CLIENT\_ORG.functional\_area  If data found  Set A CERName = FWTOPOLOGYA.local\_router  ~~Set Z OrgGroup = ASSETZ.src\_org\_id~~  Set Z OrgGroup = ASSETZ.org\_cd, <288315>  Set Z ManageOrg = ASSETZ.managing\_org\_cd  Set Z ActiveOrg = ASSETZ.active\_org\_cd  Set Z FUNCTIONALAREA= ASSETZ.functional\_area  Set Z Service = ASSETZ. Service  Set Z parentOrgGroup = blank <288315>  Set Z managedIndocator = managedIndicator <288315>  End If |
| 3.d | NOTE: this step is commented out until further development with CCI  Retrieve from GCP(DBOR) CCI database using ICORE customer\_mcn data from step 3.a  Match CUSTOMERA.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  Match CUSTOMERZ.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  and CUSTOMERA.cust\_mcn (pos. 7-9) =MCN\_ENTRPRS\_XREF.MCN\_SFX  and CUSTOMERZ.cust\_mcn (pos. 7-9) =MCN\_ENTRPRS\_XREF.MCN\_SFX    and XREF\_MCN\_ENTRPRS.ENT\_PTY\_ID = PARTY.ENTRPRS\_PTY\_ID  and PARTY.DUNS\_NB = DNB.DUNS\_NB  Retrieve DNB.(DUNS\_NB, PRNTHQ\_BIZ\_NM)  Set CCI = DUNS\_NB  Set PARENT NAME = PRNTHQ\_BIZ\_NM  Else  IF CUSTOMERA.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  CUSTOMERZ.cust\_mcn (pos. 1-6) = MCN\_ENTRPRS\_XREF.MCN\_NB  and XREF\_MCN\_ENTRPRS.ENT\_PTY\_ID = PARTY.ENTRPRS\_PTY\_ID  and PARTY.DUNS\_NB = DNB.DUNS\_NB  Retrieve DNB.(DUNS\_NB, PRNTHQ\_BIZ\_NM)  Set CCI = DUNS\_NB  Set PARENT NAME = PRNTHQ\_BIZ\_NM    Else  Set CCI = blank  Set PARENT NAME = blank  EndIf  EndIf |
| 4. | If Input AOTS.SERVICE TYPE is ‘PVC-ID’  Return record count of all data found in step 3.a and 3.b as described in “PVC-ID Record Count response”  Else  Return all records as described in “PVC-ID query response” .  EndIf |
| 4.a | Create a Customer\_Record\_Selector for each of the records retrieved from step 3 , 3.a & 3.b for each change Request ID starting from 10000. |
| 4.b | <288315>  Call ~~UC-DBOR-PB932-19~~ Common logic – Load\_Change\_PVC: To store retrieved PVC data in table Change\_PVC |
| 4.b.2 | <288315> Generate new record and load or update the Summary table  Execute common logic: Load\_GCP\_AllCustomersAffected\_sum |
| 4.b.3 | <288315> ETL  Create PVCRecord File |
| 4.c | If more then one PVC record is retrieved, GCP will only send first row of record when Component Type = “ PVC ID” .?? |
| 5. | Provide all tables/columns described in “PVC-ID QUERY RESPONSE” in BO for reporting |
| 6. | Use Case ends |

|  |  |
| --- | --- |
| **POST-CONDITIONS** | |
| **Course** | **Description** |
| 01 | DBOR ICORE data retrieved successfully |
| 02 | DBOR data returned to AOTS successfully |

|  |  |
| --- | --- |
| **EXCEPTIONS** | |
| **Exception ID** | **Action(s) on Exception** |
| 01 | Ideal Course Step 3 – DBOR ICORE database not available for data retrieval. Create Error “DBOR\_001” |
| 02 | Ideal Course Step 4 – Retrieved data cannot be formatted to return to AOTS. Create Error “DBOR\_002” |

### Tracking Table and Error Handling – <288315-US681337-upd2>

This table is populated during and after processing Client request for particular Change\_ID and CMDC\_TransID, for tracking purpose.

