Document Attributes

| Attribute | Value |
| --- | --- |
| Application ID /  Application Name | MOTS ID: 14724  PRISM ID: 13367  Application Name: GCP |
| Owner | Hans-Dieter Cordes |
| Owner Contact Information | Phone: +49 691 530 6235  Attuid: hc2164 |
| Contributors | * Joern Franz for general design * Jim Tung for data loading |
| PRISM ID | 258491 (2015-02) |
| PRISM ID(s) for predecessor | 259118 (2014-10)  254035c (2014-10)  254035b (2014-06)  254035a (2014-02)  232213h (2013-04)  232213g (2013-02)  232213f (2012-12)  232213e (2012-10)  232213 (2012-06)  232213a (2012-04) |
| Release Month | October 2014 |
| Scope | Database  ETL processing |

|  |  |
| --- | --- |
| GCP internal inbound interface(s) | Database access to:   1. (new) |
| GCP internal outbound interface(s) | Database access to:   1. (new) |
| GCP external inbound interface(s) | Database access to:   1. (new) |
| GCP external outbound interface(s) | Database access to:   1. (new) |

Revision History

The following table lists the revision history of this document:

| Author | Date | Version # | Revision Description |
| --- | --- | --- | --- |
| Hans-Dieter Cordes | 2013-07-10 | 0.01 | 254035: Initial version. |
| Hans-Dieter Cordes | 2013-08-19 | 0.02 | Added functionality;  Added details |
| Hans-Dieter Cordes | 2013-08-27 | 0.03 | Changed/corrected minor things during HLD review |
| Hans-Dieter Cordes | 2013-09-02 | 0.04 | Added inclusion of all new “GDB” tables to the “GDB\_HIST” schema;  added a “port-to-site” relationship under “HLD-254035-GCP-FLOW-ICORE-201-D [Icore Data Flow into Port to Site]”;  changed DDL file for “GDB.asset\_ext\_equipment.ddl.sql”: added column “host\_name” |
| Hans-Dieter Cordes | 2013-09-05 | 0.05 | Changed “pvc\_id” to “network\_connection\_id” for “GDB.ASSET\_EXT\_NETWORK\_CONNECTION”;  changed DDL file for “GDB.asset\_ext\_network\_connection.ddl.sql”: renamed column “pvc\_id” to “network\_connection\_id” & changed data type;  added requirements for maintaining MCN/GRC/SOC mapping for ICORE;  changed logic for loading customer data;  added the MCN/GRC/SOC triplet as a second customer identifier (having their own records in GDB) besides “customer.cust\_id”;  added requirements for associations to “organization” based on MCN/GRC/SOC triplets |
| Hans-Dieter Cordes | 2013-09-09 | 0.06 | Completely reworked the “PORT”-related stuff:  Changed GDB.PORT table layout;  Changed PORT loading stuff;  Changed PORT relationships/associations;  Removed requirements for (1) COS on PORT and (2) Interface Name on PORT; |
| Hans-Dieter Cordes | 2013-10-02 | 0.07 | Made the asset to port relationship a foreign key relationship in GDB.PORT;  Made the port to location relationship a foreign key relationship in GDB.PORT;  Added a new table “GDB.CIRCUIT\_SERVICE\_CODES” which contains “description” and “speed” data per circuit service code; used this table to load data into “GDB.ASSET\_EXT\_ACCESS\_CIRCUIT”;  Replaced embedded EXCEL file “service-data-mapping.xls” by “service-data-mapping.for-icore.xls”: |
| Hans-Dieter Cordes | 2013-10-25 | 0.08 | Made some corrections (typos etc.) while working on the “IDIS” load HLD (corrections were caused by findings in the “IDIS” load HLD) |
| Hans-Dieter Cordes | 2013-11-07 | 0.09 | Added new “log” and “phy” port GDB tables; updated other GDB DDL files; added “log” and “phy” port loading; made corrections to loading pieces to bring in sync with changed table layouts |
| Hans-Dieter Cordes | 2013-11-12 | 0.10 | Changed these SQL script files:   * DDL.table.mcn\_grc\_soc\_source\_codes.sql * DDL.table.mcn\_grc\_soc\_sites\_ids.sql * DDL.table.mcn\_grc\_soc\_circuit\_ids.sql |
| Hans-Dieter Cordes | 2013-11-18 | 0.11 | Some minor corrections after HLD review sessions finished;  added “GDB.site.ddl.sql”;  added “Primary-Keys-for-254035a.partial.xls”; |
| Hans-Dieter Cordes | 2013-11-28 | 0.12 | (1) exchanged DDL file for phy & log port;  (2) invalidated requirements:  HLD-254035-GCP-FLOW-ICORE-110-D [Icore Data Flow into Phy Port SITE (“Interface Name” data)]  HLD-254035-GCP-FLOW-ICORE-110-E [Icore Data Flow into Phy Port SITE (“id\_asset Equipment” data)]  HLD-254035-GCP-FLOW-ICORE-120-D [Icore Data Flow into Log Port SITE (“Interface Name” data)]  HLD-254035-GCP-FLOW-ICORE-120-E [Icore Data Flow into Log Port SITE (“id\_asset Equipment” data)];  (3) rearranged "Phy Port" and "Log Port" requirements, so "Phy Port" ones go in front of "Log Port" ones;  (4) added logic to link "log ports" to their "phy ports"  (5) renamed requirements due to conflict with "INSTAR" HLD:  HLD-254035-GCP-GDB-for-ICORE-001 [SERVICE\_TYPE Content and Mapping]  HLD-254035-GCP-GDB-for-ICORE-002 [CHANGE\_SYSTEM & CHANGE\_USER Content] |
| Hans-Dieter Cordes | 2013-12-02 | 0.13 | Corrected “PREMISE.zip” (now “PREMISE.prem\_zip”) and “SITE.loc\_id” (now “SITE.prem\_loc\_id”) |
| Hans-Dieter Cordes | 2013-12-05 | 0.14 | Added the setting of “GDB.LOG\_PORT.icore\_pvc\_id\_value” for support of “port unification” (e.g. see “GCP-SA-HLD-for-GCP-GDB.254035a.For-NC3+PLATINUM+SrvMgr-Load.docx”) |
| Hans-Dieter Cordes | 2013-12-12 | 0.15 | Corrected column names for “SITE” and “PREMISE” ICORE tables |
| Hans-Dieter Cordes | 2013-12-18 | 0.16 | Corrected the DDL file “GDB.circuit\_service\_codes.ddl.sql”. |
| Hans-Dieter Cordes | 2014-02-14 | 0.17 | After call with DEV + TEST team improved some wording in the contect of “l” and “r” cust id handling and in the context of choosing a value/item if more than one are returned; made “identifier” loading data more explicit in case data is missing |
| Hans-Dieter Cordes | 2014-03-13 | 0.18 | Added details on req. 130 (Data Flow into Service) |
| Hans-Dieter Cordes | 2014-03-25 | 0.19 | Replaced the 2 GDB port table DDL files by slightly corrected ones. |
| Hans-Dieter Cordes | 2014-05-12 | 1.01 | Corrected phy and log port loading by completely rewriting the corr. HLD requirements/sections. |
| Hans-Dieter Cordes | 2014-06-25 | 1.02 | Completely removed loading of COS data for ports, as it will no longer be used (COS data has been located under “access circuit” detail (COS Package) and “network connection” detail (COS e/ingressProfile) |
| Hans-Dieter Cordes | 2014-07-01 | 1.03 | Corrected/exchanged the DDL files for ICORE “DDL.view.network\_connection.sql” (small typo) and “DDL.view.log\_port\_pe\_vw.sql” (two incorrect where-statements dropped). |
| Hans-Dieter Cordes | 2014-07-07 | 2.01 | Started work on project 254035c;  Changes for 254035c can be identified by:  (1) a requirements number of HLD-254035c  (2) a text of  <BEGIN 254035c.CR-xxxxxx> ...   <END 254035c.CR-xxxxxx>  See also the section “**A Note on how to use this HLD**” |
| Hans-Dieter Cordes | 2014-07-08 | 2.02 | Corrected a typo under req. “HLD-254035-GCP-FLOW-ICORE-204 [Icore Data Flow into Phy Port (“Access Circuit Port”) to Asset]” (changed “SITE.Site\_Id” to “Cust\_Access.Site\_Id”) |
| Hans-Dieter Cordes | 2014-07-16 | 2.03 | Made minor updates/changes during the HLD review on 2014-07-16. |
| Hans-Dieter Cordes | 2014-07-23 | 2.04 | Added a “database traversal” section for CUST\_ACCESS\_CKT for “PE” port under req. “HLD-254035-GCP-FLOW-ICORE-204” |
| Hans-Dieter Cordes | 2014-08-07 | 2.04 | For req. "HLD-254035c-GCP-GDB-012 [ROLE Content and Mapping]" inserted an updated EXCEL file |
| Hans-Dieter Cordes | 2014-08-14 | 2.05 | Added req. “HLD-254035c-GCP-GDB-015” to add “MCN\_GRC” identifier type. |
| Hans-Dieter Cordes | 2014-08-15 | 2.10 | Started work on project 259118;  Changes for 259118 can be identified by:  (1) a requirements number of HLD-259118  (2) a text of  <BEGIN 259118> ...   <END 259118>  See also the section “A Note on how to use this HLD” |
| Hans-Dieter Cordes | 2014-08-19 | 2.11 | Applied minor changes (types etc.) during today’s HLD review |
| Hans-Dieter Cordes | 2014-09-09 | 2.20 | Started work on project 258491;  Changes for 258491 can be identified by:  (1) a requirements number of HLD-258491  (2) a text of  <BEGIN 258491> ...   <END 258491>  See also the section “A Note on how to use this HLD” |
| Hans-Dieter Cordes | 2014-09-29 | 2.30 | Merged requirements from 254035b (as placeholders) into this HLD. |
| Hans-Dieter Cordes | 2014-10-20 | 2.31 | Changed “service options” constants file for GDB (see below);  Changed service option constants from “SO\_...” (2 underscores) to “SO\_...” (1 underscore) |
| Hans-Dieter Cordes | 2014-10-22 | 2.32 | Added a clarification under the requirement “HLD-254035-GCP-FLOW-ICORE-010 [Icore Data Flow: Maintain MCN/GRC/SOC Mapping]”;  Changes can be identified by a text of  <BEGIN ADD-2014-10-22> ...   <END ADD-2014-10-22> |
| Hans-Dieter Cordes | 2014-10-30 | 2.33 | Due to the bug 254035c.UTC-89113 **asset unification for access circuits** had to be introduced that shall work in the same way as the organization unification. In order to be prepared for any future “asset unification” this has been changed:  All associations where assets are involved have been updated to make use of “SOURCE\_KEY\_CORRELATION.id\_source\_key\_master” if applicable (*exception: asset-to-site associations*) to take care of any type of “asset unification”.  Changes can be identified by a text of  <BEGIN 254035c.UTC-89113.003> ...   <END 254035c.UTC-89113.003> |
| Hans-Dieter Cordes | 2014-11-03 | 2.34 | Added “customer-network-to-service” associations, as requested under “Defect R3S1400012499”.  Changes can be identified by a text of  <BEGIN 254035c.DEFECT-12499> ...   <END 254035c.DEFECT-12499> |
| Hans-Dieter Cordes | 2014-11-11 | 2.35 | - Small changes after review on 2014-11-10 on the above defect fix (Defect R3S1400012499)  - Removed reference to SOURCE\_KEY\_MASTER for asset association, “id\_source\_key”, as “id\_source\_key” shall not be impacted by “asset unification” acc. to a discussion with Joern Franz |
| Hans-Dieter Cordes | 2014-11-21 | 2.36 | Added logic for access circuit speed for circuits without a “circuit service code” value.  Changes can be identified by a text of  <BEGIN 254035c.DEFECT-55173> ...   <END 254035c.DEFECT-55173> |
| Hans-Dieter Cordes | 2014-12-09 | 2.37 | Corrected database traversal/constraints logic on req. “HLD-254035-GCP-FLOW-ICORE-206-A” by changing to “outer join”. |
| Hans-Dieter Cordes | 2014-12-11 | 2.38 | Changed logic for associating service data with “customer network” assets of type “layer-2”.  Changes can be identified by a text of  <BEGIN 254035c.DEFECT-R3S1400015110> ...   <END 254035c.DEFECT-R3S1400015110> |
| Hans-Dieter Cordes | 2014-12-15 | 2.39 | Changed logic for associating service data with   * “access circuit” * ”network connection” * “customer network”   assets in general by clearly separating between “layer-2” services (ATM + FR) and “layer-3” services.  Changes can be identified by a text of  <BEGIN 254035c.DEFECT-15114> ...   <END 254035c.DEFECT-15114> |
| Hans-Dieter Cordes | 2015-02-12 | 2.40 | Added fixes for “port unification” deficiencies tracked under ticket 34627  Changes can be identified by a text of  <BEGIN DEFECT-FIX-34627.2015-02-12> ...   <END DEFECT-FIX-34627.2015-02-12> |
| Hans-Dieter Cordes | 2015-02-16 | 2.41 | Applied several changes to make sure “port-to-asset” associations are no longer migrated to the “unification master” asset in order to avoid having more than one port association per asset.  Changes can be identified by a text of  <BEGIN DEFECT-FIX-34627.2015-02-16> ...   <END DEFECT-FIX-34627.2015-02-16> |
| Hans-Dieter Cordes | 2015-02-24 | 2.42 | Applied a small, but important change to the logic for converting the “Suffix” data element when converting to a standard clci circuit format.  Changes can be identified by a text of  <BEGIN Circuit-Conv-Correction.2015-02-24> ...   <END Circuit-Conv-Correction.2015-02-24> |
| Hans-Dieter Cordes | 2015-03-12 | 2.43 | Corrected the logic for loading VPN data for the QC ticket “11688”.  Changes can be identified by a text of  <BEGIN DEFECT-FIX-11688.2015-03-12> ...   <END DEFECT-FIX-11688.2015-03-12> |
| Tofael Khan | 2015-05-19 | 2.44 | Defect 67889 (Releases\_2015\_Jan\_thru\_Apr) – need to remove extraneous ‘space’ characters from address table columns. All changes marked using <Defect 67889> |
| Tofael Khan | 2015-05-20 | 2.45 | Corrected the logic for loading 2 staging tables under requirement "HLD-254035-GCP-FLOW-ICORE-010 [Icore Data Flow: Maintain MCN/GRC/SOC Mapping]".  Changes can be identified by a text of  <BEGIN STAGING-DEFECT-FIX.2015-05-20> ...   <END STAGING-DEFECT-FIX.2015-05-20> |
| Hans-Dieter Cordes (edited by Tofael) | 2015-06-26 | 2.46 | Defect 52461 - MVL VPN related fix – changes are marked by  <BEGIN MVL-VPN-FIX.2015-06-26 Defect 52461>  <END MVL-VPN-FIX.2015-06-26 Defect 52461> |
| Tofael Khan | 2015-07-13 | 3.00 | Updated for 279006 (1510) project – changes tagged with <279006> |
| Hans-Dieter Cordes (edited by Tofael) | 2015-10-07 | 3.01 | LOG-PORT-EKT-FIX:  Created a dedicated view for “log port” objects to avoid data inconsistencies while deleting “network connections” and/or “log ports which had the same “NETWORK\_CONNECTION” view for EKT creation.  Changes are tagged with <LOG-PORT-EKT-FIX> |
| Yun Wan | 2016-01-29 | 3.02 | 286278 changes – tagged with 286278 |
| Yun Wan | 2016-02-05 | 3.03 | 286278 changes – updates after review |
| Yun Wan | 2016-02-08 | 3.04 | 279006 additional changes tagged with <REL1602> |
| Yun Wan | 2016-02-15 | 3.05 | 286278 added ete port key for log port (CE). |
| Yun Wan | 2016-03-08 | 3.06 | 286278 fixed the typo in metadata for PORT\_INFO, it should be LOGICAL\_PORT. |
| Yun Wan | 2016-03-14 | 3.07 | 286278 added logic and data mapping for asset\_ext\_network\_connection.id\_network\_type |
| Tofael Khan (on behalf of Hans-Dieter Cordes) | 2016-03-15 | 3.08 | In respect to excluding INSTAR data for MIS added one where-clause on “ETHERNET”  Changes can be identified by a text of  <BEGIN DEFECT-FIX.INSTAR-MIS-EXCLUSION.2016-03-15> ...   <END DEFECT-FIX.INSTAR-MIS-EXCLUSION.2016-03-15> |
| Yun Wan | 2016-04-02 | 3.09 | Since is\_broadband is mandatory, added logic to populate it with ‘N’ if the source doesn’t have value.  Defect 58004 icore vpn name should match on grid.vpn.assigned\_vpn\_name instead of customer\_vpn\_name. |
| Yun Wan | 04/20/2016 | 3.10 | 286284-US600815-US600161 ete key generation updates |
| Yun Wan | 05/09/2016 | 3.11 | Defect 70085 Fix, added logic to load ORGANIZATION\_IDENTIFER\_INFO |
| Yun Wan | 05/31/2016 | 3.12 | Backout changes for 286284-US600815-US600161, except remove other site’s ete\_site\_key, and other service connection’s ete\_service\_connection\_key, which are required for 270198g. |
| Yun Wan | 06/20/2016 | 3.13 | Defect 22736 fix, added logic to populate account\_usage\_type |
| Yun Wan | 06/24/2016 | 3.14 | Updated ete\_access\_key logic to do circuit id comparison instead of icore site id comparison, and cross out the requirement to wipe out ete access\_key for other circuit ids and ports |
| Yun Wan | 07/01/2016 | 3.15 | Defect 40601 fix, added logic to billing\_account\_number |
| Yun Wan | 07/29/2016 | 3.16 | Bring back 286284-US600815-US600161 ete key generation updates |
| Yun Wan | 08/12/2016 | 3.17 | 286282 US704600 draft |
| Yun Wan | 09/06/2016 | 3.18 | 286282 US704600 added association between inventory mcn/grc/soc to billing mcn/grc/soc. |
| Yun Wan | 10/10/2016 | 3.19 | Defect 47770 update |
| Tofael Khan | 11/10/2016 | 3.20 | 288655a updates – changes tagged with <288655a> |
| Tofael Khan | 11/21/2016 | 3.21 | Updated for CR-155856\_Defect-50973 to create Network VLAN Primary and Backup relationship – changes tagged with <CR-155856\_Defect-50973> |
| Tofael Khan | 12/02/2016 | 3.22 | Update for CR-156194 Defect-69175 to create association between access circuit and managed CSU/DSU – changes tagged with <CR-156194-Defect-69175> |
| Tofael Khan | 12/14/12016 | 3.23 | 290789a CR 154491 add port\_level\_cos\_ind to PHY\_PORT (PE) – changes tagged with <290789a.154491> |
| Akarsh V | 12/22/2016 | 3.24 | Defect 96256 – Updated the Access Circuit Speed for Circuit Service Code ‘DN’ from 45000 to 44736. |
| Yun Wan | 01/10/2017 | 3.25 | Ticket 000000224475375 updates: move CR60’s mcn triplet to billing Organization instead of access triplet. |
| Akarsh V | 01/18/2017 | 3.26 | 292589 CR150907 – Updated Network Connection and Access Circuit to support the new field “dhv\_flag”. |
| Yun Wan | 01/20/2017 | 3.27 | Removed tunnel vlan data from usrp. |
| Yun Wan | 02/01/2017 | 3.28 | QC129121-US856984 updates |
| Yun Wan | 02/07/2017 | 3.29 | Updates for ticket 223595795 |
| Tofael Khan | 02/08/2017 | 3.30 | <290789a-CR154491-Upd-2017-02-08> Load ‘wanInterfaceName’ for Customer Port (PHY\_PORT)  <284465c-CR158090> Load Contact data for OPT-E-WAN |
| Tofael Khan | 2/15/2017 | 3.31 | Defect 90537 – Access Speed for MLPPP needs to come from ICORE and not CIRCUIT\_SERVICE\_CODE table – change tagged with <Defect-90537> |
| Yun Wan | 3/16/2017 | 3.32 | 293772 CR160298 updates for US873985 |
| Tofael Khan | 4/2/2017 | 3.33 | 296528.161673 – add TDM indicator to PE port for Access Circuit – changes tagged with <296528.161673> |
| Akarsh V/Mahesh MP | 4/4/2017 | 3.34 | 281578b-US870920: Updated SITE\_EXT to support new elements. |
| Tej Sarju | 4/26/2017 | 3.35 | 289037c (US868358, US868359, US868366, US868362) – Updated to load Site Name, Site Status, Status Date, Site Type, WTN, Responsible Center to the GDB SITE\_EXT table. Needed to populate InquireEnterpriseLocations API.  -Updated to load multiple Access Circuit details to a new GDB ASSET\_ACCESS\_CIRCUIT\_DETAILS table. Needed to populate the InquireEnterpriseCustomerAssetDetail (IECAD) API.  Updated to load multiple Access Circuit customer port details to a new GDB CUSTOMER\_PORT\_DETAILS table. Needed to populate the InquireEnterpriseCustomerAssetDetail (IECAD) API.  289037c requires that additional SITE, EQUIPEMENT and CIRCUIT data be provided for the following AT&T services: AVPN, MIS, GMIS, EVPN, ASE, BVOIP/IPFLEX/VDNA, ANIRA [PMO], AVTS, MRS, MLAN, IP Telephony, MSS, MPLS PNT, EPLS, ATS, ISDN PRI, PL, 3rd Party Transport, UVN, POTS. |
| Yun Wan | 5/2/2017 | 3.36 | Defect 191509 updates: change access speed data mapping to use LPP data as first source. |
| Tej Sarju | 5/3/2017 | 3.37 | 289037c (US868358, US868359, US868366, US868362)  The logic given to obtain the SITE\_EXT.WTN for 289037c conflicts with the existing SITE\_EXT.WTN logic for 286278. The reqirement has been updated to first use the 289037c logic and then use the 286278 logic if the 289037c logic returns a NULL value.  The statement on “Don’t overwrite any values with a blank/null” for SITE\_EXT updates has been removed from the 289037c requirements, since GDB should load the current values from the source system. The statements on loading records (insert/update) have been updated to be more clear.  In GDB ASSET\_ACCESS\_CIRCUIT\_DETAILS, the SITE\_ID column uses CIRCUIT.A\_SITE\_ID as the source. Since this has duplicate data, CUST\_ACCESS.SITE\_ID will be used instead. |
| Tej Sarju | 5/8/2017 | 3.38 | 289037c (US868358, US868359, US868366, US868362)  The reference to SITE.SITE\_GLOBAL\_DLCI should be changed to SITE.SITE\_GLBL\_DLCI.  For the GDB CUSTOMER\_PORT\_DETAILS table, there can be multiple PVC’s associated with a single Access Circuit. PVC data will be loaded into a new GDB PVC\_DETAILS table. |
| Tej Sarju | 5/10/2017 | 3.39 | 289037c (US868358, US868359, US868366, US868362)  Added clarificiation that the new GDB ASSET\_ACCESS\_CIRCUIT\_DETAILS & CUSTOMER\_PORT\_DETAILS tables are only to be loaded for CUST\_ACCESS.site\_id key, CUST\_ACCESS.acc\_ckt <> NULL. Added ASSET\_ACCESS\_CIRCUIT\_DETAILS, CUSTOMER\_PORT\_DETAILS and PVC\_DETAILS to list of tables considered child tables of GDB ASSET in ‘deletions’ section (HLD-254035-GCP-FLOW-ICORE-CLEANUP [Icore Data Flow for Deletions]) . |
| Tofael Khan | 5/24/2017 | 3.40 | Backed out 296528.161673 changes – remove TDM indicator |
| Tej Sarju | 5/24/2017 | 3.41 | 289037c (US868358, US868359, US868366, US868362)  Added CONNECTED\_DEVICE field to GDB ASSET\_ACCESS\_CIRCUIT\_DETAILS table |
| Yun Wan | 5/25/2017 | 3.42 | Update access speed data mapping to icore.site.access\_speed for 231405909 (input from sv832h ) |
| Tej Sarju | 6/2/2017 | 3.43 | 289037c (US868358, US868359, US868366, US868362)  In GDB ASSET\_ACCESS\_CIRCUIT\_DETAILS & CUSTOMER\_PORT\_DETAILS table, the sql to traverse from ICORE.CUST\_ACCESS to ICORE.CIRCUIT uses ‘ckt\_id’ as the link. However, ‘ckt\_id’ is not unique across ICORE tables and can result in a match to a circuit belonging to another customer. The sql has been changed to use ‘site\_id’ to traverse from ICORE.CUST\_ACCESS to ICORE.CIRCUIT. In addition, the sql to retrieve SITE.site\_glbl\_dlci uses ‘cust\_id’ as the link, which can result in multiple matches. The sql has been changed to use ‘site\_id’ to traverse from ICORE.CUST\_ACCESS to ICORE.SITE. |
| Yun Wan | 06/22/2017 | 3.44 | Defect 229121 updated biller code’s usrp query. |
| Tej Sarju | 6/28/2017 | 3.45 | 289037c (US868358, US868359, US868366, US868362)  In GDB CUSTOMER\_PORT\_DETAILS & PVC\_DETAILS tables, added a SOURCE\_SYSTEM column to make traversal between these tables and the ASSET\_ACCESS\_CIRCUIT\_DETAILS table more direct during the ETL load process.  In CUSTOMER\_PORT\_DETAILS table, updated sql to eliminate unnecessary traversals through ICORE.CIRCUIT table. Updated CUSTOMER\_PORT\_DETAILS platform key to be a composite of Asset Id + Port Id, due to multiple port association with a single access circuit.  In PVC\_DETAILS table, added new fields to record L/R site\_id and L/R device\_name. |
| Tej Sarju | 7/3/2017 | 3.46 | 289037c (US868358, US868359, US868366, US868362)  In GDB CUSTOMER\_PORT\_DETAILS, updated sql for DATE\_INSTALLED to join on EQUIPMENT.equip\_id = PORT\_ASGMT.equip\_id & PORT\_ASGMT.site\_id = CUST\_ACCESS.site\_id. |
| Tej Sarju | 7/19/2017 | 3.47 | 289037c (US868358, US868359, US868366, US868362)  For GDB SITE\_EXT.wtn, the ICORE source, SITE\_CONTACT.cont\_phone, can return many results for a given SITE.id. A check has been added to only use the result where the primary contact indicator is set. SITE\_CONTACT.cont\_phone can sometimes contain more that 10 digits or alphanumberic characters. For these cases, the corresponding SITE\_EXT.wtn should be set to NULL. (Note: SITE\_EXT.wtn has a data length limit of 10 bytes, as well as the corresponding element in the API Response).  SITE\_EXT has a typo – ‘SITE\_STATUS’ should be ‘STATUS’. |
| Tej Sarju | 7/20/2017 | 3.48 | 289037c (US868358, US868359, US868366, US868362)  SITE\_CONTACT.cont\_phone can sometimes contain more that 10 digits or alphanumberic characters. Valid numbers greater than 10 digits should be captured. GDB SITE\_EXT.wtn will be increased from 10 to 20 chars. The associated API element will also be increased. |
| Tej Sarju | 8/03/2017 | 3.49 | 289037c (US868358, US868359, US868366, US868362)  As requested by ETL team, added ID\_CHANGE\_TRACKING column to all 289037c tables. |
| Akarsh V/Mahesh MP | 10/06/2017 | 3.50 | 295359 CR169138: Added logic to load Flexware and AVPN Data into new tables as part of performance improvement. Tagged with <295359-US325561> |
| Akarsh V/Byomakesh Palai | 11/22/2017 | 3.51 | Added a note to maintain Parent Child relationship.<295359> |
| Shruti CM/ Akarsh V | 11/24/2017 | 3.52 | 295359 CR169138(US325561) : Updated Access\_Ckt\_platform\_Object\_Key and  NW\_Conn\_platform\_Object\_Key as ACCESS\_CIRCUIT\_POK and  NETWORK\_CONNECTION\_POK respectively in tables AVPN\_NETWORK\_CONNECTION\_DATA, IPV4\_DATA, IPV6\_DATA, RELATED\_ASSET\_DATA, AVPN\_ACCESS\_CIRCUIT\_DATA and EQUIPMENT\_DATA  Added additional data base traversal for access method, intercon\_tech, icore.service\_option in the AVPN\_ACCESS\_CIRCUIT\_DATA Table |
| Akarsh V | 12/12/2017 | 3.53 | 295359 CR169138(US325561) : Updated the note under IPV4\_DATA and IPV6\_DATA to support ‘PE’ port info based on the clients confirmation. |
| Akarsh V | 12/20/2017 | 3.54 | 295359 CR169138: Added new tables called IPV4\_ADDRESS\_DATA, IPV6\_ADDRESS\_DATA and NETWORK\_IP\_DATA to accommodate multiple occurrances. |
| Akarsh V | 12/22/2017 | 3.55 | 295359 CR169138: Updated the condition for ITU\_CARRIER\_CD, prov\_port\_nw\_vlan\_stacking\_flag, COS\_package in AVPN\_ACCESS\_CIRCUIT\_DATA table.  295359 CR169138: Changed the NOT NULL constraint to NULL constraint for the following:  AVPN\_ACCESS\_CIRCUIT\_DATA:  Prov\_port\_ingress\_profile\_Id,Prov\_port\_ingress\_profile\_Name,Prov\_port\_egress\_profile\_Id,Prov\_port\_egress\_profile\_Name  AVPN\_NETWORK\_CONNECTION\_DATA:  Prov\_Edge\_Ingress\_Profile\_Id,Prov\_Edge\_Ingress\_Profile\_Name,Prov\_Edge\_Egress\_Profile\_Id,Prov\_Edge\_Egress\_Profile\_Name. |
| Akarsh V | 01/03/2018 | 3.56 | 295359 CR169138: Added new columns – ID, TYPE – in IPV4\_ADDRESS\_DATA and IPV6\_ADDRESS\_DATA tables as part of <Defect 388187>. |
| Akarsh V | 02/09/2018 | 3.57 | 295359 CR169138: Added the queries for COS related data for Network Connection and Access Circuit. |
| Yun Wan | 5/23/2018 | 3.58 | 000000245369061 fix to remove the duplicates from the views : TUNNEL\_VLAN\_PE\_PORT\_VW and TUNNEL\_VLAN\_VW |
| Akarsh V | 06/22/2018 | 3.59 | Added logic to redesign the managed equipment being loaded into GDB. Tagged the changes with <Redesign Managed Equip> |
| Yun Wan | 06/22/2018 | 3.60 | Added a MVL\_IND in <304329-US10770> GDB.ASSET\_EXT\_NETWORK\_CONNECTION |
| Akarsh V | 07/18/2018 | 3.61 | <PVT Defect 555310> Added the note to restrict NST\_TUNNEL, NST\_INTERNET type network\_connection assets in NETWORK\_CONNECTION\_DATA table. |
| Akarsh V | 10/17/2018 | 3.62 | <302503> Changes done as part of US454595, US454596 |
| Ravali Yemineni | 11/16/2018 | 3.63 | <302503> for the organization Identifier of subsubaccount , created customer\_ubsubaccount\_vw.  Embedded the customer\_ubsubaccount\_vw file.  Added ‘OR’ condition in ‘object\_key\_to’ to be referred as customer\_ubsubaccount\_vw.id in asset to org and site to org associations. |
| Akarsh V | 11/22/2018 | 3.64 | <302503> Updated the view customer\_ubsubaccount\_vw to check for ubsubaccountnumber is not null. |
| Akarsh V | 11/29/2018 | 3.65 | <Defect 663866> Striked out the requirement to support cust\_id column in the spreadsheet present under metadata section. This is not a defect but just for tracking purpose. |
| Akarsh V | 12/06/2018 | 3.66 | Updated the column name ‘ubsubaccountnumber’ to ‘ub\_sub\_acct\_num’ and ‘globallocationid’ to ‘global\_loc\_id’. Changes are tagged with <Defect 608684> These changes are being done based on the AID updates from ICORE. |
| Ravali/Akarsh | 01/17/2019 | 3.77 | Added Note in associations of Asset(Network Connection, Access\_Circuit, Customer\_Network) to Service as part of **302503 - tkt 690443** |

Table of Contents

[Overview 11](#_Toc406425140)

[A Note on how to use this HLD 11](#_Toc406425141)

[Problem Statement 13](#_Toc406425142)

[Design Decisions 13](#_Toc406425143)

[Requirements 14](#_Toc406425144)

[A Note on “Asset Unification” 14](#_Toc406425145)

[Database 15](#_Toc406425146)

[Service Delivery Data Flow into Golden Database [Icore Data Flow] 29](#_Toc406425147)

[Data Index Maintenance for PORT Data [254035b] 247](#_Toc406425148)

[Service Delivery Data Flow into Golden Database [Icore Data Flow] for Asset-to-Service-Option Associations (254035c) 248](#_Toc406425149)

[Service Delivery Data Flow into Golden Database [Icore Data Flow] for Static Initial Load for 259118 282](#_Toc406425150)

[Service Delivery Data Flow into Golden Database [Icore Data Flow] for Static Initial Load for 258491 284](#_Toc406425151)

[Appendix: Circuit Id Formats 286](#_Toc406425152)

[Appendix: Copies of references to other HLDs 287](#_Toc406425153)

[Alternative Designs 288](#_Toc406425154)

[Assumptions/Risks 288](#_Toc406425155)

[Traceability Matrix 289](#_Toc406425156)

[Pre-Production Disaster Recovery Planning 289](#_Toc406425157)

[Other Plans and References 290](#_Toc406425158)

[Acceptance & Approvals 290](#_Toc406425159)

## Overview

The Golden Database (GDB) on GCP will be extended and linked to ICORE. LPP-CPE (NC3), GPS, Instar, Service Manager and PLATINUM data flows into the GDB will be modified.

This will be a DBOR (DataBase Of Record) for data objects like Assets, Services, Sites, Organizations and Contacts and their associations among each other.

The database will contain read-only representations of data object anchor points using the GCP Enterprise Key Translation. The actual data objects are maintained in other DBORs. GCP will maintain the representing anchor points across the maintaining DBORs.

The database will be mission critical for C-BUS in respect to Contact objects and associations and for eStrategy applications in respect to all objects and associations.

This HLD is about everything that is needed in order to be able to load ICORE data for the objects given above into GDB.

## A Note on how to use this HLD

This HLD has initially been created for project 254035. It still contains all sections from that project.  
For projects 254035c/259118/258491 all needed additions that apply to corresponding sections that already exist in this HLD before 254035c/259118/258491 will be included in this HLD.