**This table is part of Detai Design and is defined in Detail Design document. So remove its definition from this HLD.**

**~~Table Name: GCP\_ETL\_FILECR\_INDTR~~**

|  |  |  |  |
| --- | --- | --- | --- |
| **~~Field Name~~** | **~~Type/Length~~** | **~~Required, Optional~~** | **~~Comments~~** |
| ~~GCP\_ETL\_FILECR\_INDTR\_ID~~ | ~~Number(22)~~ | ~~R~~ | ~~PK - oracle seq generated val~~ |
| ~~CMDC\_TransID~~ | ~~VARCHAR2(15)~~ | ~~R~~ |  |
| ~~change\_id~~ | ~~VARCHAR2(15)~~ | ~~R~~ |  |
| ~~CustFileName~~ | ~~VARCHAR2(255)~~ | ~~O~~ | ~~File name is Unique~~ |
| ~~PVCFileName~~ | ~~VARCHAR2(255)~~ | ~~O~~ | ~~File name is Unique~~ |
| ~~CustProcessFlag~~ | ~~VARCHAR2(50)~~ | ~~O~~ | ~~Success/Error(By ETL) - Web service must populate Q after uploading the tables~~ |
| ~~PVCFProcessFlag~~ | ~~VARCHAR2(50)~~ | ~~O~~ | ~~Success/Error(By ETL) - Web service must populate Q after uploading the tables~~ |
| ~~ProcessStatusFlag~~ | ~~VARCHAR2(50)~~ | ~~O~~ | ~~Success/Error(By ETL)~~  ~~Values in ProcessStatusFlag:~~   * ~~SP\_Processing\_Initiated - SP updates to this value as soon as API request calls the SP~~ * ~~WS\_Processing\_Initiated - web service(WS) updates to this value after WS sends synchronus response~~ * ~~WS\_Processing\_Completed - WS updates to this value after loading the tables and after setting CustProcessFlag & PVCFProcessFlag to Q~~ * ~~ETL\_Processing\_Initiated - ETL updates to this value when it starts file creation~~ * ~~ETL\_File\_Generation\_completed - ETL updates to this value when files have been generated~~ * ~~ETL\_Processing\_Completed - ETL updates to this value when files have been placed in data router~~ * ~~ETL\_DataRouter\_ERROR - ETL updates to this value if max retry has been exceeded while transferring the file to DR.~~ * ~~The Error code are listed in below table.~~  |  |  |  |  |  | | --- | --- | --- | --- | --- | | **~~Source~~** | **~~SOAP Fault Target Document~~** | | | | | ~~faultCode~~ | ~~faultCode~~ | ~~faultActor~~ | ~~responseCode~~ | ~~responseDescription~~ | | ~~100-199~~ | ~~soap:Server~~ | ~~CSI~~ | ~~100~~ | ~~Insufficient permissions~~ | | ~~200-299~~ | ~~soap:Server~~ | ~~CSI~~ | ~~200~~ | ~~Service not available~~ | | ~~300-399~~ | ~~soap:Client~~ | ~~Client~~ | ~~300~~ | ~~Data Error~~ | | ~~400-499~~ | ~~soap:Client~~ | ~~Client~~ | ~~400~~ | ~~Schema Error~~ | | ~~900~~ | ~~soap:Client~~ | ~~Client~~ | ~~900~~ | ~~Unknown Error~~ | |
| ~~CustProcessErrorCode~~ | ~~VARCHAR2(10)~~ | ~~O~~ | ~~This value can be set by any team when exception occurs in their respective processing~~ |
| ~~CustProcessErrorMessage~~ | ~~VARCHAR2(4000)~~ | ~~O~~ | ~~This value can be set by any team when exception occurs in their respective processing~~ |
| ~~PVCProcessErrorCode~~ | ~~VARCHAR2(10)~~ | ~~O~~ | ~~This value can be set by any team when exception occurs in their respective processing~~ |
| ~~PVCProcessErrorMessage~~ | ~~VARCHAR2(4000)~~ | ~~O~~ | ~~This value can be set by any team when exception occurs in their respective processing~~ |
| ~~Create\_date~~ | ~~Date~~ | ~~R~~ | ~~last\_update\_date - DATE - not null~~ |
| ~~Last\_update\_date~~ | ~~Date~~ | ~~R~~ | ~~date and time to be populated when record is inseted/updated~~ |
| ~~Retry\_count~~ | ~~Number(2)~~ | ~~O~~ | ~~can be used incase of file transfer retry has to be done from ETL to DataRouter. After configured max retry is reached then no more retry.~~ |

In case of error conditions that the data retrieval process failed, corresponding Error code and description will be populated in the “ResponseInfo” structure in the data files (in XML format). Error codes are specified in the below table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Source** | **SOAP Fault Target Document** | | | |
| faultCode | faultCode | faultActor | responseCode | responseDescription |
| 100-199 | soap:Server | CSI | 100 | Insufficient permissions |
| 200-299 | soap:Server | CSI | 200 | Service not available |
| 300-399 | soap:Client | Client | 300 | Data Error |
| 400-499 | soap:Client | Client | 400 | Schema Error |
| 900 | soap:Client | Client | 900 | Unknown Error |

The ResponseInfo structure is defined as:

<xs:complexType name="ResponseInfo">

<xs:sequence>

<xs:element name="code" type="xs:string"/>

<xs:element name="description" type="xs:string"/>

</xs:sequence>

Incase of NO Data Found for the input criteria:   
  
- ETL will generate the XML in the following formate.  
<?xml version="1.0" encoding="utf-8"?>  
<InquireCustomerCircuitDetailsByNetworkElementCustResponse>  
<Response>  
<code>300</code>  
<description>No Data Found</description>  
</Response>  
</InquireCustomerCircuitDetailsByNetworkElementCustResponse>  
  
  
System/Oracle Error/Exception occurred during the retrieval:

- ETL will generate the XML in the following formate.   
<?xml version="1.0" encoding="utf-8"?>  
<InquireCustomerCircuitDetailsByNetworkElementCustResponse>  
<Response>  
<code>400</code>  
<description>Can't connect to DataBase</description>  
</Response>  
</InquireCustomerCircuitDetailsByNetworkElementCustResponse>  
  
Time interval that CM needs wait before next try:  
- This depends on the volume. As of now we can say **30 Minutes** CM needs to wait before next try.