Using this approach keeps all design logic that belongs together at one place.

The sections/pieces added for 254035c/259118/258491 can easily be identified (see the “history” section under “2014-07-07”/”2014-08-15”/”2014-09-09”).

A high-level overview of this HLD’s contents is given below:

1. **Database items (DBA)**
   1. GDB: “service” data related additions
   2. GDB: change tracking related additions
   3. GDB: changes of already existing “asset” tables;  
       addition of new “asset” and “port” tables
   4. ICORE: addition of 3 views for abstraction of ICORE.PVC
   5. ICORE: addition of a table system for managing MCN/GRC/SOC data  
       pulled from sources other than ICORE
2. **Data Loading items (ETL)**
   1. Load MCN/GRC/SOC data for ICORE inventory from several different data sources
   2. Load customer/organization data from ICORE into GDB  
      (by ICORE “cust id” and by “MCN/GRC/SOC” via mapping)
   3. Load location/address data from ICORE into GDB
   4. Load asset (access circuit) data from ICORE into GDB
   5. Load asset (customer network (VPN)) [3 types] data from ICORE into GDB
   6. Load asset (network connection (PVC)) [3 types] data from ICORE into GDB
   7. Load port (logical and physical) data from ICORE into GDB
   8. Associate NC3 port/equipment data with the data from ICORE
   9. Load service type data from ICORE into GDB
   10. Associate location/address type data with  
        - organization(s)  
        - asset(s) (network connection (PVC); access circuit)
   11. Associate service type data with  
        - asset(s) (network connection (PVC); access circuit)
   12. Associate asset (access circuit) to its “parent” asset (access circuit)
   13. Associate physical ports with access circuits
   14. Associate network connections with access circuits
   15. Associate network connections with customer networks
   16. Associate logical ports with network connections
   17. Associate access circuits with customer/organization for ICORE “cust id” and “MCN/GRC/SOC”
   18. Associate network connections [3 types] with customer/organization for ICORE “cust id” and “MCN/GRC/SOC”
   19. Associate customer networks [3 types] with customer/organization for ICORE “cust id” and “MCN/GRC/SOC”

**NOTE** that a very similar ETL process (compared to what is requested in this HLD) already exists for loading data from INSTAR into GDB. This process is owned by the GCP-GRDB ETL group (led by Dileep Shetty), and it can be taken as an already working “model” for implementing many items from this HLD, especially items related to the Enterprise Key Translation (EKT).

## Problem Statement

Business (Service Assurance/Operations) intends to get a centralized system for service assurance inventory, registration and contact management. This system (the Golden Database (GDB)) will be used to expose (read-only) management ability for all GDB object types and objects to the customer.

The project aims to provide an infrastructure in which any object (asset, service, site, customer, contact … ) is represented once per customer, and can be reused across services/products.

Access to the infrastructure is expected to be reusable for additional systems. Long-term the use of this infrastructure might be extended to support object (asset, service, site, customer, contact … ) management beyond service assurance needs and might be linked to the service delivery processes and systems.

This project changes the data flow for the supported backend DBORs LPP-CPE (NC3), GPS, Instar and Service Manager and Platinum.

This project extends the supported backend DBORs by ICORE.

This project extends the general structure of asset objects by adding several new asset object types (circuits, network connections, customer networks) and by adding asset object data elements to be loaded into GDB so they can quickly be provided to the client without pulling them from different DBORs.

This HLD is about everything that is needed in order to be able to load ICORE data for the objects given above into GDB.

## Design Decisions

The processing rules (algorithms, SQL, etc.) in the design section are for design purposes only.

Reference content (e.g. status values, type values etc.) has to be checked for the actual database defined content values (e.g. spelling, case, etc.) during the detailed design.

Development has to use the best possible implementation for the derived SQL, database implementations and processing to ensure the implementation will be scalable, thread-safe, quickly extensible and resource efficiently executable.

Flow diagrams describe the implementation logically and do not imply a specific implementation unless explicitly stated as required. The sequence or number of steps will be optimized during development.

SQL processing shown in trigger bodies, procedures, functions, etc. is meant to provide solution suggestions and explicitly is not production ready code.

Contained schema names are describing schema names, not necessarily actual database instantiation schema names. For flip-flop enabled schemas, only the describing schema name of the flip-flop group is mentioned, while the instantiated ‘online’ and ‘offline’ synonym/view façade and backend content schema names are omitted. e.g. SCHEMA may act as a synonym for flip-flop schemas SCHEMA, SCHEMA1, SCHEMA2, SCHEMA\_LD. Development will apply data model and data content changes to the instantiated schema name(s) and will take flip-flop usage models into account.

## Requirements

<BEGIN 254035c.UTC-89113.003>

### A Note on “Asset Unification”

At the moment (2014-10-28) two different “asset unification” processes exist:

1. **Equipment Unification**  
   (a version limited to ATS sub-equipment that gets unified into “router” equipment)
2. **Access Circuit Unification**

The 2 unification processes are currently described in the HLD:

* **GCP-SA-HLD-for-GCP-GDB.xxxxxx.For-UIS-et-al-Load.docx**

Any time a reference “asset unifications” is given, then the corresponding “unification” sections from the above HLD are meant.

<END 254035c.UTC-89113.003>

### Database

#### HLD-254035-GCP-GDB-for-ICORE-001 [SERVICE\_TYPE Content and Mapping]

This requirement is an addition to the requirement “HLD-232213e-GCP-GDB-104 [SERVICE\_TYPE Content and Mapping]” of an older HLD.

(NOTE: Keep in mind that some other HLDs for this project also add data to the tables listed below.)

Ensure that the following mapping exists in the SERVICE\_TYPE, SERVICE\_TYPE\_NOTATION and DBOR\_INDICATOR tables in the GDB schema:



*Here already existing entries (that must not be changed) are given in grey colour.*

*Entries to be added are given in dark black colour.*

Ensure that the following mapping exists in the SERVICE\_TYPE and SERVICE\_TYPE\_BLOCKED\_SYSTEM tables in the GDB schema:



*Here no changes/additions have to be applied compared to the original HLD.*

*For ID\_CHANGE\_TRACKING, reference a single record for the deployment, i.e. during deployment insert one record into CHANGE\_TRACKING/CHANGE\_SYSTEM/CHANGE\_USER using the deployment database instance time, change\_system.name ‘INITIAL\_LOAD’, change\_user.name ‘INITIAL\_LOAD’.*

#### HLD-254035b-GCP-GDB-001 [SERVICE\_TYPE Content and Mapping]

This is a “placeholder” requirement. The real requirement is contained in the HLD “**GCP-SA-HLD-for-GCP-GDB.254035b.docx**” (outdated !) and “**GCP-SA-HLD-for-GCP-GDB.254035c.For-UIS-et-al-Load.docx**”.

#### HLD-254035-GCP-GDB-for-ICORE-002 [CHANGE\_SYSTEM & CHANGE\_USER Content]

Make sure the following data is added to the GCP GDB Data:

|  |  |
| --- | --- |
| **Data element** | **Value** |
| GDB.CHANGE\_SYSTEM.name | ICORE\_TO\_GDB |
| GDB.CHANGE\_USER.name | ICORE\_TO\_GDB |

<286278><REL1602>

The change tracking event for the service delivery data flow must depend on the source for 286278 enhancement:

if the record is created/updated from ICORE, use CHANGE\_SYSTEM.name = ’ICORE\_TO\_GDB’ and CHANGE\_USER.name = ’ICORE\_TO\_GDB’,

if the record is created/updated from USRP, use CHANGE\_SYSTEM.name = ’USRP\_TO\_GDB’ and CHANGE\_USER.name = ’USRP\_TO\_GDB’, and

if the record is created/updated from GRID, use CHANGE\_SYSTEM.name = ’GRID\_TO\_GDB’ and CHANGE\_USER.name = ’GRID\_TO\_GDB’.

It is sufficient to have one change tracking event for all affected records during a single execution of this process.

</286278>

#### HLD-254035b-GCP-GDB-002 [CHANGE\_SYSTEM & CHANGE\_USER Content]

This is a “placeholder” requirement. The real requirement is contained in the HLD “**GCP-SA-HLD-for-GCP-GDB.254035b.docx**” (outdated !) and “**GCP-SA-HLD-for-GCP-GDB.254035c.For-UIS-et-al-Load.docx**”.

#### HLD-254035-GCP-GDB-003 [Primary Key Creation]

Ensure for all processing that the primary key creation is based on the following logic:



Note that what is given in grey text colour should either already exist or be given in another HLD !

#### HLD-254035-GCP-GDB-004 [General GDB additions and changes]

This requirement covers general additions (tables and columns) and changes (tables and columns) to the GDB database.

This is what is needed:

* add a table “GDB.ASSET\_EXT\_CUSTOMER\_NETWORK”;  
  details contained in the embedded file:  
  
* add a table “GDB.ASSET\_EXT\_NETWORK\_CONNECTION”;  
  details contained in the embedded file:  
  
* add a table “GDB.ASSET\_EXT\_ACCESS\_CIRCUIT”;  
  details contained in the embedded file:  
  
* add a table “GDB.ASSET\_EXT\_EQUIMENT”;  
  details contained in the embedded file:  
  
* add a table “GDB.ASSET\_EXT\_ASSET\_GROUP”;  
  details contained in the embedded file:  
  
* add a table “GDB.NETWORK\_TYPE”;  
  details contained in the embedded file:  
  
* add a table “GDB.PORT\_TYPE”;  
  details contained in the embedded file:  
  
* add a table “GDB.PHY\_PORT”;  
  details contained in the embedded file:  
  
* add a table “GDB. PHY\_PORT\_IDENTIFIER”;  
  details contained in the embedded file:  
  
* add a table “GDB. PHY\_PORT\_IDENTIFIER\_VALUE”;  
  details contained in the embedded file:  
  
* add a table “GDB.LOG\_PORT”;  
  details contained in the embedded file:  
  
* add a table “GDB. LOG\_PORT\_IDENTIFIER”;  
  details contained in the embedded file:  
  
* add a table “GDB. LOG\_PORT\_IDENTIFIER\_VALUE”;  
  details contained in the embedded file:  
  
* add a table “GDB.CIRCUIT\_SERVICE\_CODES”;  
  details contained in the embedded file:  
  
* load the table “GDB.CIRCUIT\_SERVICE\_CODES” with the data contained in the embedded EXCEL file (only import the first 4 columns of that EXCEL file !):

<QC96256> Updated the speed for Service Code ‘DN’.



* change the table “GDB.SITE”;  
  details contained in the embedded file:  
  

**<289037c>**

Ensure the support of additional fields (status, statusDate, responsibleCenter, serviceLevel) in the existing GDB SITE\_EXT table as shown in the Database Schema section of the GDB HLD Aggregated API’s” document.

Ensure the support of the new GDB ASSET\_ACCESS\_CIRCUIT\_DETAILS, CUSTOMER\_PORT\_DETAILS and PVC\_DETAILS tables as shown in the Database Schema section of the “GDB HLD Aggregated API’s” document.

For maintenance/update puposes, the schema for these tables will only be shown in the “GDB HLD Aggregated API’s” document in the “HLD-289037c-GCP-100-P [GDB Schema Changes]” section. The new tables and new 289037c requirements have a “<289037c>” tag.

**<End 289037c>**

**NOTE** that the embedded SQL files have not been tested and may not be used “as is” for deployment. However, they give the details of what is needed.

Make sure all these new tables will also be available in “GDB\_HIST” and enable history management for them.

#### HLD-254035b-GCP-GDB-010 [General GDB additions and changes]

This is a “placeholder” requirement. The real requirement is contained in the HLD “**GCP-SA-HLD-for-GCP-GDB.254035b.docx**” (outdated !) and “**GCP-SA-HLD-for-GCP-GDB.254035c.For-UIS-et-al-Load.docx**”.

#### HLD-254035b-GCP-GDB-020 [EQUIPMENT\_TYPE Set of Tables]

This is a “placeholder” requirement. The real requirement is contained in the HLD “**GCP-SA-HLD-for-GCP-GDB.254035b.docx**” (outdated !) and “**GCP-SA-HLD-for-GCP-GDB.254035c.For-UIS-et-al-Load.docx**”.

#### HLD-254035b-GCP-GDB-021 [EQUIPMENT\_TYPE Content and Mapping]

This is a “placeholder” requirement. The real requirement is contained in the HLD “**GCP-SA-HLD-for-GCP-GDB.254035b.docx**” (outdated !) and “**GCP-SA-HLD-for-GCP-GDB.254035c.For-UIS-et-al-Load.docx**”.

#### HLD-254035-GCP-ICORE-010 [EKT Helper View for ICORE.PVC records]

This requirement covers the fact that in the context of data from the ICORE.pvc table data exists for both the “local” and the “remote” side (e.g. PVC.pvc\_lsite\_id and PVC.pvc\_rsite\_id). For the Enterprise Key Translation (EKT) this is “abstracted away” by making use of the “NETWORK\_CONNECTION”, “NETWORK\_CONNECTION\_INET\_VLAN” (for Internet VLAN network connections) and “CUSTOMER\_NETWORK\_INET\_VLAN” (for Internet VLAN customer networks) view.

The “NETWORK\_CONNECTION” view needs to be created in the GCP ICORE Replica database using the “ICORE” schema for execution. The DDL for that creation is contained in the following embedded file.



The “NETWORK\_CONNECTION\_INET\_VLAN” view needs to be created in the GCP ICORE Replica database using the “ICORE” schema for execution. The DDL for that creation is contained in the following embedded file.



The “CUSTOMER\_NETWORK\_INET\_VLAN” view needs to be created in the GCP ICORE Replica database using the “ICORE” schema for execution. The DDL for that creation is contained in the following embedded file.



#### HLD-302503-GCP-GDB-from-ICORE-DBA-029 [ customer\_ubsubaccount\_vw ]

PID : 302503

The “CUSTOMER\_UBSUBACCOUNT” view needs to be created in the GCP ICORE Replica database using the “ICORE” schema for execution. The DDL for that creation is contained in the following embedded file.

<Defect 608684>Updated the column name from ubsubaccountnumber to ub\_sub\_acct\_num



#### HLD-254035-GCP-ICORE-011 [New MCN/GRC/SOC Mapping Tables]

This requirement covers general additions (tables and sequences) to the ICORE database in order to support MCN/GRC/SOC management for customer/organization identification.

This is what is needed:

* add a sequence “ICORE.MCN\_GRC\_SOC\_SOURCE\_ID”;  
  details contained in the embedded file:  
    
  for this sequence execution permissions need to be given to the application user id (to be determined in detail) that maintains the MCN/GRC/SOC mapping tables !
* add a table “ICORE.MCN\_GRC\_SOC\_SOURCE\_CODES”;  
  details contained in the embedded file:  
  
* load the table “ICORE.MCN\_GRC\_SOC\_SOURCE\_CODES” with the data contained in the embedded EXCEL file:  
  <286282-US704600> added new rows</286282-US704600> <Ticket-000000224475375>  
  
* add a table “ICORE.MCN\_GRC\_SOC\_SOURCE”;  
  details contained in the embedded file:  
  Defect 22736: added triplet\_type into the table  
    
    
  for this table insert/update/deletion permissions need to be given to the application user id (to be determined in detail) that maintains the MCN/GRC/SOC mapping tables !
* add a table “ICORE.MCN\_GRC\_SOC\_SITE\_IDS”;  
  details contained in the embedded file:  
    
  Defect 22736: added prem\_country into the table.  
    
  for this table insert/update/deletion permissions need to be given to the application user id (to be determined in detail) that maintains the MCN/GRC/SOC mapping tables !
* add a table “ICORE.MCN\_GRC\_SOC\_CIRCUIT\_IDS”;

<286282-US704600> added station\_name column <Ticket-000000224475375> (removed)  
details contained in the embedded file:  
  
for this table insert/update/deletion permissions need to be given to the application user id (to be determined in detail) that maintains the MCN/GRC/SOC mapping tables !

**NOTE** that the embedded SQL files have not been tested and may not be used “as is” for deployment. However, they give the details of what is needed.

#### HLD-254035-GCP-ICORE-012 [EKT Helper View for ICORE “PE” Logical Ports]

This requirement covers the need for a dedicated EKT helper view that is needed for ICORE “PE” logical port objects after the ICORE Logical Port loading corrections after 2014-04.

For ICORE “PE” logical port objects a new dedicated EKT helper view is needed; the name of that view shall be “LOG\_PORT\_PE\_VW”.

The “LOG\_PORT\_PE\_VW” view needs to be created in the GCP ICORE Replica database using the “ICORE” schema for execution. The DDL for that creation is contained in the following embedded file.



#### HLD-254035-GCP-ENTKEY-001 [METADATA Content]

The following data must be ensured to be added to resp. contained in the GCP Meta Data tables (ref. project 188413a). The attached table contains the denormalized information for the Meta Database content (for the above EKT helper view “LOG\_PORT\_PE\_VW” only !):

<284465c-CR158090> Updated the attachment for additional metadata entries



<LOG-PORT-EKT-FIX> (BEGIN)

#### HLD-LOG-PORT-EKT-FIX-GCP-ICORE-013 [EKT Helper View for ICORE “CE” Logical Ports]

This requirement covers the need for 2 dedicated EKT helper views that are needed for ICORE “CE” logical port objects.  
*Originally both “network connection” assets and “log port” ports had made use of the same “NETWORK\_CONNECTION” resp. “NETWORK\_CONNECTION\_INET\_VLAN” view which, however, lead to data inconsistencies when data needed to be deleted due to different constraints for data to be included when loading.*

For ICORE “CE” logical port objects a new dedicated EKT helper view is needed; the name of that view shall be “LOG\_PORT\_CE\_VW”.

The “LOG\_PORT\_CE\_VW” view needs to be created in the GCP ICORE Replica database using the “ICORE” schema for execution. The DDL for that creation is contained in the following embedded file.



For ICORE “CE” logical port objects another new dedicated EKT helper view is needed; the name of that view shall be “LOG\_PORT\_CE\_INET\_VLAN\_VW”.

The “LOG\_PORT\_CE\_INET\_VLAN\_VW” view needs to be created in the GCP ICORE Replica database using the “ICORE” schema for execution. The DDL for that creation is contained in the following embedded file.



#### HLD-LOG-PORT-EKT-FIX-GCP-ENTKEY-002 [METADATA Content]

The following data must be ensured to be added to resp. contained in the GCP Meta Data tables (ref. project 188413a). The attached table contains the denormalized information for the Meta Database content (for the above EKT helper views “LOG\_PORT\_CE\_VW” and “LOG\_PORT\_CE\_INET\_VLAN\_VW” only !):



<LOG-PORT-EKT-FIX> (END)

#### HLD-254035c-GCP-GDB-010 [General GDB additions and changes]

<BEGIN 254035c.CR-Generic>

This requirement covers general additions (tables and columns) and changes (tables and columns) to the GDB database.

This is what is needed:

* add a table “GDB.SERV\_OPT”;  
  details contained in the embedded file:  
  
* add a table “GDB.SERV\_OPT\_TYPE”;  
  details contained in the embedded file:  
  
* add a table “GDB.SERV\_OPT\_TYPE\_NOTATION”;  
  details contained in the embedded file:  
  
* change the table “GDB.ASSET\_EXT\_TRUNK\_CHNL\_CIRCUIT”;  
  details contained in the embedded file:  
  
* change the table “GDB.ASSET\_EXT\_PL\_CIRCUIT\_CKL”;  
  details contained in the embedded file:  
  

**NOTE** that the embedded SQL files have not been tested and may not be used “as is” for deployment. However, they give the details of what is needed.

Make sure all these new tables will also be available in “GDB\_HIST” and enable history management for them.

<END 254035c.CR-Generic>

#### HLD-254035c-GCP-GDB-011 [OBJECT\_TYPE Content]

<BEGIN 254035c.CR-Generic>

Ensure that the following content is added to the OBJECT\_TYPE table in the GDB and METADATA schema:

|  |
| --- |
| **TYPE** |
| SERVICE\_OPTION |

Make sure the “id” value will be the same for both schemas !!!

<END 254035c.CR-Generic>

#### HLD-254035c-GCP-GDB-012 [ROLE Content and Mapping]

<BEGIN 254035c.CR-Generic>

Ensure that the following mapping is added to FUNCTION\_TYPE, FUNCTION\_ROLE, ROLE, ROLE\_NOTATION, ASSOCIATION\_TYPE in the GDB schema:



*For ID\_CHANGE\_TRACKING, reference a single record for the deployment, i.e. during deployment insert one record into CHANGE\_TRACKING/CHANGE\_SYSTEM/CHANGE\_USER using the deployment database instance time, change\_system.name ‘INITIAL\_LOAD’, change\_user.name ‘INITIAL\_LOAD’.*

<END 254035c.CR-Generic>

#### HLD-254035c-GCP-GDB-013 [SERV\_OPT\_TYPE Content and Mapping]

<BEGIN 254035c.CR-Generic>

Ensure that the following mapping exists in the SERV\_OPT, SERV\_OPT\_TYPE and SERV\_OPT\_TYPE\_NOTATION GDB schema:



In order to create EKT values for “GDB.SERV\_OPT.id” the mechanism that is used for “GDB.SERVICE.id” needs to be “copied” and then be applied to creating EKT values.

*For ID\_CHANGE\_TRACKING, reference a single record for the deployment, i.e. during deployment insert one record into CHANGE\_TRACKING/CHANGE\_SYSTEM/CHANGE\_USER using the deployment database instance time, change\_system.name ‘INITIAL\_LOAD’, change\_user.name ‘INITIAL\_LOAD’.*

<END 254035c.CR-Generic>

#### HLD-254035c-GCP-GDB-014 [SERVICE\_TYPE Content and Mapping: Add “Applications Management”]

<BEGIN 254035c.CR-Generic>

Ensure that the following mapping is added to the SERVICE, SERVICE\_TYPE and SERVICE\_TYPE\_NOTATION GDB schema:



The details are:

* Add a row to GDB.service\_type
* Update GDB.service\_type\_notation so that “id\_service\_type” points to the new “GDB.service\_type” record
* Update GDB.service so that “id\_service\_type” points to the new “GDB.service\_type” record

*For ID\_CHANGE\_TRACKING, reference a single record for the deployment, i.e. during deployment insert one record into CHANGE\_TRACKING/CHANGE\_SYSTEM/CHANGE\_USER using the deployment database instance time, change\_system.name ‘INITIAL\_LOAD’, change\_user.name ‘INITIAL\_LOAD’.*

<END 254035c.CR-Generic>

#### HLD-254035c-GCP-GDB-015 [IDENTIFIER\_TYPE Content]

<BEGIN 254035c.CR-108228>

Ensure that the following content exists or is added in/to the IDENTIFIER\_TYPE table in the GDB schema:

|  |
| --- |
| **TYPE** |
| MCN\_GRC |

*For ID\_CHANGE\_TRACKING, reference a single record for the deployment, i.e. during deployment insert one record into CHANGE\_TRACKING/CHANGE\_SYSTEM/CHANGE\_USER using the deployment database instance time, change\_system.name ‘INITIAL\_LOAD’, change\_user.name ‘INITIAL\_LOAD’.*

<END 254035c.CR-108228>

#### HLD-259118-GCP-GDB-020 [General GDB additions and changes]

<BEGIN 259118>

**STEP 1**

This requirement covers general additions (tables and columns) and changes (tables and columns) to the GDB database.

This is what is needed:

* change the table “GDB.ASSET\_EXT\_CUSTOMER\_NETWORK”;  
  details contained in the embedded file:  
  

**NOTE** that the embedded SQL files have not been tested and may not be used “as is” for deployment. However, they give the details of what is needed.

Make sure all these new tables will also be available in “GDB\_HIST” and enable history management for them.

**STEP 2**

After the table changes have been applied, and the ETL jobs to execute the initial load for the new column “id\_network\_sub\_type” have been completed, then the new column “id\_network\_sub\_type” shall be changed to become “NOT NULL”.

<END 259118>

#### HLD-259118-GCP-GDB-021 [IDENTIFIER\_TYPE Content]

<BEGIN 259118>

Ensure that the following content exists or is added in/to the IDENTIFIER\_TYPE table in the GDB schema:

|  |
| --- |
| **TYPE** |
| INSTAR\_NWKCON\_SDID\_IDENTIFIER |

*For ID\_CHANGE\_TRACKING, reference a single record for the deployment, i.e. during deployment insert one record into CHANGE\_TRACKING/CHANGE\_SYSTEM/CHANGE\_USER using the deployment database instance time, change\_system.name ‘INITIAL\_LOAD’, change\_user.name ‘INITIAL\_LOAD’.*

<END 259118>

#### HLD-286278-GCP-GDB-from-USRP-DBA-022 [ Identifier\_Type ]

The following data must be ensured to be added into GDB.

|  |  |
| --- | --- |
| **IDENTIFIER\_TYPE** | |
| ID | *HLD-270843-GCP-GDB-from-GRID-DBA-010 [ Primary Key ]* |
| NAME | “E2E\_VPN\_KEY\_ASSET\_IDENTIFIER” |
| ID\_CHANGE\_TRACKING | *HLD-270843-GCP-GDB-from-GRID-020 [ Change Tracking ]* |

#### HLD-286278-GCP-GDB-from-ICORE-DBA-023 [ ASSET\_EXT\_CUSTOMER\_NETWORK ]

The following new columns must be ensured to be added into GDB.ASSET\_EXT\_CUSTOMER\_NETWORK.

|  |  |  |  |
| --- | --- | --- | --- |
| **GDB.ASSET\_EXT\_CUSTOMER\_NETWORK** | | | |
| **Name** | **Datatype** | **Constraints** | **Comments** |
| ETE\_SITELESS\_KEY | String, A/N 20 | Nullable |  |
| ETE\_KEY\_SOURCE | String, A/N 10 | Nullable |  |
| REGION\_FLAG <302503> | String, A/N 15 | Nullable |  |

#### HLD-286278-GCP-GDB-from-ICORE-DBA-024[ ASSET\_EXT\_NETWORK\_CONNECTION ]

The following new columns must be ensured to be added into GDB.ASSET\_NETWORK\_CONNECTION.

|  |  |  |  |
| --- | --- | --- | --- |
| **GDB.ASSET\_EXT\_NETWORK\_CONNECTION** | | | |
| **Name** | **Datatype** | **Constraints** | **Comments** |
| ETE\_KEY\_SOURCE | String, A/N 10 | Nullable |  |
| <279006> |  |  |  |
| ID\_NETWORK\_TYPE | Integer, 5 | Nullable | <REL1602> |
| </279006> |  |  |  |
| DHV\_FLAG | VARCHAR2 (1) | Nullable | <292589-US847778> |
| MVL\_IND | VARCHAR2(1) | Nullable | <304329-US10770>  A full refreshment could be needed to populate the value for this column |
| REGION\_FLAG <302503> | String, A/N 15 | Nullable |  |

#### HLD-286278-GCP-GDB-from-ICORE-DBA-025 [ ASSET\_EXT\_ACCESS\_CIRCUIT ]

The following new columns must be ensured to be added into GDB.ASSET\_ACCESS\_CIRCUIT.

|  |  |  |  |
| --- | --- | --- | --- |
| **GDB.ASSET\_EXT\_ACCESS\_CIRCUIT** | | | |
| **Name** | **Datatype** | **Constraints** | **Comments** |
| ETE\_KEY\_SOURCE | String, A/N 10 | Nullable |  |
| <288655a>  THIRD\_PARTY\_FLAG | CHAR (1) | Nullable |  |
| DHV\_FLAG <292589-US847769> | VARCHAR2 (1) | Nullable |  |
| REGION\_FLAG <302503> | String, A/N 15 | Nullable |  |

#### HLD-286278-GCP-GDB-from-ICORE-DBA-026 [ ASSET\_EXT\_EQUIPMENT ]

The following new columns must be ensured to be added into GDB.ASSET\_EQUIPMENT.

|  |  |  |  |
| --- | --- | --- | --- |
| **GDB.ASSET\_EXT\_EQUIPMENT** | | | |
| **Name** | **Datatype** | **Constraints** | **Comments** |
| ETE\_KEY\_SOURCE | String, A/N 10 | Nullable |  |

#### HLD-286278-GCP-GDB-from-ICORE-DBA-027 [ SITE\_EXT ]

The following new columns must be ensured to be added into GDB.SITE\_EXT.

|  |  |  |  |
| --- | --- | --- | --- |
| **GDB.SITE\_EXT** | | | |
| **Name** | **Datatype** | **Constraints** | **Comments** |
| ETE\_KEY\_SOURCE | String, A/N 10 | Nullable |  |

#### HLD-286278-GCP-GDB-from-ICORE-DBA-028 [ PHY\_PORT ]

The following new columns must be ensured to be added into GDB.PHY\_PORT.

|  |  |  |  |
| --- | --- | --- | --- |
| **GDB.PHY\_PORT** | | | |
| **Name** | **Datatype** | **Constraints** | **Comments** |
| ETE\_KEY\_SOURCE | String, A/N 10 | Nullable |  |
| <290789a-CR154491-Upd-2017-02-08>  WAN\_INTERFACE\_NAME | VARCHAR2(100) | Nullable |  |
| ~~<296528.161673>~~  ~~tdm\_indicator~~ | ~~CHAR(1)~~ | ~~Nullable~~ |  |

#### HLD-293772-CR160298-GCP-GDB-from-USRP-DBA-029 [ Helper Views]

The following view(s) should be created in USRP replication schema:

TUNNEL\_VLAN\_PE\_PORT\_VW:

select b.\*

from usrp.pvc a, usrp.logical\_port b, usrp.port\_info c -- <000000245369061>

where a.pvc\_type = 37

and b.log\_port\_id = a.log\_port\_id

and a.buf\_ind = 'N'

and a.valid\_ind = 'Y'

and b.log\_port\_id = c.log\_port\_id -- <000000245369061>

and c.disc\_ind != 'Y' -- <000000245369061>

and (a.ete\_svc\_conn\_key is null or not exists ( --- check if same ete\_svc\_conn\_key is loaded from IEOD or not.

select 1

from gdb.ete\_key\_map

where internal\_key\_value = a.ete\_svc\_conn\_key --- only if it is populated

and ete\_key\_source = 'IEOD' and rownum<2

))

TUNNEL\_VLAN\_VW:

select distinct a.icore\_pvc\_id, b.icore\_site\_id, a.cir, a.ete\_svc\_conn\_key, b.port\_speed, b.log\_port\_id, b.ete\_site\_key, b.ete\_port\_key, b.ete\_access\_key, b.ethernet\_access\_bandwidth

from usrp.pvc a, usrp.logical\_port b, usrp.port\_info c -- <000000245369061>

where a.pvc\_type = 37 –- tunnel vlan

and b.log\_port\_id = a.log\_port\_id

and a.buf\_ind = 'N'

and a.valid\_ind = 'Y'

and b.valid\_ind = 'Y'

and b.buf\_ind = 'N'

and a.icore\_pvc\_id is not null

and b.service\_name = 1 – avpn

and b.log\_port\_id = c.log\_port\_id -- <000000245369061>

and c.disc\_ind != 'Y' -- <000000245369061>

and (a.ete\_svc\_conn\_key is null or not exists ( --- check if same ete\_svc\_conn\_key is loaded from IEOD or not.

select 1

from gdb.ete\_key\_map

where internal\_key\_value = a.ete\_svc\_conn\_key --- only if it is populated

and ete\_key\_source = 'IEOD' and rownum<2

))

#### HLD-286282-US704600-GCP-GDB-from-ICORE-DBA-029 [ Billing Account Mapping Table]

This requirement covers general additions (tables) to the ICORE database in order to support Billing Account management for customer/organization identification.

|  |  |  |  |
| --- | --- | --- | --- |
| **ICORE.GDB\_BILLING\_ACCOUNT** | | | |
| **Name** | **Datatype** | **Constraints** | **Comments** |
| ACCOUNT\_KEY | String, A/N 13 | Primary Key | CADM’s HPID or GBP’s orgacct |
| ACCT\_1\_NB | String, A/N 25 | Nullable | Invoice Account Number, UB Sub Account, or MCN |
| ACCT\_2\_NB | String, A/N 25 | Nullable | SOC |
| ACCT\_3\_NB | String, A/N 25 | Nullable | GRC |
| ACCT\_TYPE\_CD | String, A/N 2 | Nullable | ‘I’ for Invoice Account Number, ‘AS’ for Sub account number, ‘AM’ for billing MCN triplet |
| BILLER\_CD | String, A/N 2 | Nullable | Example values:  ‘IG’ for GBP (Domestic), ‘UB’ for UB(Domestic), ‘LL’ for CR60 |
|  |  |  |  |
| TO\_BE\_DELETED | String, A/N 1 | Nullable | Y, or NULL |

|  |  |  |  |
| --- | --- | --- | --- |
| **ICORE.SITE2BILLING\_ACCOUNT** | | | |
| **Name** | **Datatype** | **Constraints** | **Comments** |
| ICORE\_SITE\_ID | NUMBER 10 | Icore\_site\_id+account\_key combined Primary Key |  |
| ACCOUNT\_KEY | String, A/N 13 | ~~Foreign Key~~ |  |
| TO\_BE\_DELETED | String, A/N 1 | Nullable | Y, or NULL |

<Ticket-000000224475375>

|  |  |  |  |
| --- | --- | --- | --- |
| **ICORE.CKT2BILLING\_ACCOUNT** | | | |
| **Name** | **Datatype** | **Constraints** | **Comments** |
| ICORE\_SITE\_ID | NUMBER 10 | Icore\_site\_id+seq\_no+account\_key combined Primary Key |  |
| SEQ\_NO | NUMBER 5 |  |  |
| ACCOUNT\_KEY | String, A/N 13 | ~~Foreign Key~~ |  |
| STATION\_NAME | String, A/N 100 | Nullable |  |
| TO\_BE\_DELETED | String, A/N 1 | Nullable | Y, or NULL |

</Ticket-000000224475375>

#### HLD-286278-DBA-GCP-ENTKEY-030 [METADATA Content]

The following data must be ensured to be added to resp. contained in the GCP Meta Data tables (ref. project 188413a). <REL1602>

<286282-US704600> added new data </286282-US704600>

<302503> Updated the spreadsheet with entries for new view “customer\_ubsubaccount\_vw”



#### HLD-279006-DBA-GCP-ENTKEY-031 [NETWORK\_TYPE]

The following new values must be ensured to be added into GDB.NETWORK\_TYPE.