### Common logic – GetUcpeByCircuitID – <289116.140768-US636605> <301033>

<301033>

* Re-engineer EDF logic to use the A&AI data sourced from DMaaP instead of the A&AI batch feed.
* Support Vyatta uCPE.

|  |  |  |
| --- | --- | --- |
| **Response Fields** | **Source Tables** | **Conditions, Comments** |
|  | Step UCPE-110 to 130  Data source: A&AI | **For input Access Circuit ID**  Retrieve UCPE hardware, software components, and linked real router.  VNF - Virtual Network Function  UCPE – Universal Customer Premise Equipment   |  |  | | --- | --- | | Equip Type | Description | | FW | vFW | | RT | Virtual router: vSRX, vCSR | | SW | JVM VNF | | WX | WANX VNF <288361> | | ZZ | JDM VNF | | MRS | Real Router | | AVPN-M | Real Router | |
| **UCPE record:**  Pserver.pserver\_id  Pserver.hostname: deviceName,  pserver.equip\_type: deviceType  pserver.equip\_model: deviceModel,  pserver.equip\_vendor: equipmentVendor,  pserver.management\_option: managedBy,  If the value is null, set to ‘ATT’  pComplex.physical\_location\_id: deviceClli, | **Data source: A&AI**  aai\_pserver pserver,  aai\_pserver\_complex pComples,  aai\_physical\_link plink,  aai\_p\_interface pinterface,  aai\_p\_interface\_physical\_link pipLink | **Step UCPE-110 -** Retrieve UCPE hardware attributes  **Input:** <accessCircuit>  Note:  New table created in AAI schema containing equipment types for UCPE (including Juniper and Vyatta UCPEs), so no hardcode needed.  ucpe\_equip\_type\_set.ucpe\_equip\_type  Conditions:  plink.circuit\_id = <accessCircuit>  And pserver.hostName = pipLink.hostName  And pipLink.hostName = pinterface.hostName  And pipLink.interface\_name = pinterface.interface\_name  And pipLink.link\_name = plink.link\_name  And pserver.equip\_type IN  (ucpe\_equip\_type\_set.ucpe\_equip\_type)  And pserver.hostname = pComplex.hostname  Use <pserver\_id> for the uCPE retrieved above,   * Continue on Step UCPE-120 to retrieve Real router attributes, if it exists * Continue on Step UCPE-130 to retrieve Logical asset details |
| **Real router:**  Pnf.pnf\_name: deviceName,  Pnf.equip\_type: deviceType  Pnf.equip\_model: deviceModel,  Pnf.equip\_vendor: equipmentVendor,  Pnf.management\_option: managedBy,  If the value is null, set to ‘ATT’  pComplex.physical\_location\_type: deviceClli, | **Data source: A&AI**  aai\_pserver pserver,  aai\_pnf pnf,  aai\_p\_interface pinterface1,  aai\_p\_interface pinterface2,  aai\_p\_interface\_physical\_link pip\_link1,  aai\_p\_interface\_physical\_link pip\_link2  aai\_pnf\_complex pnfComples, | **Step UCPE-120 - Retrieve Real router attributes**  Input: <pserver\_id> from Step UCPE-110  Search path: pserver > p\_interface > physical\_link > p\_interface > pnf  Logic: pserver’s p\_interface and pnf’s p\_interface share the same physical link  Conditions:  // Map the pserver to the input ucpe hostname  pserver.pserver\_id = <pserver\_id>  And pserver.hostname = pinterface1.hostname  And pinterface1.hostname = pip\_link1.hostname  And pinterface1.interface\_name = pip\_link1.interface\_name  And pip\_link1.link\_name = pip\_link2.link\_name  // Relate to real router – pnf  And pinterface2.pnf\_name = pip\_link2.pnf\_name  And pinterface2.interface\_name = pip\_link2.interface\_name  And pinterface2.pnf\_name = pnf.pnf\_name  // Relate pnf to complex  And pnf.pnf\_id = pnfComplex.pnf\_id |
| **Logical Asset:**  **From Step UCPE-130-A**  Vnf.vnf\_id  **From Step UCPE-130-B:**  vnf.vnf\_name - deviceName,  ‘VNF-‘ + vnf.vnf\_type: deviceType  ‘’: deviceModel,  vImage.application\_vendor: equipmentVendor,  vnf.management\_option: managedBy,  If the value is null, set to ‘ATT’  deviceClli (from uCPE record) | **Data source: A&AI**  **For Step UCPE-130-B**  aai\_generic\_vnf vnf,  aai\_pserver pserver,  aai\_generic\_vnf\_pserver vnfPserver  aai\_generic\_vnf vnf,  aai\_vnf\_image vImage  aai\_generic\_vnf vnf,  aai\_pserver pserver,  aai\_generic\_vnf\_vserver vnfVserver,  aai\_vserver vserver,  aai\_vserver\_pserver vpServer  aai\_generic\_vnf vnf,  aai\_vnf\_image vImage  aai\_generic\_vnf vnf,  aai\_pserver pserver,  aai\_generic\_vnf\_nos\_server vnServer,  aai\_nos\_server nosServer,  aai\_nos\_server\_pserver nosPserver  aai\_generic\_vnf vnf,  aai\_vnf\_image vImage  **~~For Step UCPE-130-C~~**  ~~aai\_generic\_vnf vnf,~~  ~~aai\_vnf\_image vImage~~ | **Step UCPE-130 - Retrieve logical asset attributes**  Input: <pserver\_id> from Step UCPE-110  Search path:  Pserver > generic\_vnf  Pserver > vserver > generic\_vnf  Pserver > nos\_server > generic\_vnf  **Step UCPE-130-A: Getting all the VNF IDs for the uCPE:**  Conditions:  Search path: Pserver > generic\_vnf  pserver.pserver\_id = <pserver\_id>  And vnfPserver.hostname = pserver.hostName  And vnfPserver.vnf\_id = vnf.vnf\_id  And vnf.vnf\_type in ('ZZ', 'SW', 'RT', ‘FW’, ‘WX’)  And vvImage.vnf\_id (+) = vnf.vnf\_id  And vvImage.vnf\_image\_uuid = vImage.vnf\_image\_uuid (+)  Search path: Pserver > vserver > generic\_vnf  Union  pserver.pserver\_id = <pserver\_id>  And vnfVserver.vnf\_id = vnf.vnf\_id  And vnfVserver.vserver\_id = vserver.vserver\_id  And vnfVserver.vserver\_id = vpServer.vserver\_id  And vpserver.hostname = pserver.hostName  And vnf.vnf\_type in ('ZZ', 'SW', 'RT', ‘FW’, ‘WX’)  And vvImage.vnf\_id (+) = vnf.vnf\_id  And vvImage.vnf\_image\_uuid = vImage.vnf\_image\_uuid (+)  Search path: Pserver > vserver > generic\_vnf  Union  pserver.pserver\_id = <pserver\_id>  And pserver.hostname = nosPserver.hostname  And nosPserver.nos\_server\_id = vnServer.nos\_server\_id  And vnServer.vnf\_id = vnf.vnf\_id  And vnf.vnf\_type in ('ZZ', 'SW', 'RT', ‘FW’, ‘WX’)  And vvImage.vnf\_id (+) = vnf.vnf\_id  And vvImage.vnf\_image\_uuid = vImage.vnf\_image\_uuid (+)  **~~Step UCPE-130-B: Getting VNF attributes:~~**  ~~Input: <vnf\_id> list~~  ~~Search path: generic\_vnf > vnf\_image~~  ~~Conditions:~~  ~~vnf.vnf\_id in (<vnf-ld>)~~  ~~And vvImage.vnf\_id (+) = vnf.vnf\_id~~  ~~And vvImage.vnf\_image\_uuid = vImage.vnf\_image\_uuid (+)~~ |