|  |
| --- |
| **GDB.NETWORK\_TYPE <REL1602>** |
| **TYPE** |
| NST\_TUNNEL |
| NT\_TUNNEL |

#### HLD-295359-CR169138-GCP-GDB-from-ICORE-DBA-030 [ AVPN\_NETWORK\_CONNECTION\_DATA ]

<295359-US325561>

The following new columns must be ensured to be added into GDB.PHY\_PORT.

|  |  |  |  |
| --- | --- | --- | --- |
| **GDB.AVPN\_NETWORK\_CONNECTION\_DATA** | | | |
| **Name** | **Datatype** | **Constraints** | **Comments** |
| ACCESS\_CIRCUIT\_POK | string 100 | NOT NULL, CK |  |
| NETWORK\_CONNECTION\_POK | number(20) | NOT NULL, CK |  |
| Cust\_Asset\_Alias\_Name | String, A/N 100 | NULL |  |
| Connection\_Id | String, A/N 40 | NULL |  |
| Speed | Number (14,4) | NULL |  |
| ~~Port\_Address1~~ | ~~String, A/N 100~~ | ~~NULL~~ |  |
| ~~Port\_Address2~~ | ~~String, A/N 100~~ | ~~NULL~~ |  |
| ~~Prov\_Edge\_IP\_Data1\_Type~~ | ~~String, A/N 100~~ | ~~NULL~~ |  |
| ~~Prov\_Edge\_IP\_Data1\_Simple\_Network\_Address~~  ~~PROV\_EDGE\_IP\_DATA1\_NET\_ADDRESS~~ | ~~String, A/N 100~~ | ~~NULL~~ |  |
| ~~Prov\_Edge\_IP\_Data2\_Type~~ | ~~String, A/N 100~~ | ~~NULL~~ |  |
| ~~Prov\_Edge\_IP\_Data2\_Simple\_Network\_Address~~  ~~PROV\_EDGE\_IP\_DATA2\_NET\_ADDRESS~~ | ~~String, A/N 100~~ | ~~NULL~~ |  |
| Prov\_Edge\_Ingress\_Profile\_Id | String, A/N 100 | ~~NOT~~ NULL |  |
| Prov\_Edge\_Ingress\_Profile\_Name | String, A/N 100 | ~~NOT~~ NULL |  |
| Prov\_Edge\_Egress\_Profile\_Id | String, A/N 100 | ~~NOT~~ NULL |  |
| Prov\_Edge\_Egress\_Profile\_Name | String, A/N 100 | ~~NOT~~ NULL |  |
| Cos\_Detail | String, A/N 100 | NULL |  |
| Cust\_Edge\_IP\_Data~~1~~\_Type | String, A/N 100 | NULL |  |
| ~~Cust\_Edge\_IP\_Data1\_Simple\_Network\_Address~~  CUST\_EDGE\_IP\_DATA~~1~~\_NET\_ADDRESS | String, A/N 100 | NULL |  |
| ~~Cust\_Edge\_IP\_Data2\_Type~~ | ~~String, A/N 100~~ | ~~NULL~~ |  |
| ~~Cust\_Edge\_IP\_Data2\_Simple\_Network\_Address~~  ~~CUST\_EDGE\_IP\_DATA2\_NET\_ADDRESS~~ | ~~String, A/N 100~~ | ~~NULL~~ |  |
| Cust\_Edge\_Ingress\_Profile\_Id | String, A/N 100 | ~~NOT NULL~~  NULL |  |
| Cust\_Edge\_Ingress\_Profile\_Name | String, A/N 100 | ~~NOT NULL~~  NULL |  |
| Cust\_Edge\_Egress\_Profile\_Id | String, A/N 100 | ~~NOT NULL~~  NULL |  |
| Cust\_Edge\_Egress\_Profile\_Name | String, A/N 100 | ~~NOT NULL~~  NULL |  |
| Asset\_Cust\_NW\_Asset\_Type | String, A/N 100 | ~~NOT NULL~~  NULL |  |
| Asset\_Cust\_NW\_Asset\_Role | String, A/N 100 | NULL |  |
| NW\_Sub\_Type | String, A/N 100 | NULL |  |
| Bfd\_Interval | String, A/N 100 | NULL |  |
| Quality\_Of\_Service | String, A/N 100 | NULL |  |
| Asset\_Group\_Asset\_Type | String, A/N 100 | ~~NOT NULL~~  NULL |  |
| Asset\_Group\_Asset\_Role | String, A/N 100 | NULL |  |
| Managed\_Eqp\_Asset\_Type | String, A/N 100 | ~~NOT NULL~~  NULL |  |
| Managed\_Eqp\_Asset\_Role | String, A/N 100 | NULL |  |
| Service\_Option\_Name | String, A/N 50 | NULL |  |
| Service\_Option\_Display\_Name | String, A/N 100 | NULL |  |
| Network\_Type | String, A/N 100 | NULL |  |
| Dhv\_Flag | String, A/N 5 | NULL |  |
| Asset\_Cust\_NW\_POK | String, A/N 100 | NULL |  |
| Asset\_Cust\_NW\_NAME | String, A/N 100 | NULL |  |
| Asset\_Group\_POK | String, A/N 100 | NULL |  |
| Asset\_Group\_NAME | String, A/N 100 | NULL |  |
| Managed\_Eqp\_POK | String, A/N 100 | NULL |  |
| Managed\_Eqp\_NAME | String, A/N 100 | NULL |  |
| ID\_Change\_Tracking | Number (20) | Not Null |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **GDB.NETWORK\_IP\_DATA** | | | |
| **Name** | **Datatype** | **Constraints** | **Comments** |
| NETWORK\_CONNECTION\_POK | number(20) | NOT NULL |  |
| ID | Number (20) | NOT NULL, PK |  |
| Prov\_Edge\_IP\_Data\_Type | String, A/N 100 | NULL |  |
| PROV\_EDGE\_IP\_DATA\_NET\_ADDRESS | String, A/N 100 | NULL |  |
| Cust\_Edge\_IP\_Data\_Type | String, A/N 100 | NULL |  |
| Cust\_EDGE\_IP\_DATA\_NET\_ADDRESS | String, A/N 100 | NULL |  |
| ID\_Change\_Tracking | Number (20) | Not Null |  |

</295359-US325561>

#### HLD-295359-CR169138-GCP-GDB-from-ICORE-031 [ IPV4\_DATA ]

<295359-US325561>

|  |  |  |  |
| --- | --- | --- | --- |
| **IPV4\_DATA** | | | |
| **Element name** | **Type** | **Constraints** | **Comments** |
| NETWORK\_CONNECTION\_POK | number(20) | NOT NULL, PK |  |
| ~~IP\_Address\_V4~~ | ~~String, A/N 100~~ | ~~NOT NULL~~ |  |
| ~~subnet\_Mask\_V4~~ | ~~String, A/N 100~~ | ~~NOT NULL~~ |  |
| Asn\_Number | String, A/N 100 | NULL |  |
| Asn\_Override | String, A/N 100 | NULL |  |
| Restrictive\_Routing\_Fg | String, A/N 5 | NULL |  |
| md5\_Encryption\_Fg | String, A/N 5 | NULL |  |
| Mtu\_Size | NUMBER(38) | NULL |  |
| Provider\_Mtu\_Size | NUMBER(38) | NULL |  |
| Routing\_Protocol\_Ipv4\_Pe | String, A/N 100 | NULL |  |
| ID\_Change\_Tracking | Number (20) | Not Null |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **IPV4\_Address\_DATA** | | | |
| **Element name** | **Type** | **Constraints** | **Comments** |
| NETWORK\_CONNECTION\_POK | number(20) | NOT NULL, ~~CK~~ | <Defect 388187> |
| ID | Number(20) | NOT NULL, PK | <Defect 388187> |
| TYPE | String, A/N 100 | NULL | <Defect 388187> |
| IP\_Address\_V4 | String, A/N 100 | ~~NOT~~ NULL, ~~CK~~ | <Defect 388187> |
| subnet\_Mask\_V4 | String, A/N 100 | NULL |  |
| ID\_Change\_Tracking | Number (20) | Not Null |  |

</295359-US325561>

#### HLD-295359-CR169138-GCP-GDB-from-ICORE-032 [ IPV6\_DATA ]

<295359-US325561>

|  |  |  |  |
| --- | --- | --- | --- |
| **IPV6\_DATA** | | | |
| **Element name** | **Type** | **Constraints** | **Comments** |
| NETWORK\_CONNECTION\_POK | number(20) | NOT NULL, PK |  |
| ~~IP\_Address\_V6~~ | ~~String, A/N 100~~ | ~~NOT NULL~~ |  |
| ~~subnet\_Mask\_V6~~ | ~~String, A/N 100~~ | ~~NOT NULL~~ |  |
| ~~IP\_Prefix\_Length\_V6~~ | ~~String, A/N 3~~ | ~~NULL~~ |  |
| Wan\_IP\_Address | String, A/N 100 | NULL |  |
| Asn\_Number | String, A/N 100 | NULL |  |
| Asn\_Override | String, A/N 100 | NULL |  |
| Restrictive\_Routing\_Flag | String, A/N 5 | NULL |  |
| md5\_Encryption\_Flag | String, A/N 5 | NULL |  |
| Mtu\_Size | NUMBER(38) | NULL |  |
| Provider\_Mtu\_Size | NUMBER(38) | NULL |  |
| Routing\_Protocol\_Ipv6\_Pe | String, A/N 100 | NULL |  |
| ID\_Change\_Tracking | Number (20) | Not Null |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **IPV6\_ADDRESS\_DATA** | | | |
| **Element name** | **Type** | **Constraints** | **Comments** |
| NETWORK\_CONNECTION\_POK | number(20) | NOT NULL, ~~CK~~ | <Defect 388187> |
| ID | Number(20) | NOT NULL, PK | <Defect 388187> |
| TYPE | String, A/N 100 | NULL | <Defect 388187> |
| IP\_Address\_V6 | String, A/N 100 | ~~NOT~~ NULL, ~~CK~~ | <Defect 388187> |
| subnet\_Mask\_V6 | String, A/N 100 | NULL |  |
| IP\_Prefix\_Length\_V6 | String, A/N 3 | NULL |  |
| ID\_Change\_Tracking | Number (20) | Not Null |  |

</295359-US325561>

#### HLD-295359-CR169138-GCP-GDB-from-ICORE-033 [ RELATED\_ASSET\_DATA ]

<295359-US325561>

|  |  |  |  |
| --- | --- | --- | --- |
| **RELATED\_ASSET\_DATA** | | | |
| **Element name** | **Type** | **Constraints** | **Comments** |
| NETWORK\_CONNECTION\_POK | number(20) | NULL, PK |  |
| Related\_Asset\_Primary\_Obj\_Key | number(20) | NULL |  |
| Related\_Asset\_Name | Varchar2(100) | NULL |  |
| Asset\_Type | String, A/N 100 | NOT NULL |  |
| Asset\_Role | String, A/N 100 | NULL |  |
| Relation\_Type | String, A/N 50 | NOT NULL |  |
| ID\_Change\_Tracking | Number (20) | Not Null |  |

</295359-US325561>

#### HLD-295359-CR169138-GCP-GDB-from-ICORE-034 [ AVPN\_ACCESS\_CIRCUIT\_DATA ]

<295359-US325561>

|  |  |  |  |
| --- | --- | --- | --- |
| **AVPN\_ACCESS\_CIRCUIT\_DATA** |  |  |  |
| **Element name** | **Type** | **Constraints** | **Comments** |
| ACCESS\_CIRCUIT\_POK | string 100 | NOT NULL, PK |  |
| State | String, A/N 100 | NULL |  |
| Circuit\_Id | String, A/N 100 | NOT NULL |  |
| Customer\_Asset\_Alias\_Name | String, A/N 100 | NULL |  |
| Speed | Number (14,4) | NULL |  |
| Prov\_Port\_Speed | Number (14,4) | NULL |  |
| Prov\_port\_protocol | String, A/N 30 | NULL |  |
| Prov\_Mtu\_Size\_V4 | Integer | NULL |  |
| Prov\_Mtu\_Size\_V6 | Integer | NULL |  |
| Prov\_port\_Level\_Cos\_Exists\_Flag | String, A/N 5 | NULL |  |
| Prov\_port\_subRate\_Ethernet\_Flag | String, A/N 5 | NULL |  |
| Prov\_port\_NW\_Vlan\_Stacking\_Flag | String, A/N 5 | NULL |  |
| Prov\_port\_vlan\_Tag\_Control | String, A/N 100 | NULL |  |
| Prov\_port\_physical\_Interface | String, A/N 100 | NULL |  |
| Prov\_port\_ingress\_profile\_Id | String, A/N 100 | ~~NOT~~ NULL |  |
| Prov\_port\_ingress\_profile\_Name | String, A/N 100 | ~~NOT~~ NULL |  |
| Prov\_port\_egress\_profile\_Id | String, A/N 100 | ~~NOT~~ NULL |  |
| Prov\_port\_egress\_profile\_Name | String, A/N 100 | ~~NOT~~ NULL |  |
| Prov\_port\_cos\_Detail | String, A/N 20 | NULL |  |
| Cust\_Port\_Speed | Number (14,4) | NULL |  |
| Cust\_Port\_protocol | String, A/N 30 | NULL |  |
| Cust\_Port\_inv\_Site\_Id | Integer | NULL |  |
| Cust\_Port\_ingress\_profile\_Id | String, A/N 100 | ~~NOT~~ NULL |  |
| Cust\_Port\_ingress\_profile\_Name | String, A/N 100 | ~~NOT~~ NULL |  |
| Cust\_Port\_egress\_profile\_Id | String, A/N 100 | ~~NOT~~ NULL |  |
| Cust\_Port\_egress\_profile\_Name | String, A/N 100 | ~~NOT~~ NULL |  |
| Provisioning\_Access\_Arngmt\_Cd | String, A/N 10 | NULL |  |
| Cust\_Port\_NW\_Interface\_Jack | String, A/N 30 | NULL |  |
| ~~Available\_Bandwidth~~ |  |  |  |
| Cos\_Package | String, A/N 100 | NULL |  |
| Service\_Option\_Name | String, A/N 50 | NULL |  |
| Service\_Option\_Display\_Name | String, A/N 100 | NULL |  |
| Management\_Option\_Name | String, A/N 50 | NULL |  |
| Management\_Option\_Display\_Name | String, A/N 100 | NULL |  |
| Access\_Type | String, A/N 100 | NULL |  |
| Access\_Method\_Type | String, A/N 100 | NULL |  |
| Cust\_MTU\_Size\_V4 | String, A/N 100 | NULL |  |
| Cust\_MTU\_Size\_V6 | String, A/N 100 | NULL |  |
| Billing\_Option | String, A/N 100 | NULL |  |
| Min\_Bandwidth\_Commitment | VARCHAR2(10) | NULL |  |
| Access\_Inter\_connect | String, A/N 100 | NULL |  |
| Vendor | String, A/N 100 | NULL |  |
| Vendor\_Circuit\_Id\_Value | String, A/N 100 | NULL |  |
| IP\_Version | String, A/N 100 | NULL |  |
| Turn\_up\_Protocol | String, A/N 20 | NULL |  |
| Dlci | String, A/N10 | NULL |  |
| Mcn | String, A/N 100 | NULL |  |
| Grc | String, A/N 100 | NULL |  |
| Soc | String, A/N 100 | NULL |  |
| Access\_Prov\_Cd | String, A/N 4 | NULL |  |
| Ethernet\_Type | String, A/N 30 | NULL |  |
| Itu\_Carrier\_Cd | String, A/N ~~6~~ 8 | NULL |  |
| Diversity\_Option | String, A/N 100 | NULL |  |
| ID\_Change\_Tracking | Number (20) | Not Null |  |

</295359-US325561>

#### HLD-295359-CR169138-GCP-GDB-from-ICORE-034 [ EQUIPMENT\_DATA ]

<295359-US325561>

|  |  |  |  |
| --- | --- | --- | --- |
| **EQUIPMENT\_DATA** |  |  |  |
| **Element name** | **Type** | **Constraints** | **Comments** |
| ACCESS\_CIRCUIT\_POK | string 100 | NOT NULL, ~~PK~~ CK |  |
| EQUIPMENT\_POK | String 100 | NOT NULL, CK |  |
| Router\_host\_name | String, A/N 100 | NULL |  |
| vHNF\_Host\_name | String, A/N 100 | NULL |  |
| vHNF\_Customer\_Asset\_Alias | String, A/N 100 | NULL |  |
| Connected\_Circuit\_Id\_Value | String, A/N 100 | NULL |  |
| Isp\_Name | String, A/N 100 | NULL |  |
| Isp\_Speed\_Up | String, A/N 1000 | NULL |  |
| Isp\_Speed\_Down | String, A/N 1000 | NULL |  |
| Sd\_WAN\_Throughput | String, A/N 30 | NULL |  |
| Vhnf\_Part\_Number | String, A/N 50 | NULL |  |
| Ub\_Sub\_Account\_Id | String, A/N 13 | NULL |  |
| uCPE\_Host\_name | String, A/N 100 | NULL |  |
| uCPE\_Customer\_Asset\_Alias | String, A/N 100 | NULL |  |
| Ub\_Sub\_Account\_Id\_UCPE | String, A/N 13 | NULL |  |
| Management\_Option | String, A/N 100 | NULL |  |
| ID\_Change\_Tracking | Number (20) | Not Null |  |
| Equipment\_Type | String, A/N 100 | NULL |  |

#### HLD-295359-CR169138-GCP-GDB-from-ICORE-035 [ COS\_DETAIL\_INGRESS\_DATA ]

<295359-US325561>

|  |  |  |  |
| --- | --- | --- | --- |
| **COS\_DETAIL\_INGRESS\_DATA** |  |  |  |
| **Element name** | **Type** | **Constraints** | **Comments** |
| ID | Number(20) | NOT NULL, PK |  |
| Ingress\_profile\_Id | String, A/N 100 | NOT NULL | Same as Provider\_Edge\_Ingress\_profile\_id or Provider\_Port\_Ingress\_Profile\_Id |
| Ingress\_COS\_Class | String, A/N 20 | NOT NULL |  |
| Ingress\_Value | String, A/N 20 | NOT NULL |  |
| Ingress\_Policing | String, A/N 20 | NULL |  |
| Ingress\_Cos\_Element | String, A/N 20 | NULL |  |
| Ingress\_Cos\_Percentage | String, A/N 3 | NULL |  |
| ID\_Change\_Tracking | Number (20) | Not Null |  |

</295359-US325561>

#### HLD-295359-CR169138-GCP-GDB-from-ICORE-035 [ COS\_DETAIL\_INGRESS\_DATA ]

<295359-US325561>

|  |  |  |  |
| --- | --- | --- | --- |
| **COS\_DETAIL\_EGRESS\_DATA** |  |  |  |
| **Element name** | **Type** | **Constraints** | **Comments** |
| ID | Number(20) | NOT NULL, PK |  |
| Egress\_profile\_Id | String, A/N 100 | NOT NULL | Same as Provider\_Edge\_Egress\_profile\_id or Provider\_Port\_Egress\_Profile\_Id |
| Egress\_COS\_Class | String, A/N 20 | NOT NULL |  |
| Egress\_Value | String, A/N 20 | NOT NULL |  |
| Egress\_Queuing | String, A/N 20 | NULL |  |
| Egress\_Shaping | String, A/N 20 | NULL |  |
| Egress\_Cos\_Element | String, A/N 20 | NULL |  |
| Egress\_Cos\_Percentage | String, A/N 3 | NULL |  |
| ID\_Change\_Tracking | Number (20) | Not Null |  |

</295359-US325561>

### 

### End-to-End Key Search and Generation [ Icore Data Flow ]

<286284-US600815-US600161>

This requirement describe the processes to retrieve or generate end-to-end keys for assets/ports loaded from icore based on “Single ETE KEY Generation” section in GCP-SA-GDB-ETE\_KEY-HLD.

1. ETE\_ACCESS\_KEY

Pass the following data as input to “Single ETE KEY Generation”

|  |  |
| --- | --- |
| **INPUT Parameters** | |
| **Name** | **Comments** |
| ETE\_KEY\_NAME | “ETE\_ACCESS\_KEY” |
| INTERNAL\_KEY\_NAME | “ICORE\_CIRCUIT\_ID” |
| INTERNAL\_KEY\_VALUE | asset\_ext\_access\_circuit.circuit\_id\_value/icore.cust\_access.acc\_ckt |
| ID\_CHANGE\_TRACKING | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |

xt\_access\_circuit.circuit\_id\_valueed on " retrieve or generate end-to-end keys for assets/ports loaded from icore.

1. ETE\_PORT\_KEY

Pass the following data as input to “Single ETE KEY Generation”

|  |  |
| --- | --- |
| **INPUT Parameters** | |
| **Name** | **Comments** |
| ETE\_KEY\_NAME | “ETE\_PORT\_KEY” |
| INTERNAL\_KEY\_NAME | “ICORE\_SITE\_ID” |
| INTERNAL\_KEY\_VALUE | phy\_port.icore\_site\_id\_value/icore.site.site\_id |
| ID\_CHANGE\_TRACKING | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |

1. ETE\_SERVICE\_CONNECTION\_KEY

Pass the following data as input to “Single ETE KEY Generation”

|  |  |
| --- | --- |
| **INPUT Parameters** | |
| **Name** | **Comments** |
| ETE\_KEY\_NAME | “ETE\_SERVICE\_CONNECTION\_KEY” |
| INTERNAL\_KEY\_NAME | “ICORE\_PVC\_ID” |
| INTERNAL\_KEY\_VALUE | asset\_ext\_network\_connection.network\_connection\_id/icore.pvc.pvc\_id/usrp.tunnel\_vlan\_vw.icore\_pvc\_id (293772-CR160298)/icore.network\_connection.pvc\_id |
| ID\_CHANGE\_TRACKING | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |

1. ETE\_VPN\_KEY

Pass the following data as input to “Single ETE KEY Generation”

|  |  |
| --- | --- |
| **INPUT Parameters** | |
| **Name** | **Comments** |
| ETE\_KEY\_NAME | “ETE\_VPN\_KEY” |
| INTERNAL\_KEY\_NAME | “ICORE\_VPN\_ID” |
| INTERNAL\_KEY\_VALUE | asset\_ext\_customer\_network.network\_id |
| ID\_CHANGE\_TRACKING | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |

1. ETE\_SITE\_KEY

Pass the following data as input to “Single ETE KEY Generation”

|  |  |
| --- | --- |
| **INPUT Parameters** | |
| **Name** | **Comments** |
| ETE\_KEY\_NAME | “ETE\_SITE\_KEY” |
| INTERNAL\_KEY\_NAME | “ICORE\_SITE\_ID” |
| INTERNAL\_KEY\_VALUE | asset\_ext\_customer\_network.network\_id/icore.site.site\_id |
| ASSET\_ID | GDB.ASSET.id (ACCESS\_CIRCUIT) |
| ID\_CHANGE\_TRACKING | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |

1. ETE\_KEY\_SOURCE

For each found end-to-end key, search the ete\_key\_source as below:

select ete\_key\_source

from ete\_key\_map

where internal\_key\_value = <internal\_key\_value from above>

and internal\_key\_name = <internal\_key\_name from above>

and ete\_key\_name = <ete\_key\_name for the ete\_key\_value from above>

</286284-US600815-US600161>

### Service Delivery Data Flow into Golden Database [Icore Data Flow]

#### HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process]

For project 254035 customer/organization, location (site, address\_notation, location\_notation), service, asset (network, connection, circuit) and port data need to be loaded into the GCP GDB database in addition to other already existing source databases/systems.

Therefore, the following needs to be implemented:

Identify the changed (inserted records and records having updated columns that are copied into the GDB) from Icore and process only changed data into the GDB.

Use METADATA.SOURCE\_PROCESS.name=’ICORE\_TO\_GDB’ for this process.

Every execution of this process requires the maintenance of a change tracking event into GDB.CHANGE\_TRACKING, CHANGE\_SYSTEM and CHANGE\_USER. The change tracking event for the service delivery data flow must use CHANGE\_SYSTEM.name = ’ICORE\_TO\_GDB’ and CHANGE\_USER.name = ’ICORE\_TO\_GDB’. It is sufficient to have one change tracking event for all affected records during a single execution of this process.

Treat records that don’t qualify for selection constraints, or don’t qualify for selection constraints any more after an update, as logical deletes and delete those records from the GDB.

Process all inserted/updated/deleted records into the Data Index (ref. Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]).

The following HLD items specify the data mapping between Icore and the GDB per entity. Source database column content that is larger than in the GDB column definition must be truncated on the right side according to the GDB column definition.

Insert new records into the GDB.

Update changed (updated) records into the GDB.

Delete removed (deleted) properties from GDB entity objects (ORGANIZATION, SITE, ASSET (plus associated “satellite/extension” tables, if applicable), PORT (plus associated “satellite/extension” tables, if applicable), SERVICE) but not the entity object directly.

*Delete example:*

*(1) A CUSTOMER record in Icore gets deleted. Do not delete the Golden DB ORGANIZATION record immediately – the Golden DB ORGANIZATION table is an entity object and can have customer maintained contact data associated.*

*(2) An ASSET record (e.g. VPN) in Icore gets deleted. Do not delete the Golden DB ASSET record immediately – the Golden DB ASSET table is an entity object and can have customer maintained contact data associated. However, process that deletion directly into the corresponding Golden DB “satellite/extension” table (here: ASSET\_EXT\_CUSTOMER\_NETWORK) – the “satellite/extension” table associated with the Golden DB ASSET table is a property table.*

Identify new data entry combinations during processing for the following cases. Pass the following scenario information along with the ORGANIZATION.id and ORGANIZATION\_IDENTIFIER.value (Icore cust\_id), SITE.id, ASSET.id and ASSET\_IDENTIFIER.value (value + type depending on asset type (access circuit, customer network, network connection etc.)), SERVICE.id (building the organization-site-asset-service inventory hierarchy via associations) per new ASSET.id:

1. Existing organization, existing site, existing asset (inventory content change)
2. Existing organization, existing site, new asset
3. Existing organization, new site, new asset
4. New organization, new site, new asset

This process must execute the steps described in the following design items:

1. HLD-254035-GCP-FLOW-ICORE-010 [Icore Data Flow: Maintain MCN/GRC/SOC Mapping]
2. HLD-286282-US704600-GCP-FLOW-ICORE-020 [Icore Data Flow: Maintain Billing Account Mapping]
3. HLD-254035-GCP-FLOW-ICORE-100-A [Icore Data Flow into Organization CUSTOMER]
4. HLD-254035-GCP-FLOW-ICORE-100-B [Icore Data Flow into Organization MCN\_GRC\_SOC\_SOURCE]
5. HLD-286282-US704600-GCP-FLOW-ICORE-100-C [Icore Data Flow into Organization BILLING ACCOUNT]
6. HLD-286282-US704600-GCP-FLOW-ICORE-100-D [Icore Data Flow into Organization to Organization]
7. HLD-254035-GCP-FLOW-ICORE-101 [Icore Data Flow into Site]
8. HLD-254035-GCP-FLOW-ICORE-102-A [Icore Data Flow into Asset CUST\_ACCESS (“Access Circuit”)]
9. HLD-254035-GCP-FLOW-ICORE-102-B [Icore Data Flow into Asset CUST\_ACCESS\_CKT (“Access Circuit”)]
10. HLD-254035-GCP-FLOW-ICORE-103-A [Icore Data Flow into Asset VPN (Layer-3 “Customer Network”)]
11. HLD-254035-GCP-FLOW-ICORE-103-B [Icore Data Flow into Asset PVC (as VPN) (Layer-2 "Customer Network")]
12. HLD-254035-GCP-FLOW-ICORE-103-C [Icore Data Flow into Asset VPN (“Internet VLAN” “Customer Network”)]
13. HLD-254035-GCP-FLOW-ICORE-104-A [Icore Data Flow into Asset PVC (Layer-3 “Network Connection”)]
14. HLD-254035-GCP-FLOW-ICORE-104-B [Icore Data Flow into Asset PVC (Layer-2 "Network Connection")]
15. HLD-254035-GCP-FLOW-ICORE-104-C [Icore Data Flow into Asset PVC (“Internet VLAN” “Network Connection”)]
16. HLD-254035-GCP-FLOW-ICORE-110-A [Icore Data Flow into Phy Port SITE (Layer-3 “Phy Port”)]
17. HLD-254035-GCP-FLOW-ICORE-110-B [Icore Data Flow into Phy Port SITE (Layer-2 “Phy Port”)]
18. HLD-254035-GCP-FLOW-ICORE-110-D [Icore Data Flow into Phy Port SITE (“Interface Name” data)]
19. HLD-254035-GCP-FLOW-ICORE-110-E [Icore Data Flow into Phy Port SITE (“id\_asset Equipment” data)]
20. HLD-254035-GCP-FLOW-ICORE-120-A [Icore Data Flow into Log Port SITE (Layer-3 “Log Port”)]
21. HLD-254035-GCP-FLOW-ICORE-120-B [Icore Data Flow into Log Port SITE (Layer-2 “Log Port”)]
22. HLD-254035-GCP-FLOW-ICORE-120-C [Icore Data Flow into Log Port SITE (Internet VLAN “Log Port”)]
23. HLD-254035-GCP-FLOW-ICORE-120-D [Icore Data Flow into Log Port SITE (“Interface Name” data)]
24. HLD-254035-GCP-FLOW-ICORE-120-E [Icore Data Flow into Log Port SITE (“id\_asset Equipment” data)]
25. HLD-254035-GCP-FLOW-ICORE-130 [Icore Data Flow into Service]
26. HLD-254035-GCP-FLOW-ICORE-200-A [Icore Data Flow into Site to Organization (via “CUSTOMER”)]
27. HLD-254035-GCP-FLOW-ICORE-200-B [Icore Data Flow into Site to Organization (via “MRC\_GRC\_SOC\_SOURCE”)]
28. HLD-254035-GCP-FLOW-ICORE-201-A [Icore Data Flow into Asset (“Access Circuit”) to Site]
29. HLD-254035-GCP-FLOW-ICORE-201-B [Icore Data Flow into Asset (“Customer Network”) to Site]
30. HLD-254035-GCP-FLOW-ICORE-201-C [Icore Data Flow into Asset (“Network Connection”) to Site]
31. HLD-254035-GCP-FLOW-ICORE-202-A [Icore Data Flow into Asset (“Access Circuit”) to Service]
32. HLD-254035-GCP-FLOW-ICORE-202-B [Icore Data Flow into Asset (“Customer Network”) to Service]
33. HLD-254035-GCP-FLOW-ICORE-202-C [Icore Data Flow into Asset (“Network Connection”) to Service]
34. HLD-254035-GCP-FLOW-ICORE-203 [Icore Data Flow into Asset (“Access Circuit Parent”) to Asset]
35. HLD-254035-GCP-FLOW-ICORE-204 [Icore Data Flow into Port (“Access Circuit Port”) to Asset]
36. HLD-254035-GCP-FLOW-ICORE-205 [Icore Data Flow into Asset (“Network Connection Access Circuits”) to Asset]
37. HLD-254035-GCP-FLOW-ICORE-206-A [Icore Data Flow into Asset (“Customer Network for Network Connections”(Layer-3)) to Asset]
38. HLD-254035-GCP-FLOW-ICORE-206-B [Icore Data Flow into Asset (“Customer Network for Network Connections”(Layer-2)) to Asset]
39. HLD-254035-GCP-FLOW-ICORE-206-C [Icore Data Flow into Asset (“Customer Network for Network Connections”(Internet VLAN)) to Asset]
40. HLD-254035-GCP-FLOW-ICORE-207 [Icore Data Flow into Port (“Network Connection Port”) to Asset]
41. HLD-254035-GCP-FLOW-ICORE-209-A [Icore Data Flow into Asset (“Access Circuit”) to Organization CUSTOMER]
42. HLD-254035-GCP-FLOW-ICORE-209-B [Icore Data Flow into Asset (“Access Circuit”) to Organization MCN\_GRC\_SOC\_SOURCE]
43. HLD-286282-US704600-GCP-FLOW-ICORE-209-B2 [Icore Data Flow into Asset (“Access Circuit”) to Organization Billing Accounts]
44. HLD-254035-GCP-FLOW-ICORE-209-C [Icore Data Flow into Asset ( “Network Connection”) to Organization CUSTOMER]
45. HLD-254035-GCP-FLOW-ICORE-209-D [Icore Data Flow into Asset ( “Network Connection”) to Organization MCN\_GRC\_SOC\_SOURCE]
46. HLD-286282-US704600-GCP-FLOW-ICORE-209-D2 [Icore Data Flow into Asset ( “Network Connection”) to Organization Billing Accounts ]
47. HLD-254035-GCP-FLOW-ICORE-209-E [Icore Data Flow into Asset (Layer-3 “Customer Network”) to Organization CUSTOMER]
48. HLD-254035-GCP-FLOW-ICORE-209-F [Icore Data Flow into Asset (Layer-3 “Customer Network”) to Organization MCN\_GRC\_SOC\_SOURCE]
49. HLD-254035-GCP-FLOW-ICORE-209-G [Icore Data Flow into Asset (Layer-2 “Customer Network”) to Organization CUSTOMER]
50. HLD-254035-GCP-FLOW-ICORE-209-H [Icore Data Flow into Asset (Layer-2 “Customer Network”) to Organization MCN\_GRC\_SOC\_SOURCE]
51. HLD-254035-GCP-FLOW-ICORE-209-I [Icore Data Flow into Asset (Internet VLAN “Customer Network”) to Organization CUSTOMER]
52. HLD-254035-GCP-FLOW-ICORE-209-J [Icore Data Flow into Asset (Internet VLAN “Customer Network”) to Organization MCN\_GRC\_SOC\_SOURCE]
53. HLD-254035-GCP-FLOW-ICORE-CLEANUP [Icore Data Flow for Deletions]

**Logical data object relationships in ICORE:**



#### HLD-254035-GCP-FLOW-ICORE-010 [Icore Data Flow: Maintain MCN/GRC/SOC Mapping]

In ICORE the customer-related data is not fine-grained enough in respect to splitting it up to an MCN/GRC/SOC level. However, for being able to associate GDB objects with customers/organisations/accounts at the right level, it is needed to split a single ICORE customer record into as many records as different MCN/GRC/SOC combinations exist for that customer record.

In ICORE only the MCN data is reliable; GRC data is not always populated (and is on the “SITE” record). Therefore a separate process needs to be implemented to get the needed MCN/GRC/SOC triplet data. That process will have to use ICORE customer, site and customer access circuit data and try to find matching data in several other systems in order to be able to get MCN/GRC/SOC triplet data.