**<301033> Remove the below logic using AAI Feed files, replace it with logic using AAI tables from DMaap**

|  |  |
| --- | --- |
| ~~Step UCPE-110~~ | **~~For input Access Circuit ID~~**  ~~Retrieve UCPE hardware attributes~~  ~~Data source: A&AI~~ |
| **~~UCPE record:~~**  ~~deviceName,~~  ~~deviceClli,~~  ~~deviceModel,~~  ~~equipmentVendor,~~  ~~managedBy,~~  ~~deviceType~~ | **~~<289116.140768-US636605>~~****~~Retrieve and populate UCPE hardware attributes~~**  **~~Input:~~** ~~<accessCircuit>~~  ~~Data source: A&AI~~  ~~Remove Primary key restriction from Device table:~~   * ~~For AAI Device table, the Primary key restriction is removed. So there can be records with duplicate ptnii\_name values.~~ * ~~Make sure only one physical component record be retrieved.~~ * ~~There’s no specific guidelines/rules which fields will be identical for duplicate ptnii\_name values, so the logic will just pick up the first record.~~   ~~Select Unique~~  ~~device.ptnii\_name AS deviceName, --this is <ucpeHostName>, used to retrieve logical assets~~  ~~device.location\_clli AS deviceClli,~~  ~~device.equip\_vendor\_model AS deviceModel,~~  ~~device.equip\_vendor AS equipmentVendor,~~  ~~NVL (device.managed\_by, ‘ATT’) AS managedBy,~~  ~~‘UCPE-‘ + device.equip\_type AS deviceType~~  ~~From~~  ~~instar\_staging.Instar\_aai\_device device,~~  ~~instar\_staging.Instar\_aai\_interface interface~~  ~~Where 1=1~~  ~~And interface.cust\_access\_ckt = <accessCircuit>~~  ~~And device.ptnii\_name = interface.ptnii\_name~~  ~~And device.equip\_type = ‘JUNIPER UCPE’~~  ~~Continue on Step UCPE-120 to retrieve Real router attributes, if it exists~~  ~~Continue on Step UCPE-130 to retrieve Logical asset details~~ |
| **~~Step UCPE-120~~** | **~~Retrieve Real router attributes~~** |
| **~~Real router:~~**  ~~deviceName,~~  ~~deviceClli,~~  ~~deviceModel,~~  ~~equipmentVendor,~~  ~~managedBy,~~  ~~deviceType~~ | **~~<289116.140768-US636605> Retrieve and populate Real router attributes~~**  ~~Input: <ucpeHostName> from Step UCPE-110~~  ~~Data source: A&AI~~  ~~Select dev.ptnii\_name AS realRouterPtnii~~  ~~device.ptnii\_name AS deviceName,~~  ~~device.location\_clli AS deviceClli,~~  ~~device.equip\_vendor\_model AS deviceModel,~~  ~~device.equip\_vendor AS equipmentVendor,~~  ~~NVL (device.managed\_by, ‘ATT’) AS managedBy,~~  ~~‘RR-‘ + device.equip\_type AS deviceType~~  ~~From~~  ~~Instar\_aai\_dualEnded dend,~~  ~~Instar\_aai\_device dev~~  ~~Where 1=1~~  ~~And dend.a\_ptnii\_name = <ucpeHostName> -- uCPE~~  ~~And dend.z\_ptnii\_name = dev.ptnii\_name – Real router~~  ~~And dev.equip\_type IN (‘MRS’, ‘AVPN-M’)~~ |
| **~~Step UCPE-130~~** | **~~Retrieve logical asset attributes~~** |
| **~~Logical Asset:~~**  ~~deviceName,~~  ~~deviceClli,~~  ~~deviceModel,~~  ~~equipmentVendor,~~  ~~managedBy,~~  ~~deviceType~~ | **~~<289116.140768-US636605> Retrieve and populate logical asset attributes~~**  ~~Input: <ucpeHostName> from Step UCPE-110~~  ~~Data source: A&AI~~  ~~Remove Primary key restriction from Device table:~~   * ~~For AAI Device table, the Primary key restriction is removed. So there can be records with duplicate ptnii\_name values.~~ * ~~There’s no specific guidelines/rules which fields will be identical for duplicate ptnii\_name values, so ALL the logical component records will be returned. No logic to screen out the duplicate logical records with same ptnii\_name values.~~   ~~Select Unique~~  ~~device.ptnii\_name AS deviceName, --this is <physicalPtnii>, used to retrieve logical assets~~  ~~device.location\_clli AS deviceClli,~~  ~~device.equip\_vendor\_model AS deviceModel,~~  ~~device.equip\_vendor AS equipmentVendor,~~  ~~NVL (device.managed\_by, ‘ATT’) AS managedBy,~~  ~~‘VNF-‘ + device.equip\_type AS deviceType~~  ~~From~~  ~~instar\_staging.Instar\_aai\_device device,~~  ~~Instar\_aai\_virtualRelationship vRel1~~  ~~Where 1=1~~  ~~And vRel1.Dependent\_on\_name = <ucpeHostName>~~  ~~And device.ptnii\_name = vRel1.name~~  ~~And device.equip\_type in ('ZZ', 'SW', 'RT', ‘FW’, ‘WX’)~~  ~~Union~~  ~~Select Unique~~  ~~device.ptnii\_name AS deviceName, --this is <physicalPtnii>, used to retrieve logical assets~~  ~~device.location\_clli AS deviceClli,~~  ~~device.equip\_vendor\_model AS deviceModel,~~  ~~device.equip\_vendor AS equipmentVendor,~~  ~~NVL (device.managed\_by, ‘ATT’) AS managedBy,~~  ~~‘VNF-‘ + device.equip\_type AS deviceType~~  ~~From~~  ~~instar\_staging.Instar\_aai\_device device,~~  ~~Instar\_aai\_virtualRelationship vRel1,~~  ~~Instar\_aai\_virtualRelationship vRel2~~  ~~Where 1=1~~  ~~And vRel1.Dependent\_on\_name = <ucpeHostName>~~  ~~And vRel2.Dependent\_on\_name = vRel1.Name~~  ~~And device.ptnii\_name = vRel2.name~~  ~~And device.equip\_type in ('ZZ', 'SW', 'RT', ‘FW’, ‘WX’)~~ |

### Common logic – convert circuit format: CircuitFormatConversion\_ICORE-to-BMP <288315>

|  |  |
| --- | --- |
|  | Conversion Rules to convert ICORE to BMPNormalized format  Reformat input cktID to BMP format.  Input: ICORE Circuit ID  Output: BMP Normalized Circuit ID  // Positions are referenced in parenthesis e.g. (3) is position 3 in the input circuit ID.  <288315-QC73580>   * Issue: BMP format circuit values are incorrect. * Fix: no HLD update.   E.G.:  Incorrect value:’^^DHEC^^^973^^^801^ATI ‘  Correct value: ‘^^DHEC^^^973801ATI^’ where ^=space.  Set PREFIX = input cktID (1-2) or ‘ ‘  If input cktID(3) is not ‘.’  PREFIX = ‘^^‘ (2 blank spaces)  cktType = "PREFIX"  ELSE  PREFIX = input cktID(1-2)  cktType = "NoPREFIX"  Note: Strip leading zeros, left justify the prefix value and pad to the right with spaces (e.g. “01” = “1 “)  // The following steps assume that either the PREFIX was included in the ICORE Circuit ID or it wasn't included. Thus, two sets of character positions are mentioned. The first, when PREFIX is present (e.g. 23.HFGJ) and the second, when PREFIX is not present (e.g. HFGJ)  Set SVCCD = input cktID(4-5) or input cktID(1-2)  If cktType = "PREFIX"  SVCCD = input cktID (4-5)  Else  SVCCD = input cktID (1-2)  Note: Left justify and pad to the right with spaces (e.g. "H" = "H ")  Set SVCMOD = input cktID(6-7) or input cktID(3-4)  If cktType = "PREFIX"  SVCMOD = input cktID (6-7)  Else  SVCMOD = input cktID (3-4)  Note: Left justify and pad to the right with spaces (e.g. "G" = "G ")  Set SERNO = input cktID (9-14) or input cktID (6-11)  If cktType = "PREFIX"  SERNO = input cktID (9-14)  Else  SERNO = input cktID (6-11)    Note: Right justify and pad to the left with spaces (e.g. "007041 = " 7041")  Incorrect value:’^^DHEC^^^973^^^801^ATI ‘  Correct value: ‘^^DHEC^^^973801ATI^’ where ^=space.  Set SUFFIX = input cktID (16-18) or input cktID (13-15)  If cktType = "PREFIX" AND no alphanumeric characters are found after the SERNO "." delimiter (e.g. 007041..SUV )  SUFFIX = " " (3 blank spaces)  Else If cktType = "PREFIX"  SUFFIX = input cktID (16-18)  Else  SUFFIX = input cktID (13-15)    Note: Right justify and pad to the left with zeros (0). For blank SUFFIX value, pad 3 characters with spaces (e.g. "2" = "002", <blank> = " ")  Set CO = input cktID (20-23), input cktID (17-20), input cktID (14-17)  If cktType = "PREFIX" AND no alphanumeric characters are found after the SERNO "." delimiter (e.g. 007041..SUV )  CO = input cktID (14-17)  If cktType = "PREFIX"  CO = input cktID (20-23)  Else  CO = input cktID (17-20)  Set BMP reformated cktID = PREFIX + SVCCD + SVCMOD + SERNO + SUFFIX + CO |

|  |  |
| --- | --- |
| 9 | <288315>  Call ~~UC-DBOR-PB932-19~~ Common logic – Load\_Change\_PVC: To store retrieved PVC data in table Change\_PVC |
| 9.b | <288315> Generate new record and load or update the Summary table  Execute common logic: Load\_GCP\_AllCustomersAffected\_sum |