Using that process mentioned above, separate mapping tables in the GCP ICORE Replica need to be populated and maintained to enable mapping an ICORE customer record to MCN/GRC/SOC triplet data and from there associating location and asset data to an MCN/GRC/SOC triplet (and -\*NOT\*- to the “raw” ICORE customer record).

This step needs to be executed every time before loading customer/organisation data into GDB and before trying to associate data items with an organization.

Therefore, the following needs to be implemented:

**(1) Maintain the table “ICORE.MCN\_GRC\_SOC\_SOURCE”, step 1**

Using the process described below under “Logic for retrieving an MCN/GRC/SOC triplet and a POCITS\_MCN/GRC/SOC triplet for an ICORE customer”, do this:

* Update all records of “ICORE.MCN\_GRC\_SOC\_SOURCE” as follows:
  + Set “TO\_BE\_ADDED\_IN\_EKT” to ‘N’
  + Set “TO\_BE\_DELETED\_IN\_EKT” to ‘Y’
* For each triplet received from that process plus the “ICORE\_CUST\_ID” value as received from the initial SQL,   
  + check whether a corresponding record exists in “ICORE.MCN\_GRC\_SOC\_SOURCE” by comparing the MCN, GRC, SOC and ICORE\_CUST\_ID values (when comparing, use “<<NONE>>” for missing GRC and/or SOC values to get a match with the table’s contents !);  
    if a corresponding record exists in “ICORE.MCN\_GRC\_SOC\_SOURCE”, then
    - update it;   
      set “TO\_BE\_ADDED\_IN\_EKT” to ‘N’;  
      set “TO\_BE\_DELETED\_IN\_EKT” to ‘N’;  
      use the mapping from the embedded EXCEL file under “(2.2)” from that process description to compare and to set the ID\_MCN\_SOURCE, ID\_GRC\_SOURCE, ID\_SOC\_SOURCE column values;  
      for an empty GRC value use this row:  
       11 GRC << N/A >> << N/A >> << N/A >>  
      for an empty SOC value use this row:  
       14 SOC << N/A >> << N/A >> << N/A >>
  + If a corresponding record does not yet exist in “ICORE.MCN\_GRC\_SOC\_SOURCE”, then
    - insert it;   
      use the sequence “ICORE.mcn\_grc\_soc\_source\_id” to set the ID column value;  
      set “TO\_BE\_ADDED\_IN\_EKT” to ‘Y’;  
      set “TO\_BE\_DELETED\_IN\_EKT” to ‘N’;  
      for the column “GRC” use the value “<<NONE>>” in case of a missing GRC value;  
      for the column “SOC” use the value “<<NONE>>” in case of a missing SOC value;  
      use the mapping from the embedded EXCEL file under “(2.2)” from that process description to set the ID\_MCN\_SOURCE, ID\_GRC\_SOURCE, ID\_SOC\_SOURCE column values;  
      for an empty GRC value use this row:  
       11 GRC << N/A >> << N/A >> << N/A >>  
      for an empty SOC value use this row:  
       14 SOC << N/A >> << N/A >> << N/A >>

**(2) Maintain the Enterprise Key Translation (EKT) for the table “ICORE.MCN\_GRC\_SOC\_SOURCE”**

After having executed the step “(1)” above, the Enterprise Key Translation (EKT) needs to be maintained for the newly loaded data.

For this step do this:

* Delete all records from the EKT METADATA schema that are mapping to the “ID” value of “ICORE.MCN\_GRC\_SOC\_SOURCE” where “TO\_BE\_DELETED\_IN\_EKT” has the value ‘Y’.
* Insert as new records into the EKT METADATA schema where for the record in “ICORE.MCN\_GRC\_SOC\_SOURCE” the column “TO\_BE\_ADDED\_IN\_EKT” has the value ‘Y’.

NOTE: Here the internal working of the EKT in respect to data addition/updating and deletion is needed !

**(3) Maintain the table “ICORE.MCN\_GRC\_SOC\_SOURCE”, step 2**

For this step do this:

* Delete all records of “ICORE.MCN\_GRC\_SOC\_SOURCE” where “TO\_BE\_DELETED\_IN\_EKT” has the value ‘Y’.

**(4) Maintain the table “ICORE.MCN\_GRC\_SOC\_SITE\_IDS”**

Using the process described below under “Logic for retrieving an MCN/GRC/SOC triplet and a POCITS\_MCN/GRC/SOC triplet for an ICORE customer” do this:

* <BEGIN STAGING-DEFECT-FIX.2015-05-20>  
  Delete each record from “ICORE.MCN\_GRC\_SOC\_SITE\_IDS” for which no corresponding row exists from the result set as received from the initial SQL for that row’s “SITE\_ID” value and its associated MCN/GRC/SOC triplet(s), i.e. delete from “ICORE.MCN\_GRC\_SOC\_SITE\_IDS” all rows where no corresponding “SITE\_ID” value and its associated MCN/GRC/SOC triplet(s) exist as received from the initial SQL.  
  <END STAGING-DEFECT-FIX.2015-05-20>
* For each “SITE\_ID” value as received from the initial SQL and its associated MCN/GRC/SOC triplet(s),
  + check whether a corresponding record exists in “ICORE.MCN\_GRC\_SOC\_SITE\_IDS” by comparing the SITE\_ID values;  
    if a corresponding record exists in “ICORE.MCN\_GRC\_SOC\_SITE\_IDS”, then
    - check whether the source data for the MCN/GRC/SOC triplet has changed by comparing the data in “ICORE.MCN\_GRC\_SOC\_SOURCE” for the “ID\_M\_G\_S\_SOURCE” value from “ICORE.MCN\_GRC\_SOC\_SITE\_IDS” with the current MCN/GRC/SOC triplet values and the “ICORE\_CUST\_ID” value; (when comparing, use “<<NONE>>” for missing GRC and/or SOC values to get a match with the table’s contents !);  
        
      update the record if the source data has changed;   
      for the column “ID\_M\_G\_S\_SOURCE” use the “ID” value from the table “ICORE.MCN\_GRC\_SOC\_SOURCE” where a match exists between the MCN/GRC/SOC triplet associated with this “SITE\_ID” plus the associated “ICORE\_CUST\_ID” (when comparing, use “<<NONE>>” for missing GRC and/or SOC values to get a match with the table’s contents !);
    - do nothing if the source data has not changed
  + If a corresponding record does not yet exist in “ICORE.MCN\_GRC\_SOC\_SITE\_IDS”, then
    - insert it;   
      for the column “SITE\_ID” use the “SITE\_ID” value;  
      for the column “ID\_M\_G\_S\_SOURCE” use the “ID” value from the table “ICORE.MCN\_GRC\_SOC\_SOURCE” where a match exists between the MCN/GRC/SOC triplet associated with this “SITE\_ID” plus the associated “ICORE\_CUST\_ID” (when comparing, use “<<NONE>>” for missing GRC and/or SOC values to get a match with the table’s contents !);  
      <BEGIN ADD-2014-10-22>  
      NOTE:  
      If you get more than 1 "new" data item (MCN/GRC/SOC) per "site\_id", then insert all  
      <END ADD-2014-10-22>

**(5) Maintain the table “ICORE.MCN\_GRC\_SOC\_CIRCUIT\_IDS”**

Using the process described below under “Logic for retrieving an MCN/GRC/SOC triplet and a POCITS\_MCN/GRC/SOC triplet for an ICORE customer” do this:

* <BEGIN STAGING-DEFECT-FIX.2015-05-20>  
  Delete each record from “ICORE.MCN\_GRC\_SOC\_CIRCUIT\_IDS” for which no corresponding row exists from the result set as received from the initial SQL for that row’s “CIRCUIT\_ID\_SITE\_ID”/”CIRCUIT\_ID\_SEQ\_NO” pair value and its associated MCN/GRC/SOC triplet(s), i.e. delete from “ICORE.MCN\_GRC\_SOC\_CIRCUIT\_IDS” all rows where no corresponding “CIRCUIT\_ID\_SITE\_ID”/”CIRCUIT\_ID\_SEQ\_NO” pair value and its associated MCN/GRC/SOC triplet(s) exist as received from the initial SQL.  
  <END STAGING-DEFECT-FIX.2015-05-20>
* For each “CIRCUIT\_ID\_SITE\_ID”/”CIRCUIT\_ID\_SEQ\_NO” pair value as received from the initial SQL and its associated MCN/GRC/SOC triplet(s),
  + check whether a corresponding record exists in “ICORE.MCN\_GRC\_SOC\_CIRCUIT\_IDS” by comparing the CIRCUIT\_ID\_SITE\_ID and CIRCUIT\_ID\_SEQ\_NO values;  
    if a corresponding record exists in “ICORE.MCN\_GRC\_SOC\_CIRCUIT\_IDS”, then
    - check whether the source data for the MCN/GRC/SOC triplet has changed by comparing the data in “ICORE.MCN\_GRC\_SOC\_SOURCE” for the “ID\_M\_G\_S\_SOURCE” value from “ICORE.MCN\_GRC\_SOC\_CIRCUIT\_IDS” with the current MCN/GRC/SOC triplet values and the “ICORE\_CUST\_ID” value; (when comparing, use “<<NONE>>” for missing GRC and/or SOC values to get a match with the table’s contents !);  
        
      update the record if the source data has changed;   
      for the column “ID\_M\_G\_S\_SOURCE” use the “ID” value from the table “ICORE.MCN\_GRC\_SOC\_SOURCE” where a match exists between the MCN/GRC/SOC triplet associated with this “CIRCUIT\_ID\_SITE\_ID” and “CIRCUIT\_ID\_SEQ\_NO” plus the associated “ICORE\_CUST\_ID” (when comparing, use “<<NONE>>” for missing GRC and/or SOC values to get a match with the table’s contents !);
    - do nothing if the source data has not changed
  + If a corresponding record does not yet exist in “ICORE.MCN\_GRC\_SOC\_CIRCUIT\_IDS”, then
    - insert it;   
      for the column “CIRCUIT\_ID\_SITE\_ID” use the “CIRCUIT\_ID\_SITE\_ID” value;  
      for the column “CIRCUIT\_ID\_SEQ\_NO” use the “CIRCUIT\_ID\_SEQ\_NO” value;  
      for the column “ID\_M\_G\_S\_SOURCE” use the “ID” value from the table “ICORE.MCN\_GRC\_SOC\_SOURCE” where a match exists between the MCN/GRC/SOC triplet associated with this “SITE\_ID” plus the associated “ICORE\_CUST\_ID” (when comparing, use “<<NONE>>” for missing GRC and/or SOC values to get a match with the table’s contents !);  
      <BEGIN ADD-2014-10-22>  
      NOTE:  
      If you get more than 1 "new" data item (MCN/GRC/SOC) per "circuit id", then insert all  
      <END ADD-2014-10-22>

##### Logic for retrieving an MCN/GRC/SOC triplet and a POCITS\_MCN/GRC/SOC triplet for an ICORE customer

The logic for retrieving an MCN/GRC/SOC triplet for an ICORE customer is as follows:

(1)  
ICORE only contains MCN data for a customer in its “CUSTOMER” table. In addition it contains GRC data in its “SITE” table, but that data is not reliable. Therefore GRC and SOC data needs to be pulled from other databases. The data items that can be used for linking between different data from different databases are (1) the “ICORE.SITE.site\_id” and (2) the “ICORE” circuit id value. The following SQL statement gives the logic for pulling the data from ICORE that will be needed for the following data linking steps:

SELECT

distinct

trim(cu.CUST\_MCN) as ICORE\_CUST\_MCN,

case trim(si.GRC)

when 'TBD'

then null

else

trim(si.GRC)

end as ICORE\_CUST\_GRC,

trim(ca.acc\_ckt) as ICORE\_CIRCUIT\_ID,

si.SITE\_ID as SITE\_ID,

ca.SITE\_ID as CIRCUIT\_ID\_SITE\_ID,

0 as CIRCUIT\_ID\_SEQ\_NO,

cu.cust\_id as ICORE\_CUST\_ID,

p.prem\_country as prem\_country –- defect 22736 fix

FROM

icore.CUSTOMER cu,

icore.CUST\_ACCESS ca,

icore.SITE si,

icore.PORT\_ASGMT pa,

icore.PREMISE p -- defect 22736 fix

WHERE (1 = 1)

AND (cu.cust\_id = si.cust\_id)

AND (si.site\_id = ca.site\_id)

AND (si.site\_id = pa.site\_id)

AND (pa.port\_stat = 'IN-SERVICE')

AND (si.prem\_loc\_id = p.loc\_id ) –- defect 22736 fix

union

SELECT

distinct

trim(cu.CUST\_MCN) as ICORE\_CUST\_MCN,

case trim(si.GRC)

when 'TBD'

then null

else

trim(si.GRC)

end as ICORE\_CUST\_GRC,

trim(cac.circuit\_id) as ICORE\_CIRCUIT\_ID,

si.SITE\_ID as SITE\_ID,

cac.SITE\_ID as CIRCUIT\_ID\_SITE\_ID,

cac.SEQ\_NO as CIRCUIT\_ID\_SEQ\_NO,

cu.cust\_id as ICORE\_CUST\_ID,

p.prem\_country as pre\_country – defect 22736 fix

FROM

icore.CUSTOMER cu,

icore.CUST\_ACCESS\_CKT cac,

icore.SITE si,

icore.PORT\_ASGMT pa,

icore.PREMISE p -- defect 22736 fix

WHERE (1 = 1)

AND (cu.cust\_id = si.cust\_id)

AND (si.site\_id = cac.site\_id)

AND (si.site\_id = pa.site\_id)

AND (pa.port\_stat = 'IN-SERVICE')

AND (si.prem\_loc\_id = p.loc\_id ) –- defect 22736 fixorder by 1, 2, 3, 4

;

(2.1)

The following table gives an overview of where the data can be found in the different data sources we will be using.

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Source** | **Source Field for MCN** | **Source Field for GRC** | **Source Field for SOC** |
|  |  |  |  |
| **ICORE** | customer.cust\_mcn | site.grc | n/a |
|  |  |  |  |
| **USRP (1)** | port\_info.bill\_to\_mcn | port\_info.bill\_to\_grc | customer.soc |
| **USRP (2)** | customer.mcn | customer.grc | customer.soc |
| **GBP** | Trim(ORGMCN.MCN\_BASE) || Trim(ORGMCN.MCN\_SUFX) | ORGMCN.BILL\_GRP | n/a |
| **UIS.POCITS** | master\_customer\_number | Grc | soc |
| **~~CR60~~**  **~~<286282-US704600>~~** | ~~CIRCUIT\_SVC.MCN\_BASE||MCN\_SFX~~ | ~~BILL\_NO~~ | ~~SLS\_OFC~~  <Ticket-000000224475375> |

Information from ICORE comes with the base query as given above.

USRP appears twice as information is available on a port and customer level.

~~Defect-22736 and 286282-US704600:~~

~~Once data in MCN\_GRC\_SOC\_SOURCE is populated as described as above from all the sources (ICORE, USRP, GBP, UIS, and CR60). Implement the following process the populate icore.mcn\_grc\_soc\_source.triplet\_type.~~

~~For each record from MCN\_GRC\_SOC\_SOURCE and MCN\_GRC\_SOC\_SITE\_IDS tables :~~

~~select \*~~

~~from icore.mcn\_grc\_soc\_site\_ids a, icore.mcn\_grc\_soc\_source b~~

~~where a.id\_m\_g\_s\_source = b.id~~

~~Based on the mapping in section 2.2, check the source of mcn. Implement the following logic to determine the triplet type:~~

~~If any of mcn source is POCITS or CR60, populate the triplet\_type as ‘A’ for ‘ACCESS’. If the mcn sources are not POCITS or CR60, and prem\_country is USA, MEXICO, CANADA (after trimming off leading or trailing spaces), populate the triplet\_type as ‘AS’ for ‘ACCESS\_SERVICE’~~

~~If the mcn sources are not POCITS or CR60, and prem\_country is not USA, MEXICO, and CANADA, populate the triplet\_type as ‘S’ for ‘SERVICE’~~

~~Note: 22736 and 286282 may need a full data loading to populate triplet type or an one-time process to populate it for exsiting data in GDB organization\_identifier\_info and ICORE mcn\_grc\_soc\_source tables.~~

(2.2)

The following embedded EXCEL file gives a mapping between the data from the above table and the data from “ICORE.MCN\_GRC\_SOC\_SOURCE\_CODES” which is needed to maintain the “ICORE.MCN\_SOC\_GRC\_\*” (SOURCE/CIRCUIT\_IDS/SITE\_IDS) tables.

<286282-US704600> added new rows </286282-US704600> <Ticket-000000224475375> removed



(3)

Linkage to the other databases can be done using the following access logic:

1. USRP

| **USRP Table Name** | **Column Name(s)** | **Condition for selection** |
| --- | --- | --- |
| USRP.CUSTOMER,  USRP.PORT\_INFO | PORT\_INFO.icore\_site\_id,  PORT\_INFO.bill\_to\_mcn,  PORT\_INFO.bill\_to\_grc,  CUSTOMER.MCN,  CUSTOMER.GRC,  CUSTOMER.SOC | site\_id = <**icore site\_id from main icore query**>  and port\_info.customer\_id = customer.customer\_id |

|  |
| --- |
| **Sample Query**  NOTE: The following SQL is provided solely for illustrative purposes. It is the responsibility of GCP development team to create the appropriate SQL that takes into account query performance and overall processing efficiencies  SELECT  PORT\_INFO.icore\_site\_id,  PORT\_INFO.bill\_to\_mcn,  PORT\_INFO.bill\_to\_grc,  CUSTOMER.MCN,  CUSTOMER.GRC,  CUSTOMER.SOC  FROM  USRP.CUSTOMER,  USRP.PORT\_INFO  WHERE port\_info.customer\_id = customer.customer\_id  Note: Information is kept and retrieved per icore\_site\_id as key |

1. GBP

| **GBP Table Name** | **Column Name(s)** | **Condition for selection** |
| --- | --- | --- |
| ORGMCN,  ABDB\_FC FC | trim(FC.icore\_site\_id) as site\_id,  Trim(ORGMCN.MCN\_BASE) || Trim(ORGMCN.MCN\_SUFX) as MCN , ORGMCN.BILL\_GRP as GRC | site\_id = <**icore site\_id from main icore query**>  trim(ORGMCN.ORGACCT) = trim(FC.ORGACCT) |

|  |
| --- |
| **Sample Query**  NOTE: The following SQL is provided solely for illustrative purposes. It is the responsibility of GCP development team to create the appropriate SQL that takes into account query performance and overall processing efficiencies  select  trim(FC.icore\_site\_id) as site\_id,  Trim(ORGMCN.MCN\_BASE) || Trim(ORGMCN.MCN\_SUFX) as MCN , ORGMCN.BILL\_GRP as GRC  FROM  gbp1.ORGMCN,  gbp1.ABDB\_FC FC  WHERE  trim(ORGMCN.ORGACCT) = trim(FC.ORGACCT)  Note: Information is kept and retrieved per site\_id as key. This is the icore\_site\_id. |

1. POCITS (UIS)

| **UIS (POCITS)**  **Table Name** | **Column Name(s)** | **Condition for selection** |
| --- | --- | --- |
| uis\_pocits | ATT\_CKT\_ID,  MASTER\_CUSTOMER\_NUMBER,  GRC,  SOC | att\_ck\_id =  **<”**ICORE\_CIRCUIT\_ID**” from main icore query in normalized format (BMP format)**>[[1]](#footnote-1) |
|  |  |  |

|  |
| --- |
| **Sample Query**  NOTE: The following SQL is provided solely for illustrative purposes. It is the responsibility of GCP development team to create the appropriate SQL that takes into account query performance and overall processing efficiencies  select  ATT\_CKT\_ID,  MASTER\_CUSTOMER\_NUMBER,  GRC,  SOC  from  uis.uis\_pocits  Note: Information is kept and retrieved per att\_ckt\_id as key. This is the normalized format and not the format used in ICORE. Sample: It is *DHEC127819 ATI* and not *DHEC.127819..ATI or* 22HXGM 10062 SUV and not 22.HXGM.010062..SUV.  Steps to convert:  Check if there are 3 or 4 dots. If there are three dots: Contacte first word, (remove trailing zeroes from second word), left space padded third word (*DHEC.127819.801.ATI becomes DHEC127819801ATI.* If there are 4 dots: Contacte first word, second word, (remove trailing zeroes from third word), left space padded fourth word, fifith word (22.HXGM.010062..SUV becomes 22HXGM 10062 SUV*.* |

1. ~~CR60 <286282-US704600>~~ <Ticket-000000224475375> Removed

~~CR60 data should be loaded after ICORE, USRP, GBP, and UIS (POCITS) data loaded. CR60 data could retrieved as below:~~

| **~~CR60~~**  **~~Table Name~~** | **~~Column Name(s)~~** | **~~Condition for selection~~** |
| --- | --- | --- |
| ~~CIRCUIT\_SVC~~ | ~~CIRCUIT\_ID,~~  ~~MCN\_BASE||MCN\_SFX,~~  ~~BILL\_NO as GRC,~~  ~~SLS\_OFC as SOC~~ | ~~CIRCUIT\_ID (after remove all special characters) =  <”ICORE\_CIRCUIT\_ID” from main icore query after remove all special characters>~~  ~~And MCN\_BASE is not null~~  ~~Note: only compare alphabets and digits.~~ |
| ~~CIRCUIT\_ADDRESS~~ | ~~Station\_name~~  ~~Retrieve randomly one station name for each circuit id.~~ | ~~CIRCUIT\_SVC.CIRCUIT\_ID = CIRCUIT\_ADDRESS.CIRCUIT\_ID~~ |

~~CR60 data should be loaded as described as below:~~

~~If the MCN/GRC/SOC is already in MCN\_GRC\_SOC\_SOURCE, and triplet\_type is ‘S’ (SERVICE), update the triplet\_type with ‘AS’ (ACCESS\_SERVICE), otherwise, keep it as it is. If the MCN/GRC/SOC is not in MCN\_GRC\_SOC\_SOURCE table, create new mcn/grc/soc record in it with triplet\_type ‘A’ (ACCESS). Note: any unpopulated GRC and SOC will be populated with ‘<<NONE>>’.~~

(4)

The preference for the data collection is as follows:

**(A)**  
For creating an MCN/GRC/SOC triplet value **based on an ICORE MCN**,

check the data available from the data sources above for a given ICORE\_CUST\_MCN and ICORE\_SITE\_ID pair (use the corresponding ICORE\_CIRCUIT\_ID value for linking into POCITS; see above)

(A.1) For the “MCN” value:  
 Always use the ICORE “CUST\_MCN” value from the original base query

(A.2) For the “GRC”/”SOC” values, take the values from the first applicable “If” below:  
 If “USRP.port\_info.bill\_to\_grc” has a value, then use that one for “GRC”;  
 use the corresponding “USRP.customer.soc” value for “SOC”.

If “USRP.customer.grc” has a value, then use that one for “GRC”;  
 use the corresponding “USRP.customer.soc” value for “SOC”.

If “UIS.UIS\_POCITS.grc” has a value, then use that one for “GRC”;  
 use the corresponding “UIS.UIS\_POCITS.soc” value for “SOC”.

If “ORGMCN.BILL\_GRP” has a value, then use that one for “GRC”;  
 leave the “SOC” value empty.

If “ICORE\_CUST\_GRC” has a value, then use that one for “GRC”;  
 leave the “SOC” value empty.

If none of the above has a value,

then leave the “GRC” and “SOC” value empty.

(A.3) <defect-22736> <286282-US704600> if prem\_country is USA, MEXICO, CANADA (after trimming off leading or trailing spaces), and no new MCN/GRC/SOC from POCITS (described in B~~), and no different MCN/GRC/SOC from CR60~~(<Ticket-000000224475375>) for the same circuit id, populate the triplet\_type as ‘AS’ for ACCESS\_SERVICE, and otherwise, populate the triplet\_type as ‘S’ for SERVICE.

**(B)**  
For creating a corresponding MCN/GRC/SOC triplet value **based on a POCITS\_MCN**,

use each created MCN/GRC/SOC triplet as created under “(A)” and check whether a corresponding record from “UIS.UIS\_POCITS” exists using the data from the original ICORE SQL query for mapping;

if that is the case, then check first whether the <defect-22736> any of “UIS.UIS\_POCITS.master\_customer\_number”, “UIS.UIS\_POCITS.grc”, or “UIS.UIS\_POCITS.soc” value is different from the “MCN”, “GRC”, “SOC” value of the “normal” data record from “(A)”;

if that is the case, then create a second MCN/GRC/SOC triplet (based on a POCITS\_MCN) by taking the current MCN/GRC/SOC triplet and replacing its MCN value with the value from “UIS.UIS\_POCITS.master\_customer\_number”. If this POCITS\_MCN-based MCN/GRC/SOC triplet already exists in the whole set of triplets, then discard it again.

Make sure each such “POCITS” MCN/GRC/SOC triplet (not discarded) gets the same associations (see "(C)" below) as its original MCN/GRC/SOC triplet. <286282-US704600> Populate triplet\_type with ‘A’ (ACCESS) for those newly created mcn/grc/soc from POCITS.</286282-US704600>.

**(C)**  
Make sure the association between the triplets found (including the triplets that have been generated from POCITS\_MCN ~~<286282-US704600>or CR60 MCN </286282-US704600>~~ (<Ticket-000000224475375>) values) and the data from the original ICORE SQL query is kept, as it will be needed during the 3 maintenance processes given above.

#### HLD-286282-US704600-GCP-FLOW-ICORE-020 [Icore Data Flow: Maintain Billing Account Mapping]

This requirement describe the process to look billing accounts (billing mcn triplet, invoice account number, ub sub account number) for avpn assets from either CADM or GBP.

This step needs to be executed every time before loading customer/organisation data into GDB and before trying to associate data items with an organization.

This step needs to be executed every time after HLD-254035-GCP-FLOW-ICORE-010 [Icore Data Flow: Maintain MCN/GRC/SOC Mapping].

Before starting below data loading process, update all records’ TO\_BE\_DELETED with ‘Y’ for tables GDB\_BILLING\_ACCOUNT, CKT2BILLING\_ACCOUNT (Ticket-000000224475375), and SITE2BILLING\_ACCOUNT.

For each site\_id in MCN\_GRC\_SOC\_SITE\_IDS, search cadm for billing accounts as below:

select a.port\_nb as icore\_site\_id, E.HIER\_PNT\_ID as account\_key, e.acct\_1\_nb, e.acct\_2\_nb, e.acct\_3\_nb, E.ACCT\_TYPE\_CD, 'UB' as biller\_cd

from CADM.port\_TB a, CADM.SUBS\_BLNG\_ASSOC\_TB b, CADM.BUS\_ARNG\_TB c, CADM.HIER\_PNT\_ASSOC\_TB d,

CADM.HIER\_PNT\_TB e

where a.port\_nb = <icore\_site\_id>

and (a.port\_end\_dt is null or a.port\_end\_dt>sysdate)

and B.PARNT\_FK\_ID = a.port\_id

**and B.parnt\_fk\_type\_cd = 'PO' – Defect 47770**

and (B.SUBS\_BLNG\_END\_DT is null or B.SUBS\_BLNG\_END\_DT > sysdate)

and b.bus\_arng\_id = c.bus\_arng\_id

and C.HIER\_PNT\_ID=D.HIER\_PNT\_ID

and d.PARNT\_HIER\_PNT\_ID = e.hier\_pnt\_id

and (e.hier\_pnt\_end\_dt is null or e.hier\_pnt\_end\_dt > sysdate )

and (c.bus\_arng\_end\_dt is null or c.bus\_arng\_end\_dt > sysdate)

and (e.hier\_pnt\_end\_dt is null or e.hier\_pnt\_end\_dt > sysdate )

and E.UB\_ACCT\_TYPE\_CD in ('I', 'AS')

Above query could return 2 records, one for invoice account number (‘I’), and one for ub sub account number (‘AS’)

Take the account\_key above for UB sub account number (‘AS’), to retrieve mcn, grc, soc as below:

select b.hier\_pnt\_id as account\_key, b.acct\_1\_nb, b.acct\_2\_nb, b.acct\_3\_nb, B.ACCT\_TYPE\_CD, 'UB' as biller\_cd

from CADM.HIER\_PNT\_ASSOC\_TB a, CADM.HIER\_PNT\_TB b

where A.PARNT\_HIER\_PNT\_ID = <account\_key>

and A.HIER\_PNT\_ID = B.HIER\_PNT\_ID

and (A.HIER\_PNT\_ASSOC\_END\_DT is null or A.HIER\_PNT\_ASSOC\_END\_DT >sysdate)

and (B.HIER\_PNT\_END\_DT is null or B.HIER\_PNT\_END\_DT > sysdate )

and b.UB\_ACCT\_TYPE\_CD = 'AM' and not exists ( select 1 from MCN\_GRC\_SOC\_SOURCE a, MCN\_GRC\_SOC\_SITE\_IDS b

where mcn = b.acct\_1\_nb and grc = decode(b.acct\_3\_nb, null, ‘<<NONE>>’, b.acct\_3\_nb)

and soc = decode(b.acct\_2\_nb, null, ‘<<NONE>>’, b.acct\_2\_nb)

and a.id = b.id\_m\_g\_s\_source and b.site\_id = <icore\_site\_id for account\_key> )

if above queries don’t return any data, try again to retrieve billing accounts from GBP as below:

select distinct trim(orgacct)||’I’ as account\_key, c.custnum as acct\_1\_nb, null as acct\_3\_nb, 'I' as ACCT\_TYPE\_CD,

decode(trim(C.BUS\_OWNER\_ID), 'US', 'IG', null) as biller\_cd

from GBP1.ABDB\_FC a, GBP1.OSDB\_ORGACT b, gbp1.osdb\_cust c

where icore\_site\_id like '<icore\_site\_id>%'

and a.orgacct = b.orgacct

and B.CUSTNUM = c.custnum

union

select distinct trim(b.orgacct)||’AM’ as account\_key, trim(D.MCN\_BASE)||trim(D.MCN\_SUFX) as acct\_1\_nb, trim(D.BILL\_GRP) as acct\_3\_nb, 'AM' as ACCT\_TYPE\_CD,

decode(trim(C.BUS\_OWNER\_ID), 'US', 'IG', null) as biller\_cd – gbp doesn’thave soc

from GBP1.ABDB\_FC a, GBP1.OSDB\_ORGACT b, gbp1.osdb\_cust c, GBP1.ORGMCN d

where icore\_site\_id like '<icore\_site\_id>%'

and a.orgacct = b.orgacct

and B.CUSTNUM = c.custnum

and a.orgacct = d.orgacct

and trim(D.MCN\_BASE)||trim(D.MCN\_SUFX) is not null

and not exists ( select 1 from MCN\_GRC\_SOC\_SOURCE a, MCN\_GRC\_SOC\_SITE\_IDS b

where mcn = trim(D.MCN\_BASE)||trim(D.MCN\_SUFX)

and grc = decode(trim(D.BILL\_GRP), null, ‘<<NONE>>’, trim(D.BILL\_GRP)

and a.id = b.id\_m\_g\_s\_source and b.site\_id = <icore\_site\_id for account\_key> )

<Ticket-000000224475375>

For each CIRCUIT\_ID\_SITE\_ID, or CIRCUIT\_ID\_SITE\_ID/CIRCUIT\_ID\_SEQ\_NO in MCN\_GRC\_SOC\_CIRCUIT\_IDS, search CR60 for billing accounts as below:

Match the CIRCUIT\_ID\_SITE\_ID with icore.cust\_access.site\_id, and match CIRCUIT\_ID\_SITE\_ID/CIRCUIT\_ID\_SEQ\_NO with icore.cust\_access\_ckt.site\_id/seq\_no, and retrieve the cust\_access.acc\_ckt/cust\_access\_ckt.circuit\_id and remove all special characters from it (only keep alphabets and digits). Match it with CR60’s CIRCUIT\_SVC.CIRCUIT\_ID (after removed all special characters), and retrieve MCN\_BASE||MCN\_SFX as acct\_1\_nb, BILL\_NO as acct\_3\_nb, and SLS\_OFC as acct\_2\_nb, icore.cust\_access.site\_id||’XX’ as account\_key, ‘XX’ as acct\_type\_cd, ‘LL’ as biller\_cd. Meanwhile, ensure the acct\_1\_nb is populated, otherwise drop the record.

Retrieve CR60’s CIRCUIT\_ADDRESS.station\_name by matching the circuit\_id with CIRCUIT\_SVC.circuit\_id. In case multiple station\_name is returned, pick any populated one.

</Ticket-000000224475375>

All account\_keys from above queries and icore site id should be inserted/updated into SITE2BILLIN\_ACCOUNT (acct\_type\_cd !=’XX’ <Ticket-000000224475375>) table depending on if they are in the table or not. All records updated or inserted into the table should have NULL value in TO\_BE\_DELETED.

<Ticket-000000224475375>

All account\_keys (including station\_name) from above queries and icore site id/seq\_no (populate 0 for null) should be inserted/updated into CKT2BILLIN\_ACCOUNT (acct\_type\_cd =’XX’) table depending on if they are in the table or not. All records updated or inserted into the table should have NULL value in TO\_BE\_DELETED.

</Ticket-000000224475375>

All account\_keys and acct\_1\_nb, acct\_2\_nb, acct\_3\_nb, and ACCT\_TYPE\_CD should be inserted/updated into GDB\_BILLING\_ACCOUNT table depending on if they are in the table or not. All records updated or inserted into the table should have NULL value in TO\_BE\_DELETED.

Delete all records from the EKT METADATA schema that are mapping to the “account\_key” value of “ICORE.GDB\_BILLING\_ACCOUNT” and “ICORE.SITE2BILLING\_ACCOUNT” where “TO\_BE\_DELETED” has the value ‘Y’.

NOTE: Here the internal working of the EKT in respect to data addition/updating and deletion is needed !

Delete all records of “ICORE. GDB\_BILLING\_ACCOUNT”, “CKT2BILLING\_ACCOUNT” (<Ticket-000000224475375>) and “SITE2BILLING\_ACCOUNT” where “TO\_BE\_DELETED” has the value ‘Y’

#### HLD-254035-GCP-FLOW-ICORE-100-A [Icore Data Flow into Organization CUSTOMER]

This requirement tries to load ICORE customer data into GDB where the identifier is the “CUSTOMER.cust\_id” value. This data, however, has usually not the sufficient amount of granularity.