### Common logic – Load\_Change\_PVC <288315>

|  |  |
| --- | --- |
| **Change Area** | **Comments and Logic** |
| Store retrieved PVC records into to:    ARS Table Change\_PVC  CBUS Table: gcp\_change\_pvc | **<288315>**  **This step is executed when PVC records are retrieved from DB and ready to load in to table - Both ChangeId and CmdcTransID are in the input.**  **Input: PvcInfo structure containing PVC records.**  **Create Record Selector** for each of the PVC records  **<288315> Note:**   * CustomerRecordSelector is a GCP generated field that is unique for each CmdcTransID. The RecordSelector needs to be unique under each CMDCTransID (so unique under each equipment). * The value must be same for each record whether it’s sent back to Client (in file) or insert into table. * So it cannot always start with number 1 for a new transaction.   For each new transaction:   * Check whether records exist in the GCP\_change\_pvc table for the same CMDCTransID/EquipmentName (use GCP\_AllCustomersAffected\_sum table for simplicity). * If it does, determine start point for the RecordSelector for this transaction. * Else, use 1 as start point for the RecordSelector for this transaction.   Select pvc\_record\_count AS max\_cust\_record  From GCP\_AllCustomersAffected\_sum  Where Change\_id= <Input\_changeID>  And Equipment\_name = CustomerInfo.EquipmentName – from STEP-CUST-1  If such max\_record exists, then  Set pvc\_record\_selector\_base = max\_cust\_record + 1  Else  ~~Set pvc\_record\_selector\_base = 1~~  Set pvc\_record\_selector\_base = 10000 <CR-162323> to be in sync with existing setting for pvc\_record\_selectot  End If  Starting the Record Selector with **pvc\_record\_selector\_base**, and incrementing by 1 at a time for each record.  **Set Pvc\_Record\_Selector = Record Selector**  **Insert retrieved data into table “ Change\_pvc”**  **Set input data as:**  Change\_pvc.Change\_request\_id = Input\_changeID  Change\_pvc.cmdc\_transid = Input\_cmdcTransID  Change\_pvc.pvc\_record\_selector = pvc\_record\_selector  Change\_pvc.customer\_record\_selector = PvcInfo.customer\_record\_selector  Change\_pvc.pvc\_type = PvcInfo.pvcType  Change\_pvc.pvc\_id = PvcInfo.pvcId  Change\_pvc.a\_cust\_ip\_address = Pvc.PvcAEnd.custIpAddress  Change\_pvc.a\_ip\_address = Pvc.PvcAEnd.ipAddress  Change\_pvc.a\_cust\_name = Pvc.PvcAEnd.customerName  Change\_pvc.a\_mcn = Pvc.PvcAEnd.customerMcn  Change\_pvc.acust\_id = Pvc.PvcAEnd.customerID  Change\_pvc.ackt\_id = Pvc.PvcAEnd.circuitId  Change\_pvc.a\_vpn\_id = Pvc.PvcAEnd.vpnId  Change\_pvc.a\_vpn\_name = Pvc.PvcAEnd.vpnName  Change\_pvc.a\_vpn\_mcn = Pvc.PvcAEnd.vpnMcn  Change\_pvc.a\_vpn\_cust\_id = Pvc.PvcAEnd.vpnCustId  Change\_pvc.a\_site\_id = Pvc.PvcAEnd.siteId  Change\_pvc.a\_grc = Pvc.PvcAEnd.grc  Change\_pvc.a\_cer\_name = Pvc.PvcAEnd.cerName  Change\_pvc.a\_functional\_area = Pvc.PvcAEnd.functionalArea  Change\_pvc.a\_org\_group = Pvc.PvcAEnd.activeOrg  Change\_pvc.a\_manageOrg = Pvc.PvcAEnd.manageOrg  Change\_pvc.a\_typeOfService = Pvc.PvcAEnd.typeOfService  Change\_pvc.a\_service = Pvc.PvcAEnd.service  Change\_pvc.a\_restriction\_type= Pvc.PvcAEnd.restrictionType  ~~Change\_pvc.a\_provider = Pvc.PvcAEnd.provider~~ <288315-US681337-upd1>  ~~Change\_pvc.a\_equip\_name = Pvc.PvcAEnd.equipmentName~~  Change\_pvc.a\_cci = Pvc.PvcAEnd.cci  Change\_pvc.a\_parent\_name = Pvc.PvcAEnd.parentCompanyName  Change\_pvc.a\_slot = Pvc.PvcAEnd.slot  Change\_pvc.a\_port = Pvc.PvcAEnd.portNumber  Change\_pvc.a\_cir = Pvc.PvcAEnd.cir  Change\_pvc.a\_dlci = Pvc.PvcAEnd.dlci  Change\_pvc.a\_end\_vci = Pvc.PvcAEnd.endVci  Change\_pvc.a\_end\_vpi = Pvc.PvcAEnd.endVpi  Change\_pvc.a\_port\_speed = Pvc.PvcAEnd.portSpeed  Change\_pvc.a\_location\_id = Pvc.PvcAEnd.locationId  Change\_pvc.a\_clli = Pvc.PvcAEnd.clli  Change\_pvc.a\_address = Pvc.PvcAEnd.address  Change\_pvc.a\_city = Pvc.PvcAEnd.city  Change\_pvc.a\_state = Pvc.PvcAEnd.state  Change\_pvc.a\_country = Pvc.PvcAEnd.country  Change\_pvc.a\_parent\_org\_group = Pvc.PvcAEnd.parentOrgGroup  Change\_pvc.a\_circuit\_bmp\_format= Pvc.PvcAEnd.circuitBmpFormat  Change\_pvc.a\_managed\_indicator = Pvc.PvcAEnd.managedIndicator  Change\_pvc.z\_cust\_ip\_address = Pvc.PvcZEnd.custIpAddress  Change\_pvc.z\_ip\_address = Pvc.PvcZEnd.ipAddress  Change\_pvc.z\_cust\_name = Pvc.PvcZEnd.customerName  Change\_pvc.z\_mcn = Pvc.PvcZEnd.customerMcn  Change\_pvc.zcust\_id = Pvc.PvcZEnd.customerID  Change\_pvc.zckt\_id = Pvc.PvcZEnd.circuitId  Change\_pvc.z\_vpn\_id = Pvc.PvcZEnd.vpnId  Change\_pvc.z\_vpn\_name = Pvc.PvcZEnd.vpnName  Change\_pvc.z\_vpn\_mcn = Pvc.PvcZEnd.vpnMcn  Change\_pvc.z\_vpn\_cust\_id = Pvc.PvcZEnd.vpnCustId  Change\_pvc.z\_site\_id = Pvc.PvcZEnd.siteId  Change\_pvc.z\_grc = Pvc.PvcZEnd.grc  Change\_pvc.z\_cer\_name = Pvc.PvcZEnd.cerName  Change\_pvc.z\_functional\_area = Pvc.PvcZEnd.functionalArea  Change\_pvc.z\_org\_group = Pvc.PvcZEnd.activeOrg  Change\_pvc.z\_manageOrg = Pvc.PvcZEnd.manageOrg  Change\_pvc.z\_typeOfService = Pvc.PvcZEnd.typeOfService  Change\_pvc.z\_service = Pvc.PvcZEnd.service  Change\_pvc.z\_restriction\_type= Pvc.PvcZEnd.restrictionType  ~~Change\_pvc.z\_provider = Pvc.PvcZEnd.provider~~ <288315-US681337-upd1>  ~~Change\_pvc.z\_equip\_name = Pvc.PvcZEnd.equipmentName~~  Change\_pvc.z\_cci = Pvc.PvcZEnd.cci  Change\_pvc.z\_parent\_name = Pvc.PvcZEnd.parentCompanyName  Change\_pvc.z\_slot = Pvc.PvcZEnd.slot  Change\_pvc.z\_port = Pvc.PvcZEnd.portNumber  Change\_pvc.z\_cir = Pvc.PvcZEnd.cir  Change\_pvc.z\_dlci = Pvc.PvcZEnd.dlci  Change\_pvc.z\_end\_vci = Pvc.PvcZEnd.endVci  Change\_pvc.z\_end\_vpi = Pvc.PvcZEnd.endVpi  Change\_pvc.z\_port\_speed = Pvc.PvcZEnd.portSpeed  Change\_pvc.z\_location\_id = Pvc.PvcZEnd.locationId  Change\_pvc.z\_clli = Pvc.PvcZEnd.clli  Change\_pvc.z\_address = Pvc.PvcZEnd.address  Change\_pvc.z\_city = Pvc.PvcZEnd.city  Change\_pvc.z\_state = Pvc.PvcZEnd.state  Change\_pvc.z\_country = Pvc.PvcZEnd.country  Change\_pvc.z\_parent\_org\_group = Pvc.PvcZEnd.parentOrgGroup  Change\_pvc.z\_circuit\_bmp\_format= Pvc.PvcZEnd.circuitBmpFormat  Change\_pvc.z\_managed\_indicator = Pvc.PvcZEnd.managedIndicator  **<288315>**  Insert retrieved Change\_PVC data, same attributes as shown above, to GCP\_Change\_pvc  After all the PVC records inserted for this tranasction, calculate the total number of records been inserted:  <252833.83800-upd3>  Set PvcRecordCnt = the last-record\_selector (Pvc\_Record\_Selector) - 10000 <CR-162323>  **</252833.83800>** |