Implement the following:

Process the following Icore records into the GDB organization object and into the Data Index (ref. Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION column** | **Processing comment** |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘ORGANIZATION’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | id\_organization\_type | Referencing ORGANIZATION\_TYPE ‘SERVICE\_SPECIFIC\_CUSTOMER\_REPRESENTATION’ |
| CUSTOMER.cust\_id | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| CUSTOMER.cust\_name | Name |  |
| - | id\_address | NULL |
| - | is\_verification\_opted\_out | NULL |
| - | id\_organization\_unified | NULL, postprocess according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules]. |

Customer Identifier (1): ICORE Customer ID  
(only set if data is available !)

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION\_IDENTIFIER column** | **Processing comment** |
| - | id\_organization | ORGANIZATION.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘ICORE\_CUST\_ID’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_organization\_identifier | ORGANIZATION\_IDENTIFIER.id |
| - | id\_change\_tracking | ORGANIZATION\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘ICORE\_CUST\_ID’ |
| CUSTOMER.cust\_id | Value |  |

~~Customer Identifier (2): MCN Value~~ (Do not use any more; 2013-09-09; hc2164)

|  |  |  |
| --- | --- | --- |
| **~~Icore table.column~~** | **~~GDB ORGANIZATION\_IDENTIFIER column~~** | **~~Processing comment~~** |
| ~~-~~ | ~~id\_organization~~ | ~~ORGANIZATION.id~~ |
| ~~-~~ | ~~id\_change\_tracking~~ | ~~See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process]~~ |
| ~~-~~ | ~~id\_identifier\_type~~ | ~~Referencing IDENTIFIER\_TYPE ‘MCN’~~ |

|  |  |  |
| --- | --- | --- |
| **~~Icore table.column~~** | **~~GDB ORGANIZATION\_IDENTIFIER\_VALUE column~~** | **~~Processing comment~~** |
| ~~-~~ | ~~id\_organization\_identifier~~ | ~~ORGANIZATION\_IDENTIFIER.id~~ |
| ~~-~~ | ~~id\_change\_tracking~~ | ~~ORGANIZATION\_IDENTIFIER.id\_change\_tracking~~ |
| ~~-~~ | ~~id\_identifier\_type~~ | ~~Referencing IDENTIFIER\_TYPE ‘MCN’~~ |
| ~~CUSTOMER.cust\_mcn~~ | ~~Value~~ |  |

<302503>

Customer Identifier (1): UBSubAccount  
(only set if data is available !)

Create a view ICORE.CUSTOMER\_UBSUBACCOUNT\_VW

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION column** | **Processing comment** |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘ORGANIZATION’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | id\_organization\_type | Referencing ORGANIZATION\_TYPE ‘SERVICE\_SPECIFIC\_CUSTOMER\_REPRESENTATIO’ |
| CUSTOMER\_UBSUBACCOUNT\_VW.site\_id | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| CUSTOMER.cust\_name | Name | Via customer.cust\_id = customer\_ubsubaccount\_vw.cust\_id |
| - | id\_address | NULL |
| - | is\_verification\_opted\_out | NULL |
| - | id\_organization\_unified | NULL, postprocess according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules]. |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION\_IDENTIFIER column** | **Processing comment** |
| - | id\_organization | ORGANIZATION.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘UB\_ACCOUNT\_ID’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_organization\_identifier | ORGANIZATION\_IDENTIFIER.id |
| - | id\_change\_tracking | ORGANIZATION\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘UB\_ACCOUNT\_ID’ |
| CUSTOMER\_UBSUBACCOUNT\_VW.ubs~~ubaccountnumber~~ ub\_sub\_acct\_num <Defect 608684> | Value |  |

</302503>HLD-254035-GCP-FLOW-ICORE-100-B [Icore Data Flow into Organization MCN\_GRC\_SOC\_SOURCE]

This requirement tries to load ICORE customer data into GDB where the identifier is an MCN/GRC/SOC triplet for a single ICORE “CUSTOMER” record identified by “CUSTOMER.cust\_id”. This MCN/GRC/SOC triplet data has usually a much better granularity than the “CUSTOMER.cust\_id” data alone.

This requirement is dependent on “HLD-254035-GCP-FLOW-ICORE-010 [Icore Data Flow: Maintain MCN/GRC/SOC Mapping]”, which needs to be executed before this requirement can be executed.

Implement the following:

Process the following Icore records into the GDB organization object and into the Data Index (ref. Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

For each record pulled from “ICORE.CUSTOMER” proceed as follows:

Use the following database traversal and mapping to find all records in “ICORE.MCN\_GRC\_SOC\_SOURCE” that aplly to one such record:

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| CUSTOMER.cust\_id | MCN\_GRC\_SOC\_SOURCE.icore\_cust\_id |

Usually at least one record should be found (but there may be cases where nothing can be found), but usually there is a 1-to-n relationship between “CUSTOMER” and “MCN\_GRC\_SOC\_SOURCE”.

For each record found, process it into GDB as follows:

**(1) Full MCN/GRC/SOC triplet data found in the record**

This is the case if all three columns “MCN”, “GRC” and “SOC” have values that are not equal to “<<NONE>>”.

In this case process the record from “MCN\_GRC\_SOC” in GDB as follows:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION column** | **Processing comment** |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘ORGANIZATION’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | id\_organization\_type | Referencing ORGANIZATION\_TYPE ‘SERVICE\_SPECIFIC\_CUSTOMER\_REPRESENTATION’ |
| MCN\_GRC\_SOC\_SOURCE.id | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| CUSTOMER.cust\_name | name | (via “MCN\_GRC\_SOC\_SOURCE.icore\_cust\_id”) |
| - | id\_address | NULL |
| - | is\_verification\_opted\_out | NULL |
| - | id\_organization\_unified | NULL, postprocess according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules]. |

Customer Identifier (1): Standard MCN/GRC/SOC Triplet  
(only set if data is available, i.e. a full triplet exists !)

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION\_IDENTIFIER column** | **Processing comment** |
| - | id\_organization | ORGANIZATION.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘MCN\_GRC\_SOC’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_organization\_identifier | ORGANIZATION\_IDENTIFIER.id |
| - | id\_change\_tracking | ORGANIZATION\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘MCN’ |
| MCN\_GRC\_SOC\_SOURCE.mcn | Value |  |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_organization\_identifier | ORGANIZATION\_IDENTIFIER.id |
| - | id\_change\_tracking | ORGANIZATION\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘GRC’ |
| MCN\_GRC\_SOC\_SOURCE.grc | Value |  |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_organization\_identifier | ORGANIZATION\_IDENTIFIER.id |
| - | id\_change\_tracking | ORGANIZATION\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘SOC’ |
| MCN\_GRC\_SOC\_SOURCE.soc | Value |  |

<Defect-70085-Fix>

|  |  |  |
| --- | --- | --- |
| **USRP/GRID table.column** | **GDB ORGANIZATION\_IDENTIFER\_INFO column** | **Processing comment** |
| - | id\_organizaiton\_identifier | ORGANIZATION\_IDENTIFIER.id |
| - | id\_change\_tracking | ORGANIZATION\_IDENTIFIER.id\_change\_tracking |
| MCN\_GRC\_SOC\_SOURCE.TRIPLET\_TYPE  (defect 22736 fix)  Or  GRID.ACCOUNT.TRIPLET\_TYPE | ACCOUNT\_USAGE\_TYPE | select distinct a.customer\_name  from usrp.customer a where a.mcn = <MCN\_GRC\_SOC\_SOURCE.mcn>  and a.grc = <MCN\_GRC\_SOC\_SOURCE.grc>  and a.soc = <MCN\_GRC\_SOC\_SOURCE.soc>  <Defect-229121>  select distinct a.biller\_code  from usrp.mcn\_grc\_soc a where a.mcn = <MCN\_GRC\_SOC\_SOURCE.mcn>  and a.grc = <MCN\_GRC\_SOC\_SOURCE.grc>  and a.soc = <MCN\_GRC\_SOC\_SOURCE.soc>  select account\_type, triplet\_type, billing\_cust\_name, account\_number, biller\_code  from grid.account a  where mcn = <MCN\_GRC\_SOC\_SOURCE.mcn>  and grc = <MCN\_GRC\_SOC\_SOURCE.grc>  and soc = <MCN\_GRC\_SOC\_SOURCE.grc> |
| USRP.CUSTOMER.BILLER\_CODE/ GRID.ACCOUNT.BILLER\_CODE | BILLER\_CODE |  |
| USRP.CUSTOMER.CUSTOMER\_NAME/ GRID.ACCOUNT.BILLING\_CUST\_NAME | BILLING\_CUST\_NAME |  |
| GRID.ACCOUNT.ACCOUNT\_NUMBER | BILLING\_ACCOUNT\_NUMBER | Note: <QC40601>  If this will not return the result, use the following query:  select c.acct\_1\_nb as sub\_account\_number, c.ACCT\_TYPE\_CD from cadm.hier\_pnt\_tb a, cadm.hier\_pnt\_assoc\_tb b, cadm.hier\_pnt\_tb c where a.acct\_1\_nb = '<mcn>' and a.acct\_2\_nb = '<soc>' and a.acct\_3\_nb = '<grc>' and a.ACCT\_TYPE\_CD = 'AM' and a.hier\_pnt\_id=b.hier\_pnt\_id and b.parnt\_hier\_pnt\_id=c.hier\_pnt\_id and c.ACCT\_TYPE\_CD = 'AS'  </QC40601> |
| GRID.ACCOUNT.ACCOUNT\_TYPE | ACCOUNT\_TYPE |  |

</Defect-70085-Fix>

**~~(2) MCN/GRC pair data found in the record~~ (no longer supported; 2013-09-09; hc2164)**

~~This is the case if the columns “MCN” and “GRC” have values that are not equal to “<<NONE>>”, but where the “SOC” value is “<<NONE>>”.~~

~~In this case process the record from “MCN\_GRC\_SOC” in GDB as follows:~~

|  |  |  |
| --- | --- | --- |
| **~~Icore table.column~~** | **~~GDB ORGANIZATION column~~** | **~~Processing comment~~** |
| ~~-~~ | ~~id\_object\_type~~ | ~~Referencing OBJECT\_TYPE ‘ORGANIZATION’~~ |
| ~~-~~ | ~~id\_change\_tracking~~ | ~~See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process]~~ |
| ~~-~~ | ~~is\_read\_only~~ | ~~‘Y’~~ |
| ~~-~~ | ~~id\_organization\_type~~ | ~~Referencing ORGANIZATION\_TYPE ‘SERVICE\_SPECIFIC\_CUSTOMER\_REPRESENTATION’~~ |
| ~~MCN\_GRC\_SOC\_SOURCE.id~~ | ~~id [via Enterprise Key Translation]~~ | ~~Use SOURCE\_KEY.id into GDB~~ |
| ~~CUSTOMER.cust\_name~~ | ~~name~~ | ~~(via “MCN\_GRC\_SOC\_SOURCE.icore\_cust\_id”)~~ |
| ~~-~~ | ~~id\_address~~ | ~~NULL~~ |
| ~~-~~ | ~~is\_verification\_opted\_out~~ | ~~NULL~~ |
| ~~-~~ | ~~id\_organization\_unified~~ | ~~NULL, postprocess according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules].~~ |

~~Customer Identifier (2): MCN/GRC Pair~~

|  |  |  |
| --- | --- | --- |
| **~~Icore table.column~~** | **~~GDB ORGANIZATION\_IDENTIFIER column~~** | **~~Processing comment~~** |
| ~~-~~ | ~~id\_organization~~ | ~~ORGANIZATION.id~~ |
| ~~-~~ | ~~id\_change\_tracking~~ | ~~See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process]~~ |
| ~~-~~ | ~~id\_identifier\_type~~ | ~~Referencing IDENTIFIER\_TYPE ‘MCN\_GRC’~~ |

|  |  |  |
| --- | --- | --- |
| **~~Icore table.column~~** | **~~GDB ORGANIZATION\_IDENTIFIER\_VALUE column~~** | **~~Processing comment~~** |
| ~~-~~ | ~~id\_organization\_identifier~~ | ~~ORGANIZATION\_IDENTIFIER.id~~ |
| ~~-~~ | ~~id\_change\_tracking~~ | ~~ORGANIZATION\_IDENTIFIER.id\_change\_tracking~~ |
| ~~-~~ | ~~id\_identifier\_type~~ | ~~Referencing IDENTIFIER\_TYPE ‘MCN’~~ |
| ~~MCN\_GRC\_SOC\_SOURCE.mcn~~ | ~~Value~~ |  |

|  |  |  |
| --- | --- | --- |
| **~~Icore table.column~~** | **~~GDB ORGANIZATION\_IDENTIFIER\_VALUE column~~** | **~~Processing comment~~** |
| ~~-~~ | ~~id\_organization\_identifier~~ | ~~ORGANIZATION\_IDENTIFIER.id~~ |
| ~~-~~ | ~~id\_change\_tracking~~ | ~~ORGANIZATION\_IDENTIFIER.id\_change\_tracking~~ |
| ~~-~~ | ~~id\_identifier\_type~~ | ~~Referencing IDENTIFIER\_TYPE ‘GRC’~~ |
| ~~MCN\_GRC\_SOC\_SOURCE.grc~~ | ~~Value~~ |  |

~~If the current data record from “MCN\_GRC\_SOC” contains a non-null value in “POCITS\_MCN”, then process a second MCN/GRC pair into GDB using the values from above, but this time using the “POCITS\_MCN” value as the “MCN” value. This will result in a second “GDB.ORGANIZATION\_IDENTIFIER” record with a second MCN/GRC pair where only the “MCN” value is different (i.e. it is the “POCITS\_MCN” value).~~

**~~(3) Single MCN item data found in the record~~ (no longer supported; 2013-09-09; hc2164)**

~~This is the case if the column “MCN” has a value that is not equal to “<<NONE>>”, but where the “GRC” and “SOC” values are “<<NONE>>”.~~

~~In this case process the record from “MCN\_GRC\_SOC” in GDB as follows:~~

|  |  |  |
| --- | --- | --- |
| **~~Icore table.column~~** | **~~GDB ORGANIZATION column~~** | **~~Processing comment~~** |
| ~~-~~ | ~~id\_object\_type~~ | ~~Referencing OBJECT\_TYPE ‘ORGANIZATION’~~ |
| ~~-~~ | ~~id\_change\_tracking~~ | ~~See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process]~~ |
| ~~-~~ | ~~is\_read\_only~~ | ~~‘Y’~~ |
| ~~-~~ | ~~id\_organization\_type~~ | ~~Referencing ORGANIZATION\_TYPE ‘SERVICE\_SPECIFIC\_CUSTOMER\_REPRESENTATION’~~ |
| ~~MCN\_GRC\_SOC\_SOURCE.id~~ | ~~id [via Enterprise Key Translation]~~ | ~~Use SOURCE\_KEY.id into GDB~~ |
| ~~CUSTOMER.cust\_name~~ | ~~name~~ | ~~(via “MCN\_GRC\_SOC\_SOURCE.icore\_cust\_id”)~~ |
| ~~-~~ | ~~id\_address~~ | ~~NULL~~ |
| ~~-~~ | ~~is\_verification\_opted\_out~~ | ~~NULL~~ |
| ~~-~~ | ~~id\_organization\_unified~~ | ~~NULL, postprocess according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules].~~ |

~~Customer Identifier (3): MCN Value~~

|  |  |  |
| --- | --- | --- |
| **~~Icore table.column~~** | **~~GDB ORGANIZATION\_IDENTIFIER column~~** | **~~Processing comment~~** |
| ~~-~~ | ~~id\_organization~~ | ~~ORGANIZATION.id~~ |
| ~~-~~ | ~~id\_change\_tracking~~ | ~~See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process]~~ |
| ~~-~~ | ~~id\_identifier\_type~~ | ~~Referencing IDENTIFIER\_TYPE ‘MCN’~~ |

|  |  |  |
| --- | --- | --- |
| **~~Icore table.column~~** | **~~GDB ORGANIZATION\_IDENTIFIER\_VALUE column~~** | **~~Processing comment~~** |
| ~~-~~ | ~~id\_organization\_identifier~~ | ~~ORGANIZATION\_IDENTIFIER.id~~ |
| ~~-~~ | ~~id\_change\_tracking~~ | ~~ORGANIZATION\_IDENTIFIER.id\_change\_tracking~~ |
| ~~-~~ | ~~id\_identifier\_type~~ | ~~Referencing IDENTIFIER\_TYPE ‘MCN’~~ |
| ~~MCN\_GRC\_SOC\_SOURCE.mcn~~ | ~~Value~~ |  |

~~If the current data record from “MCN\_GRC\_SOC” contains a non-null value in “POCITS\_MCN”, then process a second MCN value into GDB using the values from above, but this time using the “POCITS\_MCN” value as the “MCN” value. This will result in a second “GDB.ORGANIZATION\_IDENTIFIER” record with a second MCN value where only the “MCN” value is different (i.e. it is the “POCITS\_MCN” value).~~

#### HLD-286282-US704600-GCP-FLOW-ICORE-100-C [Icore Data Flow into Organization BILLING ACCOUNT]

This requirement tries to load ICORE billing customer data into GDB where the identifiers are an MCN/GRC/SOC triplet, UB SUB ACCOUNT, or invoice account number.

This requirement is dependent on “HLD-254035-GCP-FLOW-ICORE-010 [Icore Data Flow: Maintain MCN/GRC/SOC Mapping]”, and “HLD-286282-US704600-GCP-FLOW-ICORE-020 [Icore Data Flow: Maintain Billing Account Mapping]” which need to be executed before this requirement can be executed.

Implement the following:

Process the following Icore records into the GDB organization object and into the Data Index (ref. Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

For each record pulled from “ICORE.GDB\_BILLING\_ACCOUNT” proceed as follows:

**(1) Full MCN/GRC/SOC triplet data found in the record**

This is the case if all three columns “ACCT\_1\_NB”, “ACCT\_3\_NB” and “ACCT\_2\_NB” have values that are not equal to “<<NONE>>” and ACCT\_TYPE\_CD = ‘AM’, or ‘XX’ (<Ticket-000000224475375>).

In this case process the record from “GDB\_BILLING\_ACCOUNT” in GDB as follows:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION column** | **Processing comment** |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘ORGANIZATION’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | id\_organization\_type | Referencing ORGANIZATION\_TYPE ‘BILLING\_ACCOUNT\_REPRESENTATION’ |
| GDB\_BILLING\_ACCOUNT.account\_key | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| - | Name |  |
| - | id\_address | NULL |
| - | is\_verification\_opted\_out | NULL |
| - | id\_organization\_unified | NULL, postprocess according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules]. |

Customer Identifier (1): Standard MCN/GRC/SOC Triplet  
(only set if data is available, i.e. a full triplet exists !)

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION\_IDENTIFIER column** | **Processing comment** |
| - | id\_organization | ORGANIZATION.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘MCN\_GRC\_SOC’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_organization\_identifier | ORGANIZATION\_IDENTIFIER.id |
| - | id\_change\_tracking | ORGANIZATION\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘MCN’ |
| GDB\_BILLING\_ACCOUNT.acct\_1\_nb | Value |  |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_organization\_identifier | ORGANIZATION\_IDENTIFIER.id |
| - | id\_change\_tracking | ORGANIZATION\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘GRC’ |
| GDB\_BILLING\_ACCOUNT.acct\_3\_nb | Value |  |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_organization\_identifier | ORGANIZATION\_IDENTIFIER.id |
| - | id\_change\_tracking | ORGANIZATION\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘SOC’ |
| GDB\_BILLING\_ACCOUNT.acct\_2\_nb | Value |  |

|  |  |  |
| --- | --- | --- |
| **~~USRP/GRID table.column~~** | **~~GDB ORGANIZATION\_IDENTIFER\_INFO column~~** | **~~Processing comment~~** |
| ~~-~~ | ~~id\_organizaiton\_identifier~~ | ~~ORGANIZATION\_IDENTIFIER.id~~ |
| ~~-~~ | ~~id\_change\_tracking~~ | ~~ORGANIZATION\_IDENTIFIER.id\_change\_tracking~~ |
|  | ~~ACCOUNT\_USAGE\_TYPE~~ |  |
|  | ~~BILLER\_CODE~~ |  |
|  | ~~BILLING\_CUST\_NAME~~ |  |
|  | ~~BILLING\_ACCOUNT\_NUMBER~~ |  |
|  | ~~ACCOUNT\_TYPE~~ |  |

**(2) MCN/GRC pair data found in the record**

This is the case if the columns “ACCT\_1\_NB” and “ACCT\_3\_NB” have values that are not equal to “<<NONE>>”, but where the “ACCT\_2\_NB” value is “<<NONE>>”, and ACCT\_TYPE\_CD = ‘AM’, or ‘XX’ (<Ticket-000000224475375>).

In this case process the record from GDB\_BILLING\_ACCOUNT in GDB as follows:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION column** | **Processing comment** |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘ORGANIZATION’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | id\_organization\_type | Referencing ORGANIZATION\_TYPE ‘BILLING\_ACCOUNT\_REPRESENTATION’ |
| GDB\_BILLING\_ACCOUNT.account\_key | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| - | name | NULL |
| - | id\_address | NULL |
| - | is\_verification\_opted\_out | NULL |
| - | id\_organization\_unified | NULL, postprocess according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules]. |

Customer Identifier (2): MCN/GRC Pair

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION\_IDENTIFIER column** | **Processing comment** |
| - | id\_organization | ORGANIZATION.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘MCN\_GRC’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_organization\_identifier | ORGANIZATION\_IDENTIFIER.id |
| - | id\_change\_tracking | ORGANIZATION\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘MCN’ |
| GDB\_BILLING\_ACCOUNT.acct\_1\_nb | Value |  |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_organization\_identifier | ORGANIZATION\_IDENTIFIER.id |
| - | id\_change\_tracking | ORGANIZATION\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘GRC’ |
| GDB\_BILLING\_ACCOUNT.acct\_3\_nb | Value |  |

**(3) Single MCN item data found in the record**

This is the case if the column “ACCT\_1\_NB” has a value that is not equal to “<<NONE>>”, but where the “ACCT\_3\_NB” and “ACCT\_2\_NB” values are “<<NONE>>”, and ACCT\_TYPE\_CD=’AM’ , or ‘XX’ (<Ticket-000000224475375>).

In this case process the record from “GDB\_BILLING\_ACCOUNT” in GDB as follows:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION column** | **Processing comment** |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘ORGANIZATION’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | id\_organization\_type | Referencing ORGANIZATION\_TYPE ‘BILLING\_ACCOUNT\_REPRESENTATION’ |
| GDB\_BILLING\_ACCOUNT.account\_key | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| - | name |  |
| - | id\_address | NULL |
| - | is\_verification\_opted\_out | NULL |
| - | id\_organization\_unified | NULL, postprocess according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules]. |

Customer Identifier (3): MCN Value

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION\_IDENTIFIER column** | **Processing comment** |
| - | id\_organization | ORGANIZATION.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘MCN’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_organization\_identifier | ORGANIZATION\_IDENTIFIER.id |
| - | id\_change\_tracking | ORGANIZATION\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘MCN’ |
| GDB\_BILLING\_ACCOUNT.acct\_1\_nb | Value |  |

**(4) UB SUB ACCOUNT data found in the record**

This is the case if the column “ACCT\_1\_NUMBER” has a value that is not equal to “<<NONE>>”, and ACCT\_TYPE\_CD=’AS’.

In this case process the record from “GDB\_BILLING\_ACCOUNT” in GDB as follows:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION column** | **Processing comment** |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘ORGANIZATION’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | id\_organization\_type | Referencing ORGANIZATION\_TYPE ‘SERVICE\_SPECIFIC\_CUSTOMER\_REPRESENTATION’ |
| GDB\_BILLING\_ACCOUNT.account\_key | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| - | Name |  |
| - | id\_address | NULL |
| - | is\_verification\_opted\_out | NULL |
| - | id\_organization\_unified | NULL, postprocess according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules]. |

Customer Identifier (3): UB ACCOUNT ID Value

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION\_IDENTIFIER column** | **Processing comment** |
| - | id\_organization | ORGANIZATION.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘UB\_ACCOUNT\_ID’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_organization\_identifier | ORGANIZATION\_IDENTIFIER.id |
| - | id\_change\_tracking | ORGANIZATION\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘UB\_ACCOUNT\_ID’ |
| GDB\_BILLING\_ACCOUNT.acct\_1\_nb | Value |  |

**(4) INVOICE ACCOUNT NUMBER data found in the record**

This is the case if the column “ACCT\_1\_NUMBER” has a value that is not equal to “<<NONE>>”, and ACCT\_TYPE\_CD=’I’.

In this case process the record from “GDB\_BILLING\_ACCOUNT” in GDB as follows:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION column** | **Processing comment** |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘ORGANIZATION’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | id\_organization\_type | Referencing ORGANIZATION\_TYPE ‘BILLING\_ACCOUNT\_REPRESENTATION’ |
| GDB\_BILLING\_ACCOUNT.account\_key | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| - | name |  |
| - | id\_address | NULL |
| - | is\_verification\_opted\_out | NULL |
| - | id\_organization\_unified | NULL, postprocess according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules]. |

Customer Identifier (3): BAN Value

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION\_IDENTIFIER column** | **Processing comment** |
| - | id\_organization | ORGANIZATION.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘BAN’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ORGANIZATION\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_organization\_identifier | ORGANIZATION\_IDENTIFIER.id |
| - | id\_change\_tracking | ORGANIZATION\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘BAN’ |
| GDB\_BILLING\_ACCOUNT.acct\_1\_nb | Value |  |

#### HLD-286282-US704600-GCP-FLOW-ICORE-100-D [Icore Data Flow into Organization to Organization]

This requirement tries to load association between inventory organization and billing organization into GDB.

This requirement is dependent on “HLD-254035-GCP-FLOW-ICORE-100-B [Icore Data Flow into Organization MCN\_GRC\_SOC\_SOURCE]”, and “HLD-286282-US704600-GCP-FLOW-ICORE-100-C [Icore Data Flow into Organization BILLING ACCOUNT]” which need to be executed before this requirement can be executed.

Implement the following:

Process the following Icore records into the GDB organization object and into the Data Index (ref. Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

1. ~~Inventory Organization (MCN/GRC/SOC) to Billing Organization (Invoice Account Number) -- ticket 223595795~~

~~select c.id, a.account\_key~~

~~from GDB\_BILLING\_ACCOUNT a, site2billing\_account b, icore.MCN\_GRC\_SOC\_SOURCE c, icore.MCN\_GRC\_SOC\_SITE\_IDS d~~

~~where a.account\_key =b.account\_key~~

~~and c.id = d.id\_m\_g\_s\_source~~

~~and b.icore\_site\_id = d.site\_id~~

~~and a.ACCT\_TYPE\_CD = 'I' and c.triplet\_type in (‘S’, ‘AS’)~~

|  |  |  |
| --- | --- | --- |
| **~~Icore table.column~~** | **~~GDB ASSOCIATION column~~** | **~~Processing comment~~** |
| ~~-~~ | ~~id\_change\_tracking~~ | ~~See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process]~~ |
| ~~-~~ | ~~is\_read\_only~~ | ~~‘Y’~~ |
| ~~-~~ | ~~Id\_role\_notation~~ | ~~NULL~~ |
| ~~-~~ | ~~Id\_association\_type~~ | ~~Referencing ASSOCIATION\_TYPE.id which represents:~~  ~~OBJECT\_TYPE\_WHAT ‘ORGANIZATION’, OBJECT\_TYPE\_TO ‘ORGANIZATION’, ROLE with FUNCTION\_TYPE ‘BILLED\_BY’ and FUNCTION\_ROLE NULL~~ |
| ~~MCN\_GRC\_SOC\_SOURCE.id~~ | ~~Id\_object\_what [via Enterprise Key Translation for MCN\_GRC\_SOC\_SOURCE.id]~~ | ~~Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id~~  ~~Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules] where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id~~ |
| ~~ACCOUNT\_KEY(same as in GDB\_BILLING\_ACCOUNT)~~ | ~~Id\_object\_to [via Enterprise Key Translation]~~ | ~~Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id~~  ~~Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules] where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id~~ |
| ~~MCN\_GRC\_SOC\_SOURCE.id~~ | ~~Id\_object\_what [via Enterprise Key Translation for MCN\_GRC\_SOC\_SOURCE.id]~~ | ~~Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id~~ |
| ~~-~~ | ~~Gdb\_internal\_flags~~ | ~~NULL~~ |

1. ~~Inventory Organization (MCN/GRC/SOC) to Billing Organization (MCN/GRC/SOC) ---ticket 223595795~~

~~select c.id, a.account\_key~~

~~from GDB\_BILLING\_ACCOUNT a, site2billing\_account b, icore.MCN\_GRC\_SOC\_SOURCE c, icore.MCN\_GRC\_SOC\_SITE\_IDS d~~

~~where a.account\_key =b.account\_key~~

~~and c.id = d.id\_m\_g\_s\_source~~

~~and b.icore\_site\_id = d.site\_id~~

~~and a.ACCT\_TYPE\_CD = 'AM' and c.triplet\_type in (‘S’, ‘AS’)~~

|  |  |  |
| --- | --- | --- |
| **~~Icore table.column~~** | **~~GDB ASSOCIATION column~~** | **~~Processing comment~~** |
| ~~-~~ | ~~id\_change\_tracking~~ | ~~See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process]~~ |
| ~~-~~ | ~~is\_read\_only~~ | ~~‘Y’~~ |
| ~~-~~ | ~~Id\_role\_notation~~ | ~~NULL~~ |
| ~~-~~ | ~~Id\_association\_type~~ | ~~Referencing ASSOCIATION\_TYPE.id which represents:~~  ~~OBJECT\_TYPE\_WHAT ‘ORGANIZATION’, OBJECT\_TYPE\_TO ‘ORGANIZATION’, ROLE with FUNCTION\_TYPE ‘BILLED\_BY’ and FUNCTION\_ROLE NULL~~ |
| ~~MCN\_GRC\_SOC\_SOURCE.id~~ | ~~Id\_object\_what [via Enterprise Key Translation for MCN\_GRC\_SOC\_SOURCE.id]~~ | ~~Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id~~  ~~Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules] where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id~~ |
| ~~ACCOUNT\_KEY(same as in GDB\_BILLING\_ACCOUNT)~~ | ~~Id\_object\_to [via Enterprise Key Translation]~~ | ~~Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id~~  ~~Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules] where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id~~ |
| ~~MCN\_GRC\_SOC\_SOURCE.id~~ | ~~Id\_source\_key [via Enterprise Key Translation]~~ | ~~Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id~~ |
| ~~-~~ | ~~Gdb\_internal\_flags~~ | ~~NULL~~ |

1. Billing Organization (MCN/GRC/SOC) to Billing Organization (Invoice Account Number) – ticket 223595795

select a.account\_key, c.account\_key

from GDB\_BILLING\_ACCOUNT a, site2billing\_account b, GDB\_BILLING\_ACCOUNT c, site2billing\_account d

where a.account\_key =b.account\_key

and c.account\_key =d.account\_key

and b.icore\_site\_id = d.icore\_site\_id

and a.ACCT\_TYPE\_CD = 'AM' and c.ACCT\_TYPE\_CD = 'I'

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘ORGANIZATION’, OBJECT\_TYPE\_TO ‘ORGANIZATION’, ROLE with FUNCTION\_TYPE ‘ROLLS\_UP\_TO’ and FUNCTION\_ROLE NULL |
| ACCOUNT\_KEY(same as in GDB\_BILLING\_ACCOUNT) when ACCT\_TYPE\_CD = ‘AM’ | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id  Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules] where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id |
| ACCOUNT\_KEY(same as in GDB\_BILLING\_ACCOUNT) when ACCT\_TYPE\_CD = ‘I’ | Id\_object\_to [via Enterprise Key Translation for MCN\_GRC\_SOC\_SOURCE.id] | Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id  Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules] where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id |
| ACCOUNT\_KEY(same as in GDB\_BILLING\_ACCOUNT) when ACCT\_TYPE\_CD = ‘AM’ | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id |
| - | Gdb\_internal\_flags | NULL |

1. Inventory Organization (UB Account Id) to Billing Organization (Invoice Account Number)

select a.account\_key, c.account\_key

from GDB\_BILLING\_ACCOUNT a, site2billing\_account b, GDB\_BILLING\_ACCOUNT c, site2billing\_account d

where a.account\_key =b.account\_key

and c.account\_key =d.account\_key

and b.icore\_site\_id = d.icore\_site\_id

and a.ACCT\_TYPE\_CD = 'AS' and c.ACCT\_TYPE\_CD = 'I'

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘ORGANIZATION’, OBJECT\_TYPE\_TO ‘ORGANIZATION’, ROLE with FUNCTION\_TYPE ‘BILLED\_BY’ and FUNCTION\_ROLE NULL |
| ACCOUNT\_KEY(same as in GDB\_BILLING\_ACCOUNT) when ACCT\_TYPE\_CD = ‘AS’ | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id  Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules] where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id |
| ACCOUNT\_KEY(same as in GDB\_BILLING\_ACCOUNT) when ACCT\_TYPE\_CD = ‘I’ | Id\_object\_to [via Enterprise Key Translation for MCN\_GRC\_SOC\_SOURCE.id] | Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id  Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules] where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id |
| ACCOUNT\_KEY(same as in GDB\_BILLING\_ACCOUNT) when ACCT\_TYPE\_CD = ‘AS’ | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id |
| - | Gdb\_internal\_flags | NULL |

#### <Defect 67889>HLD-Defect-67889-FIX-GCP-ONE-TIME-DATA-CLEANUP

The execution of this process requires the maintenance of a change tracking event into GDB.CHANGE\_TRACKING, CHANGE\_SYSTEM and CHANGE\_USER. The change tracking event for the service delivery data flow must use CHANGE\_SYSTEM.name = ’ICORE\_TO\_GDB’ and CHANGE\_USER.name = ’ICORE\_TO\_GDB’. It is sufficient to have one change tracking event for all affected records during a single execution of this process.