### Common logic – Load\_GCP\_AllCustomersAffected\_sum <288315>

|  |  |
| --- | --- |
| **Change Area** | **Comments and Logic** |
| Generate new record and load or update the Summary table | **<288315> Generate new record and load or update the Summary table**  **Create summary record for this ChangeID/Equipment and load into or update the summary table:**  Check the summary table for the ChangeID and EquipmentName. If they exist, then update the record. Otherwise, insert the new record:  Select ocunt(\*) as sum\_count  From Update GCP\_AllCustomersAffected\_sum  Where Change\_id = <Input\_Change\_ID>,  And Equipment\_name = CustomerInfo.EquipmentName  If sum\_count > 0, then  Update GCP\_AllCustomersAffected\_sum  Set  Customer\_record\_count = CustomerRecordCnt,  Pvc\_record\_count = PvcRecordCnt,  Date\_Record\_added = date()  Where Change\_id = <Input\_Change\_ID>,  And Equipment\_name = CustomerInfo.EquipmentName  Else  Insert into GCP\_AllCustomersAffected\_sum  Set  Record\_id = Oracle Sequence,  Change\_id = <Input\_Change\_ID>,  Customer\_record\_count = CustomerRecordCnt,  Pvc\_record\_count = PvcRecordCnt,  Equipment\_name = CustomerInfo.EquipmentName<252833.83800-upd2>  Date\_Record\_added = date() (Oracle Date function)  End ID |
|  |  |

## Alternative Designs

N/A.

No alternative designs will be considered beyond the design already in production.

## Assumptions/Risks

N/A – Nothing changed here

## Traceability Matrix

|  |  |
| --- | --- |
| Requirement ID | Design Element Identifier |
|  |  |

## Pre-Production Disaster Recovery Planning

N/A – Nothing changed here

## Other Plans and References

N/A – Nothing changed here

|  |
| --- |
|  |

## 

## Acceptance & Approvals

Overview

The Approvers of this work product agree that this document is acceptable and complete to the best of their knowledge and will be used by the project team as an official deliverable for the project. It is further agreed that this document can now be baselined and any changes to these sections from this point forward must follow the Managing Change in the IT UP.

Embed evidence of approval in the review table below, or use the PRISM Approval Functionality in the Project Workflow Module Workflow Template View.

Approvers

|  |  |  |  |
| --- | --- | --- | --- |
| **ATTUID and Name** | **Role** | **Group/Application** | **Version Approved, Approval Date and Approval Evidence** |
| mh6892 – Ming Ho | Artifact Owner - Application Architect | GCP | see PRISM for evidence of approval |
|  | Artifact Contributor | GCP  Note: is approving for both the GCP Dev Mgr and GCP Test Mgr | see PRISM for evidence of approval |
|  | Lead PM |  | see PRISM for evidence of approval |
|  |  |  |  |