**Note: This is only a one time cleanup of address records**

UPDATE the following GDB.ADDRESS\_NOTATION columns to remove all “leading” and “trailing” spaces:

1. ADDRESS\_NOTATION.CITY
2. ADDRESS\_NOTATION.ADDRESS\_LINE1
3. ADDRESS\_NOTATION.POSTAL\_CODE

#### HLD-254035-GCP-FLOW-ICORE-101 [Icore Data Flow into Site]

Implement the following:

Process the following Icore records into the GDB site object and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

ICORE site address:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ADDRESS\_NOTATION column** | **Processing comment** |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘ADDRESS\_NOTATION’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | id\_address\_notation\_type | Referencing ADDRESS\_NOTATION\_TYPE ‘INVENTORY\_ADDRESS\_REPRESENTATION’ |
| PREMISE.loc\_id | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| PREMISE.prem\_country | Country\_code | *Try to make sure the country data is converted to 3-character ISO country code data* |
| PREMISE.prem\_state | Subdivision |  |
| PREMISE.prem\_city | <Defect 67889> Trim(City) |  |
| PREMISE.prem\_address | <Defect 67889> Trim(Address\_line1) |  |
| - | Address\_line2 | Set to NULL |
| - | Address\_line3 | Set to NULL |
| PREMISE.prem\_zip | <Defect 67889> Trim(Postal\_code) |  |
| SITE.clli | Clli |  |

Database traversals (for “SITE.clli”):

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| PREMISE.loc\_id | SITE.prem\_loc\_id |

**NOTE**: If more then one “site” record exists per “premise” record, then take the one where “site.clli” sorts lowest.

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOCATION\_NOTATION column** | **Processing comment** |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘LOCATION\_NOTATION’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| SITE.site\_id | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
|  | Building | Set to NULL |
| SITE.floor | Floor |  |
| SITE.room | Room | Remove trailing spaces if any. |
| SITE.prem\_loc\_id (=PREMISE.loc\_id) | Id\_address\_notation [via Enterprise Key Translation for PREMISE.loc\_id] | ADDRESS\_NOTATION.id |
| site.~~globalLocationID~~ global\_loc\_id <Defect 608684> | Global\_location\_id <302503> |  |

**<289037c-ICORE-010>**

**GDB SITE\_EXT**

*This project adds new columns to GDB SITE\_EXT. The layout of GDB SITE\_EXT is usually preceded by the GDB SITE, SITE\_IDENTIFIER and SITE\_IDENTIFIER\_VALUE tables [for logical flow purposes]. But, since they are already defined elsewhere in this HLD for <286278>, they will not be repeated here.*

~~Insert a new record or update an existing record if Site Id exists in GDB SITE table. Don’t overwrite any values with a blank/null.~~

~~A one time load/update of the GDB SITE table for all ICORE Site Id’s will be needed.~~

|  |  |  |
| --- | --- | --- |
| **~~Icore table.column~~** | **~~GDB SITE column~~** | **~~Processing comment~~** |
| ~~-~~ | ~~id\_object\_type~~ | ~~Referencing OBJECT\_TYPE ‘SITE’~~ |
| ~~-~~ | ~~id\_change\_tracking~~ | ~~See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process]~~ |
| ~~-~~ | ~~is\_read\_only~~ | ~~‘Y’~~ |
| ~~-~~ | ~~id\_site\_type~~ | ~~Referencing SITE\_TYPE ‘INVENTORY\_SITE\_REPRESENTATION’~~ |
| ~~SITE.site\_id~~ | ~~id [via Enterprise Key Translation]~~ | ~~Use SOURCE\_KEY.id into GDB~~ |
| ~~SITE.site\_id~~ | ~~Id\_location\_notation [via Enterprise Key Translation]~~ | ~~LOCATION\_NOTATION.id from ICORE Site address~~ |
| ~~SITE.prem\_loc\_id (=PREMISE.loc\_id)~~ | ~~Id\_address\_notation [via Enterprise Key Translation]~~ | ~~ADDRESS\_NOTATION.id from ICORE Site address~~ |

|  |  |  |
| --- | --- | --- |
| **~~Icore table.column~~** | **~~GDB SITE\_IDENTIFIER column~~** | **~~Processing comment~~** |
| ~~-~~ | ~~id\_site~~ | ~~SITE.id~~ |
| ~~-~~ | ~~id\_change\_tracking~~ | ~~See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process]~~ |
| ~~-~~ | ~~id\_identifier\_type~~ | ~~Referencing IDENTIFIER\_TYPE ‘SIMPLE\_SITE\_IDENTIFIER’~~ |

|  |  |  |
| --- | --- | --- |
| **~~Icore table.column~~** | **~~GDB SITE\_IDENTIFIER\_VALUE column~~** | **~~Processing comment~~** |
| ~~-~~ | ~~id\_site\_identifier~~ | ~~SITE\_IDENTIFIER.id~~ |
| ~~-~~ | ~~id\_change\_tracking~~ | ~~SITE\_IDENTIFIER.id\_change\_tracking~~ |
| ~~-~~ | ~~id\_identifier\_type~~ | ~~Referencing IDENTIFIER\_TYPE ‘SIMPLE\_SITE\_IDENTIFIER’~~ |
| ~~SITE.site\_id~~ | ~~Value~~ |  |

A one time load/update of the GDB SITE\_EXT table for all ICORE Site Id’s will be needed to load the new column data.~~(done at same time as for GDB SITE table above).~~

On subsequent loads, insert a new record in GDB SITE\_EXT if a new Site ID has been added to ICORE SITE. Update existing records in GDB SITE\_EXT table for any SITE\_EXT.ID\_SITE where there has been a change in the ICORE source data. ~~Don’t overwrite any values with a blank/null~~.

|  |  |  |
| --- | --- | --- |
| **ICORE table.column** | **GDB SITE\_EXT column** | **Processing comment** |
| - | id\_site | Same as ID from GDB SITE *[derived via Enterprise Key Translation on ICORE Site Id]* |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| SITE\_CONTACT.CONT\_PHONE | WTN | Select SITE\_CONTACT.CONT\_PHONE from SITE\_CONTACT, SITE where SITE\_CONTACT.SITE\_ID = SITE.SITE\_ID and SITE\_CONTACT.CONT\_PRIMARY='Y' and SITE.SITE\_ID = <input ICORE Site Id>  If NULL value is returned, then use existing logic to retrieve the WTN as defined by <289278>.  If result contains any non-standard telephone data (i.e. alpha chars, hexadecimal, etc), then set WTN to NULL. |
| SITE.E2E\_SITE\_KEY | ETE\_SITE\_KEY | See End-to-End Key Search and Generation [ Icore Data Flow ]. *Same as existing logic to retrieve the ETE\_SITE\_KEY as defined by <289278>.* |
| PREMISE.PREM\_CAT, SITE.COMPLEX\_CAT | SITE\_TYPE | Select PREMISE.PREM\_CAT from PREMISE, SITE where PREMISE.LOC\_ID = SITE.PREM\_LOC\_ID and PREMISE.PREM\_CAT <> ‘X’ and SITE.SITE\_ID = <input ICORE Site Id> else select SITE.COMPLEX\_CAT from SITE where SITE.SITE\_ID = <input ICORE Site Id> |
| SITE\_STATUS.SITE\_STAT | STATUS  ~~SITE\_STATUS~~ | Select SITE\_STATUS.SITE\_STAT from SITE\_STATUS, SITE where SITE\_STATUS.SITE\_STAT\_ID = SITE.SITE\_STAT\_ID and SITE.SITE\_ID = <input ICORE Site Id> |
| SITE.MOD\_DATE | STATUS\_DATE | Select SITE.MOD\_DATE from SITE where SITE.SITE\_ID = <input ICORE Site Id> |
| CUSTOMER.WORKCTR | RESPONSIBLE\_CENTER | Select CUSTOMER.WORKCTR from CUSTOMER, SITE where CUSTOMER.CUST\_ID = SITE.CUST\_ID and SITE.SITE\_ID = <input ICORE Site Id> |
| COS\_PROFILE.PROFILE\_NAME | SERVICE\_LEVEL | Select COS\_PROFILE.PROFILE\_NAME from COS\_PROFILE, SITE where COS\_PROFILE.PROFILE\_ID = SITE.INGRESS\_PROFILE\_ID and SITE.SITE\_ID = <input ICORE Site Id> |
|  | REGION\_FLAG <302503> | Set this to ‘OOR’, if the icore.service.serv\_name = ‘OPT-E-WAN’ and icore.service\_option.serv\_opt = ‘ASEoD’ |

*\*region, site name not available in ICORE*

*<302503>*

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SERVICE\_ASGMT.site\_id | SITE.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |

*</302503>*

**<End of 289037c-ICORE-010>**

<286278>

Create or update GDB.SITE’s id\_address\_notation\_1, id\_address\_notation\_2, id\_address\_notation\_3 if they are populated, and update them or create new address\_notation records individually. Don’t create new record if mandatory fields don’t have values, or update any existing values with a null/blank. For updates, only update listed columns except the primary key. Each data element take usrp’s value if it is populated, otherwise take grid’s value.

Lec Address:

|  |  |  |
| --- | --- | --- |
| **USRP/GRID table.column** | **GDB ADDRESS\_NOTATION column** | **Processing comment** |
| USRP.CUSTOMER\_LOCATION.cust\_location\_id  Or  GRID.ADDRESS.ADDRESS\_ID | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB  For insert only |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘ADDRESS\_NOTATION’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | id\_address\_notation\_type | Referencing ADDRESS\_NOTATION\_TYPE ‘INVENTORY\_ADDRESS\_REPRESENTATION’ |
| USRP.CUSTOMER\_LOCATION.loc\_country  See Appendix to translate numeric value to country name.  Or  usrp.customer\_location.iso\_country\_code if loc\_country is not populated.  Or  GRID.ADDRESS.COUNTRY\_CODE if populated, else GRID.ADDRESS.COUNTRY | Country\_code | select distinct a.cust\_location\_id, a.customer\_id, loc\_country, …  from usrp.port\_info a, usrp.customer\_location c  where a.icore\_site\_id = <SITE.site\_id>  and a.cust\_location\_id = c.cust\_location\_id  or from grid if value is not populated from usrp  select b.\*  from grid.port p, grid.site a, grid.address b  where a.lec\_address\_id = b.address\_id and a.ete\_site\_key =p.ete\_site\_key  and p.icore\_site\_id = <SITE.site\_id>  Try to make sure the country data is converted to 3-character ISO country code data (the country code data in the BVOIP CSI database is in 2-character ISO format) |
| USRP.CUSTOMER\_LOCATION.loc\_state  or  GRID.ADDRESS.STATE | Subdivision |  |
| USRP.CUSTOMER\_LOCATION.loc\_city  or  GRID.ADDRESS.CITY | City |  |
| USRP.CUSTOMER\_LOCATION.loc\_str1  or  GRID.ADDRESS.STREET\_ADDRESS\_1 | Address\_line1 |  |
| USRP.CUSTOMER\_LOCATION.loc\_str2  or  GRID.ADDRESS.STREET\_ADDRESS\_2 | Address\_line2 |  |
| USRP.CUSTOMER\_LOCATION.loc\_zip  or  GRID.ADDRESS.ZIP\_CODE | Postal\_code |  |
|  | ADDRESS\_TYPE | ‘LEC’ |
| GRID.ADDRESS.TIME\_ZONE | TIME\_ZONE |  |
| GRID.ADDRESS.ADDRESS\_ID | SOURCE\_ADDRESS\_ID |  |

|  |  |  |
| --- | --- | --- |
| **GRID table.column** | **GDB LOCATION\_NOTATION column** | **Processing comment** |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘LOCATION\_NOTATION’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| USRP.CUSTOMER\_LOCATION.cust\_location\_id  Or  GRID.ADDRESS.ADDRESS\_ID | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB  For insert only |
| USRP.CUSTOMER\_LOCATION.locatioin\_bldg  or  GRID.ADDRESS.BUILDING | Building |  |
| USRP.CUSTOMER\_LOCATION.loc\_floor  or  GRID.ADDRESS.FLOOR | Floor |  |
| USRP.CUSTOMER\_LOCATION.loc\_room  or  GRID.ADDRESS.ROOM | Room |  |
| USRP.CUSTOMER\_LOCATION.loc\_suite  or  GRID.ADDRESS.SUITE | Suite |  |
| - | Id\_address\_notation [via Enterprise Key Translation] | ADDRESS\_NOTATION.id |

CPE SHIP TO Address:

AVPN doesn’t have cpe ShipTo address.

Billing Address:

|  |  |  |
| --- | --- | --- |
| **GRID table.column** | **GDB ADDRESS\_NOTATION column** | **Processing comment** |
| USRP.CUST\_BILLING\_ACCT’s mcn +grc + soc  Or  GRID.ADDRESS.ADDRESS\_ID | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB  For insert only |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘ADDRESS\_NOTATION’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | id\_address\_notation\_type | Referencing ADDRESS\_NOTATION\_TYPE ‘INVENTORY\_ADDRESS\_REPRESENTATION’ |
| USRP.CUST\_BILLING\_ACCT.state  or  GRID.ADDRESS.STATE | Subdivision |  |
| USRP.CUSTOMER.bill\_country  See Appendix to translate numeric value to country name.  Or  GRID.ADDRESS.COUNTRY\_CODE if populated, else GRID.ADDRESS.COUNTRY | Country\_code | select distinct a.bill\_country, b.\*  from usrp.customer a, usrp.cust\_billing\_acct b  where a.customer\_id = <customer\_id from ‘Lec Address’ query result>  and a.mcn= b.mcn and a.grc=b.grc  and a.soc = b.soc  or from grid if value is not populated from usrp  select b.\*  from grid.port p, grid.site a, grid.address b  where a.cpe\_ship\_to\_address\_id = b.address\_id and a.ete\_site\_key =p.ete\_site\_key  and p.icore\_site\_id = <SITE.site\_id>  Try to make sure the country data is converted to 3-character ISO country code data (the country code data in the BVOIP CSI database is in 2-character ISO format) |
| USRP.CUST\_BILLING\_ACCT.city  or  GRID.ADDRESS.CITY | City |  |
| USRP.CUST\_BILLING\_ACCT.billing\_address1  or  GRID.ADDRESS.STREET\_ADDRESS\_1 | Address\_line1 |  |
| USRP.CUST\_BILLING\_ACCT.billing\_address2  or  GRID.ADDRESS.STREET\_ADDRESS\_2 | Address\_line2 |  |
| USRP.CUST\_BILLING\_ACCT.zip  or  GRID.ADDRESS.ZIP\_CODE | Postal\_code |  |
|  | ADDRESS\_TYPE | ‘BILLING’ |
| GRID.ADDRESS.TIME\_ZONE | TIME\_ZONE |  |
| GRID.ADDRESS.ADDRESS\_ID | SOURCE\_ADDRESS\_ID |  |

</286278>

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB SITE column** | **Processing comment** |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘SITE’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | id\_site\_type | Referencing SITE\_TYPE ‘INVENTORY\_SITE\_REPRESENTATION’ |
| SITE.site\_id | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| SITE.site\_id | Id\_location\_notation [via Enterprise Key Translation] | LOCATION\_NOTATION.id from ICORE Site address |
| SITE.prem\_loc\_id (=PREMISE.loc\_id) | Id\_address\_notation [via Enterprise Key Translation] | ADDRESS\_NOTATION.id from ICORE Site address |
| <286278> |  |  |
|  | id\_address\_notation\_1 | ADDRESS\_NOTATION.id from Lec Address |
|  | id\_address\_notation\_2 | ADDRESS\_NOTATION.id from Biling Address |
|  | id\_address\_notation\_3 | AVPN doesn’t have CPE ship to address |
| </286278> |  |  |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB SITE\_IDENTIFIER column** | **Processing comment** |
| - | id\_site | SITE.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘SIMPLE\_SITE\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB SITE\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_site\_identifier | SITE\_IDENTIFIER.id |
| - | id\_change\_tracking | SITE\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘SIMPLE\_SITE\_IDENTIFIER’ |
| SITE.site\_id | Value |  |

<286278> <286284-US600815-US600161>

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB SITE\_IDENTIFIER column** | **Processing comment** |
| - | id\_site | SITE.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘E2E\_SITE\_KEY\_SITE\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB SITE\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_site\_identifier | SITE\_IDENTIFIER.id |
| - | id\_change\_tracking | SITE\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘E2E\_SITE\_KEY\_SITE\_IDENTIFIER’ |
|  | Value | <286284-US600815-US600161>  See End-to-End Key Search and Generation [ Icore Data Flow ]  Note: pass NEW\_KEY\_IND with ‘N’, don’t create new key if didn’t find one.  ~~select e2e\_site\_key, ‘LPP’ as ete\_key\_source~~  ~~from icore.site~~  ~~where e2e\_site\_key is not null~~  ~~and site\_id = <icore.SITE.site\_id>~~  ~~or~~  ~~select distinct ete\_site\_key, ‘USRP’ as ete\_key\_source~~  ~~from usrp.logical\_port~~  ~~where ete\_site\_key is not null~~  ~~and icore\_site\_id = <icore.SITE.site\_id>~~  ~~and ete\_site\_key is not null~~  ~~and valid\_ind = ‘Y’~~  ~~or~~  ~~select distinct ete\_site\_key, , ‘IEOD’ as ete\_key\_source~~  ~~from grid.port~~  ~~where ete\_site\_key is not null~~  ~~and icore\_site\_id = <icore.SITE.site\_id>~~ |

Insert a new record or update an existing record if GDB.SITE.ID exists in GDB.SITE\_EXT table. Don’t overwrite any values with a blank/null.

|  |  |  |
| --- | --- | --- |
| **GRID table.column** | **GDB SITE\_EXT column** | **Processing comment** |
| - | id\_site | GDB.SITE.ID (only for insert) |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| trim(usrp.customer\_location.working\_phone)  GRID.SITE.WTN  <281578b-US870920> | WTN | Insert/Update  select distinct trim(c.working\_phone),  trim(c.no\_working\_phone\_check\_ind),  trim(c.new\_const\_ind)  from usrp.logical\_port a, usrp.logical\_port\_asgmt b, usrp.customer\_location c  where icore\_site\_id = <SITE.site\_id>  and a.log\_port\_id = b.log\_port\_id  and b.cust\_location\_id = c.cust\_location\_id |
| trim(usrp.customer\_location.no\_working\_phone\_check\_ind)  GRID.SITE.UNDER\_CONSTRUCTION\_IND  <281578b-US870920> | UNDER\_CONSTRUCTION\_IND | Insert/Update  See above |
| Trim(usrp.customer\_location.new\_const\_ind)  GRID.SITE.NO\_ON\_SITE\_TN <281578b-US870920> | NO\_ON\_SITE\_TN | Insert/Update  See above |
| - | ETE\_SITE\_KEY | <286284-US600815-US600161>  See End-to-End Key Search and Generation [ Icore Data Flow ]  Note: pass NEW\_KEY\_IND with ‘N’, don’t create new key if didn’t find one.  ~~<270198g>~~  ~~if there is any other sites with the same ete\_site\_key on SITE\_EXT.ete\_site\_key, wipe those values in site\_ext to blank.~~ |
|  | ETE\_KEY\_SOURCE | <286284-US600815-US600161>  See End-to-End Key Search and Generation [ Icore Data Flow ]  Note: pass NEW\_KEY\_IND with ‘N’, don’t create new key if didn’t find one. |
| <Begin 281578b-US870920> |  |  |
| USRP.Customer\_location.local\_company\_name  Grid.Site.cust\_business\_name | CUSTOMER\_BUSINESS\_NAME |  |
| Usrp.Customer\_location.loc\_aft  Grid.Address.address\_format\_type | ADDRESS\_FORMAT\_TYPE |  |
| Usrp.customer\_location.loc\_str\_num  Grid.Address.street\_number | STREET\_NUMBER |  |
| Usrp.customer\_location.loc\_str\_name  GRID. ADDRESS.STREET\_NAME | STREET\_NAME |  |
| Usrp.customer\_location.loc\_str\_type  Grid. ADDRESS.STREET\_TYPE | STREET\_TYPE |  |
| Usrp.customer\_location.loc\_str\_num\_prefix  Grid. ADDRESS.STREET\_NUMBER\_PREFIX | STREET\_NUMBER\_PREFIX |  |
| Usrp.customer\_location.loc\_str\_num\_suffix  Grid. ADDRESS.STREET\_NUMBER\_SUFFIX | STREET\_NUMBER\_SUFFIX |  |
| Usrp.customer\_location.loc\_str\_directional  Grid.Address.Street\_Direction | STREET\_DIRECTION |  |
| Usrp.customer\_location.loc\_str\_post\_directional  Grid.Address.Street\_Direction\_Suffix | STREET\_TRAILING\_DIRECTION |  |
| Usrp.customer\_location.loc\_room  Grid.Address.room | ROOM |  |
| Usrp.Customer\_location.loc\_floor  Grid.Address.floor | FLOOR |  |
| Usrp.Customer\_location.location\_bldg  Grid.Address.Building | BUILDING |  |
| Usrp.Customer\_location.loc\_suite  Grid.address.suite | SUITE |  |
| Usrp.Customer\_location.ld1  Grid.Site.Loc\_Designator1 | LOCATION\_DESIGNATOR1 |  |
| Usrp.Customer\_location.~~ldv1~~ lv1  Grid.Site.Loc\_Designator\_Value1 | LOCATION\_DESIGNATOR\_VALUE1 |  |
| Usrp.Customer\_location.ld2  Grid.Site.Loc\_Designator2 | LOCATION\_DESIGNATOR2 |  |
| Usrp.Customer\_location.~~ldv2~~ lv2  Grid.Site.Loc\_Designator\_Value2 | LOCATION\_DESIGNATOR\_VALUE2 |  |
| Usrp.Customer\_location.ld3  Grid.Site.Loc\_Designator3 | LOCATION\_DESIGNATOR3 |  |
| Usrp.Customer\_location.~~ldv3~~ lv3  Grid.Site.Loc\_Designator\_Value3 | LOCATION\_DESIGNATOR\_VALUE3 |  |
| Usrp.Customer\_location.npa\_nxx  Grid.Site.Loc\_Npa.Nxx | LOCATION\_NPA\_NXX |  |
| Usrp.Ckt\_loc.aprn  Grid.Site.Access\_Project\_Number | ACCESS\_PROJECT\_NUMBER | ~~Please find the generic query for the elements in USRP.CKT\_LOC table.~~  ~~SELECT \*~~  ~~FROM usrp.circuit uc, usrp.ckt\_loc ucktl, usrp.customer\_location ucl, usrp.logical\_port ulp, usrp.port\_acty\_info pai~~  ~~WHERE uc.circuit\_id = ucktl.circuit\_id~~  ~~AND upai.ckt\_id = uc.circuit\_id~~  ~~AND upai.log\_port\_id = ulp.log\_port\_id~~  ~~AND upai.cust\_location\_id = ucl.cust\_location\_id~~  ~~AND icore\_site\_id = <site.site\_id>~~  Please find the generic query for the elements in USRP.SALES\_ORDER table.  SELECT \*  FROM usrp.circuit uc, usrp.ckt\_loc ucktl, usrp.customer\_location ucl, usrp.logical\_port ulp, usrp.port\_acty\_info upai , usrp.sales\_order uso, usrp.activity ua  WHERE uc.circuit\_id = ucktl.circuit\_id  AND upai.ckt\_id = uc.circuit\_id  AND upai.log\_port\_id = ulp.log\_port\_id  AND upai.cust\_location\_id = ucl.cust\_location\_id  AND upai.log\_port\_id = ua.ntwk\_entity\_id  AND ua.so\_id = uso.sales\_order\_id  AND ua.ntwk\_entity\_type = 1  AND ua.valid\_ind = 'Y'  AND icore\_site\_id = <site.site\_id>  or  Please find the generic query for the elements in GRID.SITE, GRID.ADDRESS tables  ~~Select \*~~  ~~FROM Grid.Site gs, GRID.ADDRESS ga WHERE Ga.Address\_Id = Gs.Lec\_Address\_Id AND gs.ete\_site\_key = i\_site\_key;~~  SELECT \*  FROM  GRID.SITE S  LEFT OUTER JOIN  GRID.ADDRESS A  ON  A.ADDRESS\_ID=S.LEC\_ADDRESS\_ID  WHERE S.ETE\_SITE\_KEY=I\_SITE\_KEY; |
| GRID.SITE.ADDR\_VALIDATION\_REQUEST\_ID | ADDRESS\_VALIDATION\_REQUEST\_ID |  |
| GRID.SITE.VALIDATION\_STATUS | ADDRESS\_VALIDATION\_STATUS |  |
| GRID.SITE.REMARK\_ADDR\_VALIDATION | REMARKS\_TO\_FROM\_ADDR\_VAL~~IDATION~~ |  |
| GRID.SITE.ACCESS\_PROVIDER | ACCESS\_PROVIDER |  |
| Usrp.Ckt\_loc.cust\_spec\_access\_vendor  Grid. SITE.ACCESS\_PROVIDER\_CODE | ACCESS\_PROVIDER\_CODE |  |
| Usrp.CKT\_LOC.AAV\_NPA\_NXX  Grid. SITE.ALT\_ACCESS\_PROVIDER | ALTERNATE\_ACCESS\_PROVIDER |  |
| Usrp.Ckt\_loc.~~case\_ind~~ casc\_ind  Grid. SITE.APPLY\_CASE\_IND | APPLY\_CASE\_FLAG |  |
| Usrp.Customer\_location.cust\_prem\_clli  Grid. SITE.CP\_CLLI | CP\_CLLI |  |
| Usrp.Customer\_location.site\_id  Grid. SITE.GCP\_SITE\_ID | SITE\_ID |  |
| Usrp.Customer\_location.site\_name  Grid. SITE.GCP\_SITE\_NAME | SITE\_NAME |  |
| GRID.ADDRESS.TIME\_ZONE | TIME\_ZONE |  |
| Usrp.Ckt\_Loc.special\_on\_net\_offer\_ind  Grid. SITE.SPEC\_ONNET\_OFFER\_IND | SPL\_ON\_NET\_OFFER\_PRICING\_FLAG |  |
| Usrp.Ckt\_loc.special\_on\_net\_offer\_token  Grid. SITE.ONNET\_TOKEN\_ID | On\_net\_token\_id |  |
| Usrp.Ckt\_loc.special\_on\_net\_offer\_remark  Grid. SITE.SPEC\_ONNET\_REMARK | Spl\_on\_net\_offer\_remarks |  |
| GRID.ADDRESS.CUSTOMER\_LOCATION\_NAME | CUSTOMER\_LOCATION\_NAME |  |
| GRID.ADDRESS.SITE\_ALIAS\_NAME | SITE\_ALIAS\_NAME |  |
| GRID.ADDRESS.ROUTE | ROUTE |  |
| GRID.ADDRESS.PO\_BOX | PO\_BOX |  |
| GRID.ADDRESS.GEO\_CODE | GEO\_CODE |  |
| USRP.Logical\_port.site\_bundle\_type  Grid. SITE.SITE\_BUNDLE\_TYPE | SITE\_BUNDLE\_TYPE |  |
| USRP.Logical\_port.primary\_bundle\_desc  Grid. SITE.PRI\_BUNDLE\_DESC | PRIMARY\_BUNDLE\_DESCRIPTION |  |
| USRP.Logical\_port.secondary\_bundle\_desc  Grid. SITE.SEC\_BUNDLE\_DESC | SECONDARY\_BUNDLE\_DESCRIPTION |  |
| USRP.Logical\_port.tertiary\_bundle\_desc  Grid. SITE.TER\_BUNDLE\_DESC | TERTIARY\_BUNDLE\_DESCRIPTION |  |
| USRP.Logical\_port.site\_bundle\_id  Grid. SITE.SITE\_BUNDLE\_ID | SITE\_BUNDLE\_ID |  |
| GRID.SITE.LOCATION\_ALIAS | LOCATION\_ALIAS |  |
| USRP.Sales\_order.express\_product  GRID.SITE.EXPRESS\_PRODUCT | EXPRESS\_PRODUCT |  |
| USRP.Sales\_order.express\_offer\_type  GRID.SITE.EXPRESS\_OFFER | EXPRESS\_OFFER |  |
| USRP.Sales\_order.express\_sla  GRID.SITE.EXPRESS\_SLA | EXPRESS\_SLA |  |
| GRID.SITE.MARKET\_OFFER\_NAME | MARKETING\_OFFER\_NAME |  |
| MAX(SALES\_ORDER.management\_option)  GRID.SITE.MANAGEMENT\_OPTION | Management\_Option |  |
| GRID.SITE.SERVICE\_FLAVOR | Service\_Flavor |  |
| GRID.SITE.RESILIENCY\_OPTION | Resiliency\_option |  |
| <END 281578b-US870920> |  |  |

</286278>

#### HLD-254035-GCP-FLOW-ICORE-102-A [Icore Data Flow into Asset CUST\_ACCESS (“Access Circuit”)]

The following section is about “Access Circuit” assets.

Implement the following:

Process the following Icore records into the GDB asset object and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

*NOTE: Only include records where “ICORE.CUST\_ACCESS.acc\_ckt” is not null and is not empty !*

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET column** | **Processing comment** |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘ASSET’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| CUST\_ACCESS.site\_id | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| - | id\_asset\_type | Referencing ASSET\_TYPE ‘ACCESS\_CIRCUIT’ |

**NOTE:**

For ICORE circuit id values it is not probable that they are already in CLCI format. Nevertheless it is best to check whether a circuit id can be converted to the CLCI format, and if yes, then check whether the CLCI format value is identical to the “raw” value as retrieved directly from ICORE. If the latter is the case, then no “raw” format identifier data set shall be created !

Asset Identifier (1): “Raw” Circuit ID value

(Only to be created if the “raw” value and the “CLCI” value are different; see “NOTE” above)

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘CIRCUIT\_ID\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_asset\_identifier | ASSET\_IDENTIFIER.id |
| - | id\_change\_tracking | ASSET\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘CIRCUIT\_ID\_IDENTIFIER’ |
| CUST\_ACCESS.acc\_ckt | Value |  |

Asset Identifier (2): “Standard US Domestic Circuit ID Format” Circuit ID value  
*(only to be created if a conversion is successful; see “Logic for converting a US Domestic circuit id value from ICORE into the standard US Domestic Circuit ID Format”)*

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘STANDARD\_FORMAT\_CIRCUIT\_ID\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_asset\_identifier | ASSET\_IDENTIFIER.id |
| - | id\_change\_tracking | ASSET\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘STANDARD\_FORMAT\_CIRCUIT\_ID\_IDENTIFIER’ |
| CUST\_ACCESS.acc\_ckt converted to a “Standard US Domestic Circuit ID Format” Circuit ID value (see “Logic for converting a US Domestic circuit id value from ICORE into the standard US Domestic Circuit ID Format”) | Value |  |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_EXT\_ACCESS\_CIRCUIT column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| (1) Use “CUST\_ACCESS.acc\_ckt” if no CLCI-formatted value could be converted; (2) Use the CLCI-formatted value of “CUST\_ACCESS.acc\_ckt” if a CLCI-formatted value could be converted; | circuit\_id\_value |  |
|  | id\_circuit\_id\_identifier\_type | (1) Referencing IDENTIFIER\_TYPE ‘CIRCUIT\_ID\_IDENTIFIER’ if no CLCI-formatted value could be converted; (2) Referencing IDENTIFIER\_TYPE ‘STANDARD\_FORMAT\_CIRCUIT\_ID\_IDENTIFIER’ if a CLCI-formatted value could be converted; |
| - | id\_circuit\_service\_codes  (used for “description” data) | <to be derived; see “Logic for deriving the access circuit “description” and “speed” values”>;  NOTE:  Use “GDB.CIRCUIT\_SERVICE\_CODES.id” where “GDB.CIRCUIT\_SERVICE\_CODES.SERVICE\_CODE” matches the service code as derived from the currently processed circuit id value (in CLCI format; it does not work for “raw”-only circuit id values);  Use NULL if no match could be found |
| - | speed | 231405909:  Icore.site.access\_speed,  If it is not populated, follow the below:  ~~Defect 191509: use the logic under “Logic for pulling the access circuit “speed” values from “port\_asgmt”” to set the value.~~  ~~If no value is populated form “port\_asgmt” then follow the below:~~  <to be derived; see “Logic for deriving the access circuit “description” and “speed” values”>;  NOTE:  Use “GDB.CIRCUIT\_SERVICE\_CODES.speed” where “GDB.CIRCUIT\_SERVICE\_CODES.SERVICE\_CODE” matches the service code as derived from the currently processed circuit id value (in CLCI format; it does not work for “raw”-only circuit id values);  ~~<BEGIN 254035c.DEFECT-55173>~~  ~~If no match could be found, then use the logic under “Logic for pulling the access circuit “speed” values from “port\_asgmt”” to set the value.~~  ~~<END 254035c.DEFECT-55173>~~ |
| CUST\_ACCESS.mpc | parent\_channel |  |
| CUST\_ACCESS.ia\_ckt | parent\_circuit\_id\_value |  |
| <279006> SITE.BROADBAND\_IND | Is\_broadband | Use defined default value in the table definition when site.broadband\_ind is null |
| <279006> SITE.UPSTREAM\_ASYM\_SPEED | Upload\_speed |  |
| <286278> |  |  |
|  | ete\_access\_key | <286284-US600815-US600161>  See End-to-End Key Search and Generation [ Icore Data Flow ]  ~~select distinct ete\_access\_key, ‘USRP’ as ete\_key\_source~~  ~~from usrp.logical\_port~~  ~~where valid\_ind = 'Y'~~  ~~and icore\_site\_id = <site.id>~~  ~~or~~  ~~select distinct ete\_access\_key, ‘IEOD’ as ete\_key\_source~~  ~~from grid.circuit~~  ~~where clci (after removed all special chars) = <circuit\_id\_value after removed all special chars>~~  ~~Note: removing all special characters for a circuit id means to remove all special chars but keeping all alphabets and numbers, for example, for /KRGP/12345/ATI,keep only KRGP12345ATI~~  ~~if there is any other asset\_ext\_access\_circuit records with the same ete\_access\_key but associated with some CE port with a different icore site id:~~  ~~PHY\_PORT (with a different icore site id value, port type referencing to ‘CE’)-> ( USED BY/NULL ) > ASSET (access circuit asset with the same ete\_access\_key, but with a different asset.id)~~  ~~Wipe out those asset\_ext\_access\_circuit.ete\_access\_key to blank/null.~~ |
|  | ete\_key\_source | <286284-US600815-US600161>  See End-to-End Key Search and Generation [ Icore Data Flow ]  ~~See above. ete\_key\_source should be blank/null when ete\_access\_key is blank/null.~~ |
| </286278> |  |  |
| <288655a>  GPS.AT\_CIRCUIT\_BASE.AT3RDPARTYTYPE  Or  SIDBOR.CIRCUIT.thirdpartytype | Third\_party\_flag | Set third\_party\_flag = ‘Y’, if the retrieved value from one of the steps below = ‘LOA’ or ‘COR’;  Else set third\_party\_flag = ‘N’  Match with GPS.AT\_CIRCUIT\_BASE.atcircuitname = circuit\_id\_value  Or  Match with SIDBOR.CIRCUIT.telco\_ckt\_id = circuit\_id\_value  (Use ‘Access Circuit Matching logic’ section from below for the above comparison) |
| IPFR.DHV\_INDICATOR <292589-US847769> | DHV\_FLAG | Note: Need to check all the PVCs associated to the site and if atleast one PVC has the DHV\_INDICATOR set to ‘Y’, then, the value in GDB should also be set to ‘Y’.  For example, if four PVCs are returned for a particular lsite\_id or rsite\_id, and 2 PVCs are having the DHV\_INDICATOR=’Y’ and 2 PVCs are having the DHV\_INDICATOR=’N’. Since there is atleast one PVC with DHV\_INDICATOR =’Y’, the value in GDB should also be set to ‘Y’. |
|  | Region\_flag <302503> | Set this to ‘OOR’, if icore.service.serv\_name = ‘OPT-E-WAN’ and icore.service\_option.serv\_opt = ‘ASEoD’ |

<302503>

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_ACCESS\_CIRCUIT\_DETAILS column** | **Processing comment** |
| - | asset\_id | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| site.LECID | LEC\_CIRCUIT\_ID | Referencing service\_option.serv\_type = ‘ASEoD’ |

</302503>

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| CUST\_ACCESS.site\_id | PORT\_ASGMT.site\_id |
| <279006> CUST\_ACCESS.site\_id | SITE.site\_id |
| CUST\_ACCESS\_CKT.SITE\_ID | PVC.PVC\_LSITE\_ID or PVC.RSITE\_ID <292589> |
| PVC.PVC\_ID | IPFR.PVC\_ID <292589> |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |
| CUST\_ACCESS | ACC\_CKT | (must not be empty or null) |

**Data Blocking (1)**

Do not process asset records that are linked to a GDB.SERVICE\_TYPE having a record in GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM for Icore:

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| CUST\_ACCESS.site\_id | SITE.site\_id |
| SERVICE\_ASGMT.site\_id | SITE.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |
| GDB.SERVICE\_TYPE\_NOTATION.service\_name | SERVICE.serv\_name |
| GDB.SERVICE\_TYPE\_NOTATION.id\_service\_type | GDB.SERVICE\_TYPE.id |
| GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM.id\_service\_type | GDB.SERVICE\_TYPE.id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| SERVICE\_TYPE\_BLOCKED\_SYSTEM | BLOCKED\_SYSTEM | 13609 |

*Note:*

*The BLOCKED\_SYSTEM constraint value is the MOTS-ID of the system.*

**Data Blocking (2)**

Do not process asset records where the special logic for blocking INSTAR items hold.

The “<input site id variable (dep. on current scenario)>” here is “CUST\_ACCESS.site\_id”.

See “Logic for excluding INSTAR items”.

##### Access Circuit Matching logic

<288655a>

1. First, take the GPS.AT\_CIRCUIT\_BASE.atcircuitname or SIDBOR.CIRCUIT.telco\_ckt\_id, remove all special characters except letters and digits, for example, ‘DHEC/231081/808/ATI’ should be ‘DHEC231081808ATI’ after removing all special characters, matches with GDB.ASSET\_EXT\_ACCESS\_CIRCUIT.circuit\_id\_value (after removing all special characters too). A function-based index might be needed on GDB.ASSET\_EXT\_ACCESS\_CIRCUIT.circuit\_id\_value. The Oracle replace function might be good enough for this purpose.
2. If there is any match found, use the first record.
3. If there is no match found, convert the GPS.AT\_CIRCUIT\_BASE.atcircuitname or SIDBOR.CIRCUIT.telco\_ckt\_id, to Standard Domestic Circuit ID Format as described in the section (Logic for converting a US Domestic circuit id value from ICORE into the standard US Domestic Circuit ID Format (CLCI)), try to match with GDB.ASSET\_EXT\_ACCESS\_CIRCUIT.circuit\_id\_value again (from step 1), if there is exactly one matched, use this ID\_ASSET as matched GDB.ASSET.ID.

##### Logic for excluding INSTAR items

Items from the ICORE database where the service is “MIS” (or a flavour of “MIS”) need to be excluded, as INSTAR is the “master” database here, but they cannot easily be identified using the standard “Blocked System” approach, so a specific approach is needed which is given below.

Items where the following “Database Traversal” and “Database Constraints” hold true need to be excluded/blocked from loading:

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| <input site id variable (dep. on current scenario)> | SITE.site\_id |
| SERVICE\_ASGMT.site\_id | SITE.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| SERVICE\_OPTION | SERV\_OPT | In (“GMIS”, “MIS - Basic”, “MIS - Plus”)  *(if needed, truncate spaces from db values !)* |

<BEGIN DEFECT-FIX.INSTAR-MIS-EXCLUSION.2016-03-15>

The following traversals and constraints need to be added to ones from above

via a logical “OR” for both the “traversal” and the “constraints”.

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| <input site id variable (dep. on current scenario)> | SITE.site\_id |
| SERVICE\_ASGMT.site\_id | SITE.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| SERVICE | SERV\_NAME | ETHERNET |

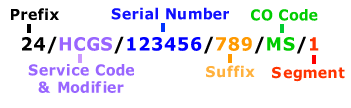
<END DEFECT-FIX.INSTAR-MIS-EXCLUSION.2016-03-15>

##### US Domestic Circuit ID Format

The following is an excerpt from the “eStrategy API.docx” document (owned by Russ Soph).

BDMap interface uses the slash delimited CLCI format for all US Domestic circuit IDs.

**Serial Number Format**



* **Prefix:** 1-2 alphanumeric characters. This is an optional field. If omitted the leading slash is also omitted.
* **Service Code & Modifier:** 2-4 alphabetic characters (usually 4). This is a required field.
* **Serial Number:** 1-6 digits. This is a required field.
* **Suffix:** 3 character suffix to the serial number may be required (rarely used). If omitted the slashes are still required.
* **CO (Company) Code:** 2-4 alphabetic characters. This is a required field. AT&T typically uses the following ATI (legacy T) or ATX (new company):
* **Segment:** 1-3 alphanumeric characters. optional for non-multi-point circuits. multi-point segments map to Circuit End Location, e.g. CLK1 = A, CLK2 = B. If omitted the trailing slash is also omitted.

MOW Circuit IDs do not conform to any standard format and must be passed *as-is* in a freeform string format.

(See also “Appendix: Circuit Id Formats”)

##### Logic for converting a US Domestic circuit id value from ICORE into the standard US Domestic Circuit ID Format (CLCI)

The following section gives the logic for converting a US Domestic circuit id value from ICORE into the standard US Domestic Circuit ID Format as given above.

**NOTE** that if the ICORE circuit id value is not for a US Domestic circuit id or does not follow the ICORE-internal rules for formatting US Domestic circuit id values, then the conversion will fail and no standard US Domestic Circuit ID Format value can be generated.

The possible circuit id values from ICORE are like these ones below:

43/HFGM/000169    /SUV   
14/YHGP/000057 004/SUV   
45.HXGM.005177..SUV   
45.YHGJ.000897.001.SUV

00/QHYU/010560/001/SBZ /  
 22/HFFJ/001458/ SUV /  
 22/HFFS/000071 SUV /

| **Character Position** | **Name** | **Data Type** |
| --- | --- | --- |
| 1-2 | Prefix | Char(2) |
| 3-4 | Service Code | Char(2) |
| 5-6 | Modifier | Char(2) |
| 7-12 | Serial Number | Number(6) |
| 13-15 | Suffix | Char(3) |
| 16-19 | Company Code | Char(4) |

Normalize it into what is called the “**BMP format**” as defined below. Replace with space if a position is missing.

|  |  |  |
| --- | --- | --- |
| **Field Name** | **BMP Length** | **BMP Serial Format Rule** |
| Prefix | VARCHAR2(2) A/N or Blanks | strip leading zeroes and left justify, fixed length, can be A/N. |
| Service Code | VARCHAR2(2) A/N or Blanks | **Mandatory**: left justify, fixed length |
| Modifier | VARCHAR2(2) A/N or Blanks | left justify, fixed length |
| Serial Number | NUMBER(6) Numeric | **Mandatory**: strip leading zeroes and right justify |
| Suffix | VARCHAR2(3) A/N or 3-Blanks | <BEGIN Circuit-Conv-Correction.2015-02-24> ~~Zero fill to 3 (except all blanks), fixed length, can be A/N.~~ Fill from left to right, omitting leading zeroes; for all zeroes or empty, leave empty. <END Circuit-Conv-Correction.2015-02-24> |
| Company Code | VARCHAR2(4) A/N or Blanks | left justify, fixed length |

For prefix , left justify, strip leading zeros, blank pad on the right if only one A/N letter.

For serial Number, replace the leading zeros with spaces, and right justify.

For company code, add one a space at the end if only three characters

For suffix, three spaces if empty or if numeric, right pad with zeros when less than three digits.

Examples

|  |  |
| --- | --- |
| **ASI\_XREF\_CKT\_ID** | **CUSTOMER\_CKT\_CLCI** |
| 43/HFGM/000169    /SUV | ‘43HFGM 169 SUV ‘ |
| 45.YHGJ.000897.001.SUV | ‘45YHGJ 897001SUV ‘ |
| 00/QHYU/010560/001/SBZ / | ‘ QHYU 10560001SBZ ‘ |
|  |  |

A complete C program that executes this conversion algorithm is attached as an OLE object.



As the last step, create a “**Standard US Domestic Circuit ID Format**” value from this “**BMP format**” value by inserting a “/” character between these components of the “**BMP format**”:

Prefix, Service Code, Modifier, Serial Number, Suffix, Company Code

If a “segment” component exists in addition at the end, then add a “/” character in front of it, too.

Remove all space characters (“ “) from the resulting string.

If the resulting string starts with a “/” character, then remove that character.

##### Logic for deriving the access circuit “description” and “speed” values

The following table gives the mapping for deriving the access circuit “description” and “speed” values for a given circuit id value in the format as described above under “US Domestic Circuit ID Format”.

The data from the table below is stored in the database table “GDB.CIRCUIT\_SERVICE\_CODES”. Any data needed while working with circuit description and/or speed data should be taken from that database table. For setting either a “id\_circuit\_service\_codes” or “speed” value, the “service code” value from a circuit id value in standard “US Domestic Circuit ID Format” (CLCI) format is mapped into the “service\_code” column, and the corresponding values from the column “id” or “speed” are used.

*NOTE that for circuit id values not in that format (as for MoW circuit ids !) “description” and “speed” values cannot be derived and must therefore be left empty !*

| **Service Code** | **Description** | **Speed** |
| --- | --- | --- |
| AA | PACKET ACCESS LINE - ANALOG | 56K |
| AB | PACKET NETWORK TRUNK | 56K |
| AC | LCPC.4 Service | 1.5M |
| AF | COMMERCIAL AUDIO (FULL TIME) | 1.5M |
| AG | T155/SONET Service OC-3 (Optical Carrier) | 155M |
| AH | Synchronous Digital Hierarchy (SDH) | 1.5M |
| AJ | PACKET ACCESS LINK | 56K |
| AL | ALTERNATE SERVICES | 56K |
| AM | PACKET-INTERNETWORK LINK | 56K |
| AQ | ACCUNET SPECTRUM OF DIGITAL SERVICES 9.6 Kb/S | 9.6K |
| AR | ACCUNET SPECTRUM OF DIGITAL SERVICES 56 Kb/S | 56K |
| BB | APLS GigE full duplex Gigabit Ethernet 50 Mbps | 50Mbps |
| BC | APLS GigE full duplex Gigabit Ethernet 150 Mbps | 150Mbps |
| BE | APLS GigE full duplex Gigabit Ethernet 600 Mbps | 600Mbps |
| BF | APLS GigE full duplex Gigabit Ethernet 10 GigE | 10Gig |
| CA | SSN ACCESS | 56K |
| CE | SSN DIRECTLY HOME SUBSCRIBER LINE | 56K |
| CH | OCC DIGITAL FACILITY - HIGH SPEED(OBSOLETE) | 1.5M |
| CL | CENTREX CO LINE | 56K |
| CN | SSN NETWORK TRUNK | 56K |
| CO | OCC OVERSEAS CONNECTING FACILITY (OBSOLETE) | 56K |
| CS | CHANNEL SERVICE | 56K |
| CT | SSN TIE TRUNK | 56K |
| CV | OCC VOICE GRADE FACILITY(OBSOLETE) | 56K |
| CX | CENTREX STATION LINE(OBSOLETE) | 56K |
| CY | DIGITAL DATA 19.2 Kb/s | 19.2K |
| DA | DIGITAL DATA OFF NET EXTENSION | 9.6K |
| DB | ACCUNET Reserved 1.5Mb/s ACCESS LINE | 1.5M |
| DC | DIGITAL DATA 384 Kb/s | 384K |
| DD | DISPATCH DATA(Obsolete) | 56K |
| DE | DIGITAL DATA 512 Kb/s | 512K |
| DF | ACCUNET Reserved 1.5Mb/s HUB-TO-HUB | 1.5M |
| DG | ACCUNET Reserved 1.5Mb/s HUB-TO-EARTH STATION | 1.5M |
| DH | DIGITAL SERVICE 1.5 Mb/s (Includes Derived Channels) | 1.5M |
| DI | DIRECT-IN-DIAL | 1.5M |
| DJ | DIGIT TRUNK | 1.5M |
| DM | Interlata Digital Service Higher Than 45 Mb/s (Obsolete) | 56K |
| DN | DIGITAL SERVICE 45 Mb/s | 45M |
| DO | DIRECT-OUT-DIAL | 56K |
| DP | DIGITAL DATA-2.4 Kb/s | 2.4K |
| DQ | DIGITAL DATA 4.8 Kb/s | 4.8K |
| DR | DIGITAL DATA 9.5 Kb/s | 9.6K |
| DS | DIGITAL DATA SUBRATE Speeds (Obsolete) | VARIOUS |
| DT | DIGITAL DATA 768 Kb/s | 768K |
| DU | DIGITAL DATA 2.048 Mb/s | 2.048M |
| DV | DIGITAL DATA 64 Kb/s | 64K |
| DW | DIGITAL DATA 56 Kb/s | 56K |
| DX | DIGITAL DATA 128 Kb/s | 128K |
| DY | DIGITAL SERVICE UNDER 1.0 Mb/s | VARIOUS |
| DZ | DIGITAL DATA 256 Kb/s | 256K |
| EA | SWITCHED ACCESS (OBSOLETE) | 56K |
| EB | END OFFICE TRUNK | 56K |
| EC | TANDEM TRUNK | 56K |
| EF | ENTRANCE FACILITY VOICE GRADE | 56K |
| EM | EMERGENCY REPORTING CENTER TRUNK | 56K |
| EN | NETWORK ACCESS FACILITY(OBSOLETE) | 56K |
| EO | EQUIPMENT ONLY(OBSOLETE) | 56K |
| EP | ENTRANCE FACILITY PROGRAM GRADE | 56K |
| EQ | EQUIPMENT ONLY (NETWORK ELEMENT) ASSIGNMENT | EQPT |
| ES | EXTENSION SERVICE-VOICE GRADE | 56K |
| EW | OFF NETWORK MTS/WATS EQUIVALENT SERVICE | 56K |
| FD | PRIVATE LINE DATA - ANALOG | 56K |
| FP | FULL PERIOD TELEPHONE (OBSOLETE) | 56K |
| FR | FIRE DISPATCH | 56K |
| FT | FOREIGN EXCHANGE TRUNK | 56K |
| FV | VOICE GRADE FACILITY | 56K |
| FX | FOREIGN EXCHANGE LINE | 56K |
| GC | GOVERNMENT CONTROL TELEGRAPH (OBSOLETE) | 56K |
| GD | GOVERNMENT DATA (OBSOLETE) | 56K |
| GL | GOVERNMENT PRIVATE LINE (OBSOLETE) | 56K |
| GM | GOVERNMENT METERING (OBSOLETE) | 56K |
| GP | GOVERNMENT TELEPHONE (OBSOLETE) | 56K |
| GQ | GOVERNMENT UNKNOWN (OBSOLETE) | 56K |
| GR | GOVERNMENT FULL PERIOD RADIO(OBSOLETE) | 56K |
| GX | GOVERNMENT FX (OBSOLETE) | 56K |
| IA | DIGITAL DATA 192 Kb/s | 192K |
| IB | DIGITAL DATA 1.152 Mb/s | 1.152M |
| IC | DIGITAL DATA 320 Kb/s | 320K |
| ID | DIGITAL DATA 448 Kb/s | 448K |
| IE | DIGITAL DATA 576 Kb/s | 576K |
| IF | DIGITAL DATA 640 Kb/s | 640K |
| IG | DIGITAL DATA 704 Kb/s | 704K |
| IH | Fractional ACCUNET(r) T45 4.6 Mb/s | 4.6M |
| II | DIGITAL DATA 1.204 Mb/s | 1.204M |
| IJ | Fractional ACCUNET(r) T45 6.2 Mb/s | 6.2M |
| IK | Fractional ACCUNET(r) T45 7.7 Mb/s | 7.7M |
| IL | Fractional ACCUNET(r) T45 10.8 Mb/s | 10.8M |
| IM | ACCUNET(r) T32 Service | 32M |
| IN | DIGITAL DATA 1.536 Mb/s | 1.536M |
| IP | DIGITAL DATA 1.984 Mb/s | 1.984M |
| IQ | Synchronous Digital Hierarchy (SDH) | 45M |
| IS | ACCU-RING LAN WAN 10Mbps | 10M |
| IT | INTERTANDEM TIE TRUNK | 56K |
| IU | ACCU-RING LAN WAN 100Mbps | 100M |
| IV | OC12 (Optical Carrier) SONET Service | 622M |
| IW | OC-48 Inter-Office Channel Service | 2.5G |
| IX | APLS OC-192 Service | 10G |
| IY | Digital Data 6.312 Mbps | 6.312M |
| IZ | APLS GigE full duplex Gigabit Ethernet 1 GigE | 1 Gig |
| JD | CONTRACT SERVICES MEASURED REMOTE PORT | 56K |
| JE | CONTRACT SERVICES DATA TRANSMISSION CAPABILITY | 56K |
| JF | CONTRACT SERVICES/TARIFF 12 DIGITAL SERVICE | 1.5M |
| KA | AT&T SERVING OFFICE SWITCH CONNECTION (CPA)(OBSOLETE) | 56K |
| KB | VOICE-GRADE CONNECTION TO HYBRID NODAL SERVICE | 56K |
| KC | CONNECTION TO HYBRID NODAL SERVICE 56 Kb/s | 56K |
| KJ | ISDN B CHANNEL | 56K |
| KT | SWITCHING SYSTEM CONNECTION | 56K |
| KZ | ISDN D CHANNEL | 56K |
| MA | CELLULAR ACCESS TRUNK (2 WAY) | 56K |
| ND | NETWORK DATA LINE | 56K |
| OC | CENTREX CU STATION LINE OFF PREMISES(OBSOLETE) | 56K |
| OI | OFF-PREMISES INTERCOMMUNICATION STATION LINE | 56K |
| ON | OFF-NETWORK ACCESS LINE | 56K |
| OP | OFF-PREMISES EXTENSION | 56K |
| OS | OFF-PREMISES PBX STATION LINE (DIAL) (MANUAL) | 56K |
| PA | PROTECTIVE ALARM (AC) | 56K |
| PG | PAGING | 56K |
| PL | PRIVATE LINE - VOICE | 56K |
| PM | PROTECTIVE MONITORING | 56K |
| PR | PROTECTIVE RELAYING - VOICE GRADE | 56K |
| PX | PBX STATION LINE | 56K |
| QL | Circuits assigned incorrectly in database. | 56K |
| QS | PACKET ACCESS LINE - DIGITAL | 56K |
| RT | RADIO LAND LINE | 56K |
| SA | SATELLITE TIE TRUNK | 56K |
| SN | SSN-SPECIAL ACCESS TERMINATION | 56K |
| SS | DATAPHONE SELECT-A-STATION | 56K |
| TA | TANDEM TIE TRUNK | 56K |
| TD | DISPATCH TIE LINE (OBSOLETE) | 56K |
| TL | NON-TANDEM TIE TRUNK | 56K |
| UA | Fractional ACCUNET(r) T45 3.088 Mb/s (2) | 3.088M |
| UB | Fractional ACCUNET(r) T45 9.2 Mb/s (6) | 9.2M |
| UC | Fractional ACCUNET(r) T45 12 Mb/s (8) | 12M |
| UD | DIGITAL DATA 4.096 Mb/s | 4.09M |
| UE | DIGITAL DATA 6.144 Mb/s | 6.144M |
| UF | DIGITAL DATA 8.192 Mb/s | 8.192M |
| UG | DIGITAL DATA 832 Kb/s | 832K |
| UH | DIGITAL DATA 896 Kb/s | 896K |
| UI | DIGITAL DATA 960 Kb/s | 960K |
| UJ | DIGITAL DATA 1.280 Mb/s | 1.28M |
| UK | DIGITAL DATA 1.920 Mb/s | 1.92M |
| VF | COMMERCIAL TELEVISION (FULL TIME) | VARIOUS |
| VM | CONTROL/REMOTE METERING - VOICE GRADE | 56K |
| VR | NON-COMMERCIAL TELEVISION | VARIOUS |
| ZA | ALARM CIRCUIT | VARIOUS |
| ZC | CALL AND TALK CIRCUIT | 56K |
| ZD | DATA LINE SWITCHING TEST CIRCUITS(OBSOLETE) | 56K |
| ZF | ORDER CIRCUIT FACILITY | VARIOUS |
| ZM | MEASUREMENT AND RECORDING CIRCUIT | VARIOUS |
| ZP | TEST CIRCUIT PLANT SERVICE CENTER | 56K |
| ZS | SWITCHING, CONTROL AND TRANSFER CIRCUIT | VARIOUS |
| ZV | ORDER CIRCUIT SERVICE | 56K |

<QC129121-US856984>

For AVPNoLS circuits, their service codes are always ‘KR’, but the circuits could have various speeds, a static value defined in GDB.CIRCUIT\_SERVICE\_CODES is incorrect. GCP will remove this record for ‘KR’, and let access’s speed lookup will take the ‘Fall-back’ approach from port\_asgmt.port\_speed.

</QC129121-US856984>

<Defect-90537>

For circuits, where PORT\_ASGMT.pos\_option = ‘MLPPP’, use PORT\_ASGMT.port\_speed value instead of the value from the above table.

</Defect-90537>

<BEGIN 254035c.DEFECT-55173>

##### Logic for pulling the access circuit “speed” values from “port\_asgmt”

For “access circuits” where no speed value can be determined via its “circuit service code” the data shall be pulled from “ICORE.port\_asgmt” (as a “fall-back”). The column to pull the data from is

* **ICORE.port\_asgmt.port\_speed**

Database traversals: (CUST\_ACCESS)

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| CUST\_ACCESS.site\_id | PORT\_ASGMT.site\_id |

Database traversals: (CUST\_ACCESS\_CKT)

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| CUST\_ACCESS\_CKT.site\_id | PORT\_ASGMT.site\_id |

<END 254035c.DEFECT-55173>

**<289037c-ICORE-020>**

***GDB ASSET\_ACCESS\_CIRCUIT\_DETAILS***

The following only applies to asset type = ACCESS\_CIRCUIT. Additionally, for source key = CUST\_ACCESS.site\_id, only records where CUST\_ACCESS.acc\_ckt <> NULL should be loaded.

A one time load/update of the GDB ASSET\_ACCESS\_CIRCUIT\_DETAILS table for all GDB ASSET.id’s with asset type = ACCESS\_CIRCUIT [ICORE access circuits in CUST\_ACCESS] will be needed.

On subsequent loads, insert a new record in GDB ASSET\_ACCESS\_CIRCUIT\_DETAILS if a new GDB ASSET.id with asset type = ACCESS\_CIRCUIT has been added [ICORE access circuit added to CUST\_ACCESS]. Update existing records in GDB ASSET\_ACCESS\_CIRCUIT\_DETAILS if the ICORE source data has changed.

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_ACCESS\_CIRCUIT\_DETAILS column** | **Processing comment** |
| - | ASSET\_ID | Same as ID in GDB ASSET and ID\_ASSET in ASSET\_EXT\_ACCESS\_CIRCUIT *[derived via Enterprise Key Translation on ICORE CUST\_ACCESS.site\_id]* |
| - | SOURCE\_SYSTEM | Set to “ICORE” |
| CIRCUIT.circ\_cat | SERVICE\_TYPE | Select CIRCUIT.circ\_cat from CIRCUIT, CUST\_ACCESS where CIRCUIT.a\_site\_id = CUST\_ACCESS.site\_id and CUST\_ACCESS.site\_id = <input ICORE Circuit key> |
| CUST\_ACCESS.site\_id | SITE\_ID\*  [via Enterprise Key Translation] | Same as Site Id in GDB SITE  select CUST\_ACCESS.site\_id from CUST\_ACCESS where CUST\_ACCESS.site\_id = <input ICORE Circuit key>]  Perform Enterprise Key Translation on CUST\_ACCESS.site\_id to derive GDB Site Id |
| - | SITE\_NAME | Set to NULL |
| - | STATUS | Set to NULL |
| - | STATUS\_DATE | Set to NULL |
| - | DATE\_INSTALLED | Set to NULL |
| - | DATE\_DEINSTALLED | Set to NULL |
| CIRCUIT.notes | NOTES | Select CIRCUIT.notes from CIRCUIT, CUST\_ACCESS where CIRCUIT.a\_site\_id = CUST\_ACCESS.site\_id and CUST\_ACCESS.site\_id = <input ICORE Circuit key> |
| - | CONNECTIVITY\_TYPE | Set to NULL |
| - | USED\_FOR | Set to NULL |
| - | LEC\_PTT | Set to NULL |
| - | MANAGED\_THIRD\_PARTY\_INDICATOR | Set to NULL |
| - | THIRD\_PARTY\_TYPE | Set to NULL |
| - | MAINTENANCE\_CONTROL\_OFFICE | Set to NULL |
| - | ATT\_LEGAL\_ENTITY\_ID | Set to NULL |
| EQUIPMENT.equip\_name | CONNECTED\_DEVICE | Select EQUIPMENT.equip\_name from EQUIPMENT, CIRCUIT, CUST\_ACCESS where EQUIPMENT.equip\_id = CIRCUIT.a\_equip\_id and CIRCUIT.a\_site\_id = CUST\_ACCESS.site\_id and CUST\_ACCESS.site\_id = <input ICORE Circuit key> |
| - | ID\_CHANGE\_TRACKING | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |

\*This is the EKT-transformed Site Id that will be sent out in the IECAD API reponse to the client. It is not the Site Id found in the ICORE CUST\_ACCESS or SITE tables.

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| CUST\_ACCESS.site\_id | CIRCUIT.a\_site\_id |
| CIRCUIT.a\_equip\_id | EQUIPMENT.equip\_id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| CUST\_ACCESS | ACC\_CKT | (must not be empty or null) |

**<End of 289037c-ICORE-020>**

**<289037c-ICORE-030>**

***GDB CUSTOMER\_PORT\_DETAILS***

The following GDB CUSTOMER\_PORT\_DETAILS tableapplies to asset type = ACCESS\_CIRCUIT [not NETWORK CONNECTION]. Additionally, for source key = CUST\_ACCESS.site\_id, only records where CUST\_ACCESS.acc\_ckt <> NULL should be loaded.

Where multiple ports are associated with an ACCESS\_CIRCUIT (i.e. CUST\_ACCESS.site\_id), the query on PORT\_ASGMT.nad\_complete will return a nad\_complete date for each port (PORT\_ASGMT.cust\_port) associated with the circuit. As a result, there can be multiple entries for nad\_complete and other port attributes (.slot, .port) for a given CUSTOMER\_PORT\_DETAILS.asset\_id . Since asset\_id will not always be a unique value in each row, a composite key = asset\_id + port\_id has to be used.

A one time load/update of the GDB CUSTOMER\_PORT\_DETAILS table for all GDB ASSET.id’s with asset type = ACCESS\_CIRCUIT [ICORE access circuits in CUST\_ACCESS] will be needed.

On subsequent loads, insert a new record in GDB CUSTOMER\_PORT\_DETAILS if a new GDB ASSET.id with asset type = ACCESS\_CIRCUIT has been added [ICORE access circuit added to CUST\_ACCESS]. Update existing records in GDB CUSTOMER\_PORT\_DETAILS for any circuits where there has been a change in the ICORE source data.

If a query returns multiple results for ICORE.PORT\_ASGMT (ie. multiple PORT\_ASGMT.cust\_port rows), then each returned row should translate to a row in the GDB.CUSTOMER\_PORT\_DETAILS table with the same .asset\_id, but a different .port\_id. An example of how the CUSTOMER\_PORT\_DETAILS table should be populated is shown at the end of this requirement.

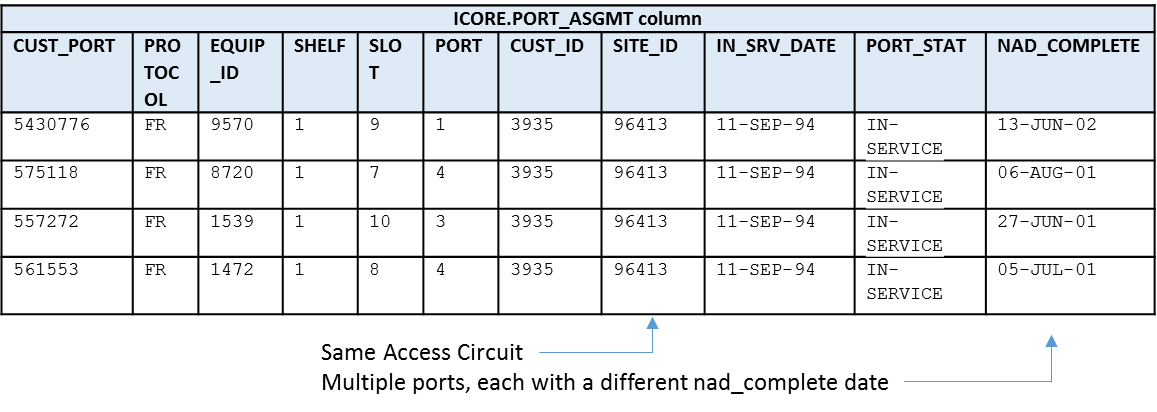
|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB CUSTOMER\_PORT\_DETAILS column** | **Processing comment** |
| - | ASSET\_ID | Same as ID in GDB ASSET, ID\_ASSET in ASSET\_EXT\_ACCESS\_CIRCUIT or ASSET\_ID in ASSET\_ACCESS\_CIRCUIT\_DETAILS *[derived via Enterprise Key Translation on ICORE CUST\_ACCESS.site\_id]*  *Composite Key = ASSET\_ID + PORT\_ID* |
| PORT\_ASGMT.cust\_port | PORT\_ID | select PORT\_ASGMT.cust\_port from PORT\_ASGMT, CUST\_ACCESS where PORT\_ASGMT.site\_id = CUST\_ACCESS.site\_id and CUST\_ACCESS.site\_id = <input ICORE Circuit key>  *Composite Key = ASSET\_ID + PORT\_ID* |
| - | SOURCE\_SYSTEM | Set to “ICORE” |
| ~~CIRCUIT.A\_END\_NETWK\_ID , CIRCUIT.A\_ASSOC\_SWITCH, CIRCUIT.A\_ASSOC\_SHELF , CIRCUIT.A\_ASSOC\_SLOT~~  PORT\_ASGMT.shelf,  PORT\_ASGMT.slot,  PORT\_ASGMT.port | PORT\_INTERFACE | ~~select ‘net:’ || A\_END\_NETWK\_ID || ',switch:' || A\_ASSOC\_SWITCH || ‘,shelf:' || A\_ASSOC\_SHELF || ‘,slot:’ || A\_ASSOC\_SLOT from CIRCUIT, CUST\_ACCESS where CIRCUIT. a\_site\_id =~~~~CUST\_ACCESS. site\_id and CUST\_ACCESS.site\_id~~~~=~~ *~~<input ICORE Circuit key>~~*  select 'Shelf:' ||shelf||', Slot:'||slot||', Port:'||port from PORT\_ASGMT where PORT\_ASGMT.site\_id = <input ICORE Circuit key/CUST\_ACCESS.site\_id>  *[concatenate output with added prefix}* |
| PORT\_ASGMT.port\_stat | STATUS | select PORT\_ASGMT.port\_stat from PORT\_ASGMT, ~~CIRCUIT,~~ CUST\_ACCESS where PORT\_ASGMT.site\_id ~~equip\_id = CIRCUIT.a\_equip\_id and CIRCUIT.a\_site\_id~~ = CUST\_ACCESS.site\_id and CUST\_ACCESS.site\_id = <input ICORE Circuit key> |
| PORT\_ASSGMT.nad\_complete | STATUS\_DATE | select PORT\_ASGMT.nad\_complete from PORT\_ASGMT, ~~CIRCUIT~~, CUST\_ACCESS where PORT\_ASGMT.site\_id ~~equip\_id = CIRCUIT.a\_equip\_id and CIRCUIT.a\_site\_id~~ = CUST\_ACCESS.site\_id and CUST\_ACCESS.site\_id = <input ICORE Circuit key> |
| EQUIPMENT.install\_date | DATE\_INSTALLED | Select EQUIPMENT.install\_date from EQUIPMENT, PORT\_ASGMT, CUST\_ACCESS where EQUIPMENT.equip\_id = PORT\_ASGMT.equip\_id and PORT\_ASGMT.site\_id = CUST\_ACCESS.site\_id and CUST\_ACCESS.site\_id = <input ICORE Circuit key> |
| - | DATE\_DEINSTALLED | Set to NULL |
| SITE.site\_glbl\_dlci | GLOBAL\_DLCI | Select SITE.site\_glbl\_dlci from SITE, CUST\_ACCESS where SITE.site\_id = CUST\_ACCESS.site\_id and CUST\_ACCESS.site\_id = <input ICORE Circuit key> |
| - | ID\_CHANGE\_TRACKING | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| ~~CUST\_ACCESS.site\_id~~ | ~~CIRCUIT.a\_site\_id~~ |
| ~~CIRCUIT.a\_equip\_id~~ | ~~EQUIPMENT.equip\_id~~ |
| ~~CIRCUIT.a\_equip\_id~~ | ~~PORT\_ASGMT.equip\_id~~ |
| CUST\_ACCESS.site\_id | PORT\_ASGMT.site\_id |
| CUST\_ACCESS.site\_id | EQUIPMENT.site\_id |
| CUST\_ACCESS.site\_id | SITE.site\_id |

Example:

Query on ICORE.PORT\_ASGMT returns multiple rows for PORT\_ASGMT.site\_id = CUST\_ACCESS.site\_id = 96413:



Recording of ICORE.PORT\_ASGMT query results for PORT\_ASGMT.site\_id = CUST\_ACCESS.site\_id = 96413 in CUSTOMER\_PORT\_DETAILS table:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **GDB.CUSTOMER\_PORT\_DETAILS column** | | | | | | | | |
| **ASSET\_ID** | **PORT\_ID** | **SOURCE\_SYSTEM** | **PORT\_INTERFACE** | **STATUS** | **STATUS\_DATE** | **DATE\_INSTALLED** | **DATE\_DEINSTALLED** | **GLOBAL\_DLCI** |
| 654325 | 5430776 | ICORE | Shelf: 1, Slot:9, Port:1 | IN-SERVICE | 13-JUN-02 | 09-SEP-94 |  | 100 |
| 654325 | 575118 | ICORE | Shelf: 1, Slot:7, Port:4 | IN-SERVICE | 06-AUG-01 | 09-SEP-94 |  | 100 |
| 654325 | 557272 | ICORE | Shelf: 1, Slot:10, Port:3 | IN-SERVICE | 27-JUN-01 | 09-SEP-94 |  | 100 |
| 654325 | 561553 | ICORE | Shelf: 1, Slot:8, Port:4 | IN-SERVICE | 05-JUL-01 | 09-SEP-94 |  | 100 |
| 152324 | 343553 | ICORE | Shelf: 1, Slot:11, Port:3 | IN-SERVICE | 21-FEB-00 | 14-FEB-00 |  | 45 |
| 865021 | 395368 | ICORE | Shelf: 1, Slot:1, Port:31 | IN-SERVICE | 03-JAN-03 | 28-DEC-02 |  | 89 |

**<End of 289037c-ICORE-030>**

**<289037c-ICORE-040>**

***GDB PVC\_DETAILS***

A one time load/update of the GDB PVC\_DETAILS table for all GDB ASSET Id’s with asset type = ACCESS\_CIRCUIT [ICORE access circuits in CUST\_ACCESS] will be needed.

On subsequent loads, insert a new record in GDB PVC\_DETAILS if a new PVC Id has been added to ICORE PVC for an existing GDB Asset Id in PVC\_DETAILS OR a new GDB ASSET Id with asset type = ACCESS\_CIRCUIT has been added [ICORE access circuit added to CUST\_ACCESS].

Update existing records in GDB PVC\_DETAILS for any existing PVC Id’s in GDB PVC\_DETAILS for which there has been a change in ICORE PVC.

A single Access Circuit can be associated with multiple unique PVC’s. So, an ASSET\_ID (EKT-transformed CUST\_ACCESS.site\_id) in ASSET\_ACCESS\_CIRCUIT\_DETAILS/CUSTOMER\_PORT\_DETAILS can be associated with many PVC Id’s.

The PVC table is linked to CUST\_ACCESS by site id. Since a given CUST\_ACCESS.site\_id can the local site for a PVC connection or the remote site for a PVC connection, the query for PVC details needs to be done on both local site and remote site attributes in the PVC table. A query similar to the following should be done:

select PVC.pvc\_id, PVC.pvc\_lsite\_id, PVC.pvc\_rsite\_id, PVC.lswitch, PVC.rswitch, PVC.pvc\_ldlci, PVC.pvc\_rdlci, PVC.pvc\_lvpi, PVC.pvc\_lvci, PVC.pvc\_rvpi, PVC.pvc\_rvci from PVC where PVC.pvc\_lsite\_id = <*CUST\_ACCESS.site\_id key value*> or PVC.pvc\_rsite\_id = <*CUST\_ACCESS.site\_id key value*>.

For a given <*CUST\_ACCESS.site\_id key value*>, there may be matches on only PVC.pvc\_lsite\_id, on only PVC.pvc\_rsite\_id or on both*. [Local site and remote site are relative to which access circuit you’re starting from. A CUST\_ACCESS.site\_id can be local or remote, based on your perspective].*

Copy the PVC.pvc\_id, PVC.lswitch, PVC.rswitch, PVC.pvc\_ldlci, PVC.pvc\_rdlci, PVC.pvc\_lvpi, PVC.pvc\_lvci, PVC.pvc\_rvpi and PVC.pvc\_rvci values into the corresponding columns in GDB PVC\_DETAILS. The PVC.pvc\_lsite\_id and PVC.pvc\_rsite\_id values require Enterprise Key Translation.

Since many PVC ID’s can be associated with a GDB ASSET ID and the same PVC ID can be associated with two different ASSET ID’s, a composite key (ASSET ID + PVC ID) will be used in GDB PVC\_DETAILS.

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB PVC\_DETAILS column** | **Processing comment** |
| - | ASSET\_ID | Same ASSET\_ID in ASSET\_ACCESS\_CIRCUIT\_DETAILS *[derived via Enterprise Key Translation on ICORE CUST\_ACCESS.site\_id key value]*    Composite key used in PVC\_DETAILS: ASSET\_ID + PVC\_ID |
| PVC.pvc\_id | PVC\_ID | Composite key used in PVC\_DETAILS: ASSET\_ID + PVC\_ID |
| - | SOURCE\_SYSTEM | Set to “ICORE” |
| PVC.pvc\_lsite\_id | LOCAL\_GDB\_SITE\_ID  *[via Enterprise Key Translation]* | Same as Site Id in GDB SITE  Perform Enterprise Key Translation on PVC.pvc\_lsite\_id to derive GDB Site Id |
| PVC.pvc\_rsite\_id | REMOTE\_GDB\_SITE\_ID  *[via Enterprise Key Translation]* | Same as Site Id in GDB SITE  Perform Enterprise Key Translation on PVC.pvc\_rsite\_id to derive GDB Site Id |
| PVC.lswitch | LOCAL\_DEVICE\_NAME |  |
| PVC.rswitch | REMOTE\_DEVICE\_NAME |  |
| PVC.pvc\_ldlci | LOCAL\_DLCI |  |
| PVC.pvc\_rdlci | REMOTE\_DLCI |  |
| PVC.pvc\_lvpi | LOCAL\_VPI |  |
| PVC.pvc\_lvci | LOCAL\_VCI |  |
| PVC.pvc\_rvpi | REMOTE\_VPI |  |
| PVC.pvc\_rvci | REMOTE\_VCI |  |
| - | ID\_CHANGE\_TRACKING | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |

Example:

For Asset Id = 614844, there are three associated PVC’s. For Asset Id= 919655, there are two associated PVC’s. PVC Id = 2640884 is associated with two Access Circuits.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **GDB.PVC\_DETAILS column** | | | | | | | | | | | | |
| **ASSET\_ID** | **PVC\_ID** | **SOURCE\_**  **SYSTEM** | **LOCAL\_**  **GDB\_**  **SITE\_ID** | **REMOTE\_GDB\_SITE\_ID** | **LOCAL\_DEVICE\_NAME** | **REMOTE\_DEVICE\_NAME** | **LOCAL\_**  **DLCI** | **REMOTE\_**  **DLCI** | **LOCAL\_**  **VPI** | **LOCAL\_**  **VCI** | **REMOTE\_**  **VPI** | **REMOTE\_**  **VCI** |
| 614844 | 5079619 | ICORE | 707434 | 106754 | phlapa83 | atlnga04 | 841 | 194 | 1 | 816 | 1 | 41 |
| 614844 | 5027972 | ICORE | 707434 | 107043 | phlapa83 | MXCYMXE2 | 696 | 38 | 1 | 105 | 1 | 198 |
| 614844 | 5226638 | ICORE | 707434 | 716056 | phlapa83 | okldca22 | 988 | 405 | 1 | 34 | 1 | 194 |
| 728329 | 2578372 | ICORE | 804271 | 726098 | n54nyc509 | hrbgpa01 | 50 | 3 | 1 | 41 | 1 | 40 |
| 919655 | 3953682 | ICORE | 919543 | 740301 | n54nyc504 | attga504 | 167 | 188 | 1 | 40 | 1 | 154 |
| 919655 | 3042220 | ICORE | 236804 | 740301 | nwknj021 | attga504 | 16 | 220 | 1 | 31 | 1 | 125 |
| 83921\* | 2640884 | ICORE | 83965 | 65893 | jcvlfl60 | rcmdva03 | 211 | 153 | 1 | 81 | 1 | 110 |
| 655092\* | 2640884 | ICORE | 83965 | 65893 | jcvlfl60 | rcmdva03 | 211 | 153 | 1 | 81 | 1 | 110 |

\**One asset is on the left side (PVC\_LSITE) of the PVC and the other is on the right side (PVC\_RSITE) of the same PVC.*

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| CUST\_ACCESS.site\_id | PVC.pvc\_lsite\_id |
| CUST\_ACCESS.site\_id | PVC.pvc\_rsite\_id |

**<End of 289037c-ICORE-040>**

#### HLD-254035-GCP-FLOW-ICORE-102-B [Icore Data Flow into Asset CUST\_ACCESS\_CKT (“Access Circuit”)]

The following section is about “Access Circuit” assets. The data handled here has “master” data in the ICORE “CUST\_ACCESS” table.

Implement the following:

Process the following Icore records into the GDB asset object and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

*NOTE: Only include records where “ICORE.CUST\_ACCESS\_CKT.circuit\_id” is not null and is not empty !*

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET column** | **Processing comment** |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘ASSET’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| CUST\_ACCESS\_CKT.site\_id; CUST\_ACCESS\_CKT.seq\_no | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| - | id\_asset\_type | Referencing ASSET\_TYPE ‘ACCESS\_CIRCUIT’ |

**NOTE:**

For ICORE circuit id values it is not probable that they are already in CLCI format. Nevertheless it is best to check whether a circuit id can be converted to the CLCI format, and if yes, then check whether the CLCI format value is identical to the “raw” value as retrieved directly from ICORE. If the latter is the case, then no “raw” format identifier data set shall be created !

Asset Identifier (1): “Raw” Circuit ID value

(Only to be created if the “raw” value and the “CLCI” value are different; see “NOTE” above)

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘CIRCUIT\_ID\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_asset\_identifier | ASSET\_IDENTIFIER.id |
| - | id\_change\_tracking | ASSET\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘CIRCUIT\_ID\_IDENTIFIER’ |
| CUST\_ACCESS\_CKT.circuit\_id | Value |  |

Asset Identifier (2): “Standard US Domestic Circuit ID Format” Circuit ID value  
*(only to be created if a conversion is successful; see “Logic for converting a US Domestic circuit id value from ICORE into the standard US Domestic Circuit ID Format”)*

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘STANDARD\_FORMAT\_CIRCUIT\_ID\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_asset\_identifier | ASSET\_IDENTIFIER.id |
| - | id\_change\_tracking | ASSET\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘STANDARD\_FORMAT\_CIRCUIT\_ID\_IDENTIFIER’ |
| CUST\_ACCESS\_CKT.circuit\_id converted to a “Standard US Domestic Circuit ID Format” Circuit ID value (see “Logic for converting a US Domestic circuit id value from ICORE into the standard US Domestic Circuit ID Format”) | Value |  |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_EXT\_ACCESS\_CIRCUIT column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| (1) Use “CUST\_ACCESS\_CKT.circuit\_id” if no CLCI-formatted value could be converted; (2) Use the CLCI-formatted value of “CUST\_ACCESS\_CKT.circuit\_id” if a CLCI-formatted value could be converted; | circuit\_id\_value |  |
|  | id\_circuit\_id\_identifier\_type | (1) Referencing IDENTIFIER\_TYPE ‘CIRCUIT\_ID\_IDENTIFIER’ if no CLCI-formatted value could be converted; (2) Referencing IDENTIFIER\_TYPE ‘STANDARD\_FORMAT\_CIRCUIT\_ID\_IDENTIFIER’ if a CLCI-formatted value could be converted; |
| - | id\_circuit\_service\_codes | <to be derived; see “Logic for deriving the access circuit “description” and “speed” values”>;  NOTE:  Use “GDB.CIRCUIT\_SERVICE\_CODES.id” where “GDB.CIRCUIT\_SERVICE\_CODES.SERVICE\_CODE” matches the service code as derived from the currently processed circuit id value (in CLCI format; it does not work for “raw”-only circuit id values);  Use NULL if no match could be found |
| - | Speed | <to be derived; see “Logic for deriving the access circuit “description” and “speed” values”>;  NOTE:  Use “GDB.CIRCUIT\_SERVICE\_CODES.speed” where “GDB.CIRCUIT\_SERVICE\_CODES.SERVICE\_CODE” matches the service code as derived from the currently processed circuit id value (in CLCI format; it does not work for “raw”-only circuit id values);  <BEGIN 254035c.DEFECT-55173>  ~~Use NULL if no match could be found~~  If no match could be found, then use the logic under “Logic for pulling the access circuit “speed” values from “port\_asgmt”” to set the value.  <END 254035c.DEFECT-55173> |
| CUST\_ACCESS.mpc  (via  CUST\_ACCESS\_CKT.site\_id = CUST\_ACCESS.site\_id  ) | parent\_channel |  |
| CUST\_ACCESS.ia\_ckt  (via  CUST\_ACCESS\_CKT.site\_id = CUST\_ACCESS.site\_id  ) | parent\_circuit\_id\_value |  |
| <279006> SITE.BROADBAND\_IND | Is\_broadband | Use defined default value in the table definition when site.broadband\_ind is null |
| <279006> SITE.UPSTREAM\_ASYM\_SPEED | Upload\_speed |  |
| <286278> |  |  |
|  | ete\_access\_key | <286284-US600815-US600161>  See End-to-End Key Search and Generation [ Icore Data Flow ]  ~~select distinct ete\_access\_key, ‘USRP’ as ete\_key\_source~~  ~~from usrp.logical\_port~~  ~~where valid\_ind = 'Y'~~  ~~and icore\_site\_id = <site.id>~~  ~~or~~  ~~select distinct ete\_access\_key, ‘IEOD’ as ete\_key\_source~~  ~~from grid.circuit~~  ~~where clci (after removed all special chars) = <circuit\_id\_value after removed all special chars>~~  ~~Note: removing all special characters for a circuit id means to remove all special chars but keeping all alphabets and numbers, for example, for /KRGP/12345/ATI,keep only KRGP12345ATI~~  ~~if there is any other asset\_ext\_access\_circuit records with the same ete\_access\_key but associated with some CE port with a different icore site id:~~  ~~PHY\_PORT (with a different icore site id value, port type referencing to ‘CE’)-> ( USED BY/NULL ) > ASSET (access circuit asset with the same ete\_access\_key, but with a different asset.id)~~  ~~Wipe out those asset\_ext\_access\_circuit.ete\_access\_key to blank/null.~~ |
|  | ete\_key\_source | <286284-US600815-US600161>  See End-to-End Key Search and Generation [ Icore Data Flow ]  ~~See above. ete\_key\_source should be blank/null when ete\_access\_key is blank/null.~~ |
| </286278> |  |  |
| <288655a>  GPS.AT\_CIRCUIT\_BASE.AT3RDPARTYTYPE  Or  SIDBOR.CIRCUIT.thirdpartytype | Third\_party\_flag | Set third\_party\_flag = ‘Y’, if the retrieved value from one of the steps below = ‘LOA’ or ‘COR’;  Else set third\_party\_flag = ‘N’  Match with GPS.AT\_CIRCUIT\_BASE.atcircuitname = circuit\_id\_value  Or  Match with SIDBOR.CIRCUIT.telco\_ckt\_id = circuit\_id\_value  (Use ‘Access Circuit Matching logic’ section from below for the above comparison) |
| IPFR.DHV\_INDICATOR <292589-US847769> | DHV\_FLAG | Note: Need to check all the PVCs associated to the site and if atleast one PVC has the DHV\_INDICATOR set to ‘Y’, then, the value in GDB should also be set to ‘Y’.  For example, if four PVCs are returned for a particular lsite\_id or rsite\_id, and 2 PVCs are having the DHV\_INDICATOR=’Y’ and 2 PVCs are having the DHV\_INDICATOR=’N’. Since there is atleast one PVC with DHV\_INDICATOR =’Y’, the value in GDB should also be set to ‘Y’. |

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| CUST\_ACCESS\_CKT.site\_id | PORT\_ASGMT.site\_id |
| <279006> CUST\_ACCESS\_CKT.site\_id | SITE.site\_id |
| CUST\_ACCESS\_CKT.SITE\_ID | PVC.PVC\_LSITE\_ID or PVC.RSITE\_ID <292589> |
| PVC.PVC\_ID | IPFR.PVC\_ID <292589> |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |
| CUST\_ACCESS\_CKT | CIRCUIT\_ID | (is not empty or null) |

**Data Blocking (1)**

Do not process asset records that are linked to a GDB.SERVICE\_TYPE having a record in GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM for Icore:

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| CUST\_ACCESS\_CKT.site\_id | SITE.site\_id |
| SERVICE\_ASGMT.site\_id | SITE.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |
| GDB.SERVICE\_TYPE\_NOTATION.service\_name | SERVICE.serv\_name |
| GDB.SERVICE\_TYPE\_NOTATION.id\_service\_type | GDB.SERVICE\_TYPE.id |
| GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM.id\_service\_type | GDB.SERVICE\_TYPE.id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| SERVICE\_TYPE\_BLOCKED\_SYSTEM | BLOCKED\_SYSTEM | 13609 |

*Note:*

*The BLOCKED\_SYSTEM constraint value is the MOTS-ID of the system.*

**Data Blocking (2)**

Do not process asset records where the special logic for blocking INSTAR items hold.

The “<input site id variable (dep. on current scenario)>” here is “CUST\_ACCESS\_CKT.site\_id”.

See “Logic for excluding INSTAR items”.

#### HLD-254035-GCP-FLOW-ICORE-103-A [Icore Data Flow into Asset VPN (Layer-3 “Customer Network”)]

The following section is about “Customer Network (Layer-3)” assets. These types of assets are identified by having an entry in the ICORE.VPN table and additional records in the ICORE.PVC table that are connected to each other via the ICORE IPFR table. When the data is checked for selectability, only the “customer” side data from the ICORE.PVC table must be taken into account. (Details to be found below and at other appropriate places.)

Implement the following:

Process the following Icore records into the GDB asset object and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

*NOTE: Exclude (joined) records where (under “Database traversals”) for an associated PVC* ***both*** *the “PVC.pvc\_lcust\_id” and the “PVC.pvc\_rcust\_id” value point to a “real” customer. (This is a “degenerated” case.) Also to be excluded are records where at both sides AT&T is the customer. So only records where at one side a real customer is located and at the other side AT&T is located shall be taken into account. The VPN value may still be added/loaded if other associated PVC records do are “valid”, i.e. have AT&T at one end and a real customer at the other end.*

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET column** | **Processing comment** |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘ASSET’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| VPN.vpn\_id | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| - | id\_asset\_type | Referencing ASSET\_TYPE ‘CUSTOMER\_NETWORK’ |

Identifier (1): VPN ID

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘VPN\_ID\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_asset\_identifier | ASSET\_IDENTIFIER.id |
| - | id\_change\_tracking | ASSET\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘VPN\_ID\_IDENTIFIER’ |
| VPN.vpn\_id | Value |  |

Identifier (2): VPN Name (only insert if “VPN.vpn\_name” exists and is not empty or null)

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘VPN\_NAME\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_asset\_identifier | ASSET\_IDENTIFIER.id |
| - | id\_change\_tracking | ASSET\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘VPN\_NAME\_IDENTIFIER’ |
| VPN.vpn\_name | Value |  |

<286278>

Identifier (1): ETE\_VPN\_KEY

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE “E2E\_VPN\_KEY\_ASSET\_IDENTIFIER” |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_asset\_identifier | ASSET\_IDENTIFIER.id |
| - | id\_change\_tracking | ASSET\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE “E2E\_VPN\_KEY\_ASSET\_IDENTIFIER” |
| - | Value | <286284-US600815-US600161>  See End-to-End Key Search and Generation [ Icore Data Flow ]  ~~select e2e\_vpn\_key, ‘LPP’ as ete\_key\_source~~  ~~from icore.vpn~~  ~~where vpn\_id = <vpn\_id)~~  ~~or~~  ~~select ete\_vpn\_key, ‘USRP’ as ete\_key\_source~~  ~~from usrp.vpn~~  ~~where icore\_vpn\_id =<vpn\_id>~~  ~~and ete\_vpn\_key is not null~~  ~~or above query doesn’t return any value:~~  ~~select ete\_vpn\_key, ‘IEOD’ as ete\_key\_source, ete\_siteless\_key~~  ~~from grid.vpn~~  ~~where assigned\_vpn\_name =<vpn\_name> -- defect 58004 fix~~  ~~and ete\_vpn\_key is not null~~  ~~remove the same asset identifier (same ete\_vpn\_key) for other Customer Network assets.~~ |

</286278>

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_EXT\_CUSTOMER\_NETWORK column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| VPN.vpn\_id | network\_id |  |
| VPN.vpn\_name | network\_name |  |
|  | id\_network\_type | Referencing NETWORK\_TYPE ‘NT\_VPN’ |
| <BEGIN 259118> |  |  |
|  | <BEGIN DEFECT-FIX-11688.2015-03-12> |  |
| VPN.vpn\_type | id\_network\_sub\_type | (1) referencing NETWORK\_TYPE ‘**NST\_HUB**’ if  VPN.vpn\_type = ‘hub’;  (2) referencing NETWORK\_TYPE ‘**NST\_SPOKE**’ if  VPN.vpn\_type = ‘spoke’;  (3) referencing NETWORK\_TYPE ‘**NST\_HUB\_SPOKE**’ if  VPN.vpn\_type =  ('hub-spoke' or 'hub\_spoke');  (4) referencing NETWORK\_TYPE ‘**NST\_ANY\_TO\_ANY**’ if  VPN.vpn\_type =  (NULL or blank or 'any-to-any');  (5) referencing NETWORK\_TYPE ‘**NST\_UNDEFINED**’ if  VPN.vpn\_type =  <any other value>; |
|  | <END DEFECT-FIX-11688.2015-03-12> |  |
| <BEGIN 259118> |  |  |
| <286278> |  |  |
| - | ETE\_SITELESS\_KEY | select ete\_siteless\_key  from usrp.vpn  where icore\_vpn\_id =<vpn\_id>  and ete\_siteless\_key is not null  or above query doesn’t return any value:  select ete\_siteless\_key  from grid.vpn  where assigned\_vpn\_name =<vpn\_name> -- defect 58004 fix  and ete\_siteless\_key is not null |
|  | ete\_key\_source | <286284-US600815-US600161>  See End-to-End Key Search and Generation [ Icore Data Flow ]  ~~See above. ete\_key\_source should be blank/null when ete\_vpn\_key asset identifier is removed.~~ |
| </286278> |  |  |
|  | Region\_flag <302503> | Set this to ‘OOR’, if icore.service.serv\_name = ‘OPT-E-WAN’ and icore.service\_option.serv\_opt = ‘ASEoD’ |

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| VPN.vpn\_id | IPFR.vpn\_id |
| PVC.pvc\_id | IPFR.pvc\_id |
| PVC.pvc\_lsite\_id <or> PVC.pvc\_rsite\_id <take only the one for the “customer” side; see logic under:  Logic for identifying a “customer” side in the ICORE.PVC table> | PORT\_ASGMT.site\_id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |

**Data Blocking (1)**

Do not process asset records (VPN) where **ALL** records returned by the logic below are linked to a GDB.SERVICE\_TYPE having a record in GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM for Icore *(if at least one record does NOT have a “blocked” service type, then include that VPN record !)*:

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| VPN.vpn\_id | IPFR.vpn\_id |
| PVC.pvc\_id | IPFR.pvc\_id |
| PVC.pvc\_lsite\_id <or> PVC.pvc\_rsite\_id < take only the one for the “customer” side; see logic under:  Logic for identifying a “customer” side in the ICORE.PVC table> | SITE.site\_id |
| SERVICE\_ASGMT.site\_id | SITE.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |
| GDB.SERVICE\_TYPE\_NOTATION.service\_name | SERVICE.serv\_name |
| GDB.SERVICE\_TYPE\_NOTATION.id\_service\_type | GDB.SERVICE\_TYPE.id |
| GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM.id\_service\_type | GDB.SERVICE\_TYPE.id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| SERVICE\_TYPE\_BLOCKED\_SYSTEM | BLOCKED\_SYSTEM | 13609 |

*Note:*

*The BLOCKED\_SYSTEM constraint value is the MOTS-ID of the system.*

**Data Blocking (2)**

Do not process asset records where the special logic for blocking INSTAR items hold.

The “<input site id variable (dep. on current scenario)>” here is “PVC.pvc\_lsite\_id <or> PVC.pvc\_rsite\_id”.

See “Logic for excluding INSTAR items”.

Do not process asset records (VPN) where **ALL** records returned by the logic under “Data Blocking (2)” will be blocked *(if at least one record is NOT “blocked”, then include that VPN record !)*:

##### Logic for identifying a “customer” side in the ICORE.PVC table

The logic for identifying a “customer” side in the ICORE.PVC table is as follows:

(1) Check whether the “PVC.pvc\_rcust\_id” points to AT&T as the customer.  
If it does, then the “r” side values point to AT&T, and the “l” side values represent the “customer” side, and data needs to be taken from the “l” side values.

(2) If the above does not hold, then check whether the “PVC.pvc\_lcust\_id” points to AT&T as the customer.  
If it does, then the “l” side values point to AT&T, and the “r” side values represent the “customer” side, and data needs to be taken from the “r” side values.

(3) If both of the above does not hold, then both sides are owned by AT&T as the customer, and the corresponding PVC record needs to be dropped.

AT&T’s “customer.cust\_id” value is usually “395”, the corresponding “customer.cust\_name” value is “AT&T INTERSPAN INVENTORY”. (This value may change, so this needs to be re-verified regularly).

<BEGIN MVL-VPN-FIX.2015-06-26 Defect 52461>

***HLD-258491-GCP-FLOW-ICORE-103-A [Icore Data Flow into Asset VPN (Layer-3, MVL “Customer Network”)]***

The following section is about “Customer Network (Layer-3, MVL)” assets. These types of assets are identified by having an entry in the ICORE.VPN table and additional records in the ICORE.PVC table where the PVC is a so-called “MVL PVC”. VPNs that are taken into account here as “MVL VPNs” shall **-\*NOT\*-** be loaded if they have already been loaded a “standard” Layer-3 VPNs. When the data is checked for selectability, only the “customer” side data from the ICORE.PVC table must be taken into account. (Details to be found below and at other appropriate places.)

Implement the following:

Process the following Icore records into the GDB asset object and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

*NOTE: Exclude (joined) records where (under “Database traversals”) for an associated PVC* ***both*** *the “PVC.pvc\_lcust\_id” and the “PVC.pvc\_rcust\_id” value point to a “real” customer. (This is a “degenerated” case.) Also to be excluded are records where at both sides AT&T is the customer. So only records where at one side a real customer is located and at the other side AT&T is located shall be taken into account. The VPN value may still be added/loaded if other associated PVC records are “valid”, i.e. have AT&T at one end and a real customer at the other end.*

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET column** | **Processing comment** |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘ASSET’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| VPN.vpn\_id | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| - | id\_asset\_type | Referencing ASSET\_TYPE ‘CUSTOMER\_NETWORK’ |

Identifier (1): VPN ID

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘VPN\_ID\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_asset\_identifier | ASSET\_IDENTIFIER.id |
| - | id\_change\_tracking | ASSET\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘VPN\_ID\_IDENTIFIER’ |
| VPN.vpn\_id | Value |  |

Identifier (2): VPN Name (only insert if “VPN.vpn\_name” exists and is not empty or null)

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘VPN\_NAME\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_asset\_identifier | ASSET\_IDENTIFIER.id |
| - | id\_change\_tracking | ASSET\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘VPN\_NAME\_IDENTIFIER’ |
| VPN.vpn\_name | Value |  |

<286278>

Identifier (3): ETE\_VPN\_KEY

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE “E2E\_VPN\_KEY\_ASSET\_IDENTIFIER” |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_asset\_identifier | ASSET\_IDENTIFIER.id |
| - | id\_change\_tracking | ASSET\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE “E2E\_VPN\_KEY\_ASSET\_IDENTIFIER” |
| - | Value | <286284-US600815-US600161>  See End-to-End Key Search and Generation [ Icore Data Flow ]  ~~select e2e\_vpn\_key, ‘LPP’ as ete\_key\_source~~  ~~from icore.vpn~~  ~~where vpn\_id = <vpn\_id)~~  ~~or~~  ~~select ete\_vpn\_key, ‘USRP’ as ete\_key\_source~~  ~~from usrp.vpn~~  ~~where icore\_vpn\_id =<vpn\_id>~~  ~~and ete\_vpn\_key is not null~~  ~~or above query doesn’t return any value:~~  ~~select ete\_vpn\_key, ‘IEOD’ as ete\_key\_source, ete\_siteless\_key~~  ~~from grid.vpn~~  ~~where assigned\_vpn\_name =<vpn\_name> -- defect 58004 fix~~  ~~and ete\_vpn\_key is not null~~  ~~remove the same asset identifier (same ete\_vpn\_key) for other Customer Network assets.~~ |

</286278>

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_EXT\_CUSTOMER\_NETWORK column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| VPN.vpn\_id | network\_id |  |
| VPN.vpn\_name | network\_name |  |
|  | id\_network\_type | Referencing NETWORK\_TYPE ‘NT\_VPN’ |
| VPN.vpn\_type | id\_network\_sub\_type | (1) referencing NETWORK\_TYPE ‘**NST\_HUB**’ if  VPN.vpn\_type = ‘hub’;  (2) referencing NETWORK\_TYPE ‘**NST\_SPOKE**’ if  VPN.vpn\_type = ‘spoke’;  (3) referencing NETWORK\_TYPE ‘**NST\_HUB\_SPOKE**’ if  VPN.vpn\_type =  ('hub-spoke' or 'hub\_spoke');  (4) referencing NETWORK\_TYPE ‘**NST\_ANY\_TO\_ANY**’ if  VPN.vpn\_type =  (NULL or blank or 'any-to-any');  (5) referencing NETWORK\_TYPE ‘**NST\_UNDEFINED**’ if  VPN.vpn\_type =  <any other value>; |
| <286278> |  |  |
|  | ETE\_SITELESS\_KEY | select ete\_vpn\_siteless\_key  from usrp.vpn  where icore\_vpn\_id =<vpn\_id>  and ete\_siteless\_key is not null  or above query doesn’t return any value:  select ete\_siteless\_key  from grid.vpn  where assigned\_vpn\_name =<vpn\_name> -- defect 58004 fix  and ete\_siteless\_key is not null |
|  | ete\_key\_source | <286284-US600815-US600161>  See End-to-End Key Search and Generation [ Icore Data Flow ]  ~~See above. ete\_key\_source should be blank/null when ete\_vpn\_key asset identifier is removed.~~ |
| </286278> |  |  |

Database traversals (ICORE + INSTAR involved):

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| **INSTAR.**ENDPT2VPN.vpn\_id | **ICORE**.VPN.vpn\_id |
| **INSTAR**.ENDPT2VPN.endpoint\_id | **INSTAR**.IP\_PORT\_ASGMT.ip\_endpt |
| **INSTAR**.IP\_SERV\_ACC\_PT.serv\_acc\_pt\_id | **INSTAR**.IP\_PORT\_ASGMT.serv\_acc\_pt\_id |
| **INSTAR**.IP\_SERV\_ACC\_PT.icore\_pvc\_id | **ICORE**.NETWORK\_CONNECTION.pvc\_id |
| PVC.pvc\_id | **ICORE**.NETWORK\_CONNECTION.pvc\_id |
| PVC.pvc\_lsite\_id <or> PVC.pvc\_rsite\_id <take only the one for the “customer” side; see logic under:  Logic for identifying a “customer” side in the ICORE.PVC table> | PORT\_ASGMT.site\_id |
| **ICORE**.PVC\_EXTENSION.pvc\_id | **ICORE**.NETWORK\_CONNECTION.pvc\_id |

Database constraints (ICORE + INSTAR involved):

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| **INSTAR.**ENDPT2VPN.vpn\_id | Status | ‘ACTIVE’ |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| **ICORE**.PORT\_ASGMT | PORT\_STAT | IN-SERVICE |
| **ICORE**.PVC\_EXTENSION | MVL | ‘Y’ |

**From the MVL VPN data that is found by the above database traversals and constraints any VPN ID/record shall be excluded (i.e. shall not be loaded, because it will already have been loaded under the Layer-3 VPN data load !) that matches the database traversals and constraints below:**

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| VPN.vpn\_id | IPFR.vpn\_id |
| PVC.pvc\_id | IPFR.pvc\_id |
| PVC.pvc\_lsite\_id <or> PVC.pvc\_rsite\_id <take only the one for the “customer” side; see logic under:  Logic for identifying a “customer” side in the ICORE.PVC table> | PORT\_ASGMT.site\_id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |

**Data Blocking (1)**

Do not process asset records (VPN) where **ALL** records returned by the logic below are linked to a GDB.SERVICE\_TYPE having a record in GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM for Icore *(if at least one record does NOT have a “blocked” service type, then include that VPN record !)*:

Database traversals:

|  |  |  |
| --- | --- | --- |
| **From table.column** | | **To table.column** |
| **INSTAR.**ENDPT2VPN.vpn\_id | **ICORE**.VPN.vpn\_id | |
| **INSTAR**.ENDPT2VPN.endpoint\_id | **INSTAR**.IP\_PORT\_ASGMT.ip\_endpt | |
| **INSTAR**.IP\_SERV\_ACC\_PT.serv\_acc\_pt\_id | **INSTAR**.IP\_PORT\_ASGMT.serv\_acc\_pt\_id | |
| **INSTAR**.IP\_SERV\_ACC\_PT.icore\_pvc\_id | **ICORE**.NETWORK\_CONNECTION.pvc\_id | |
| PVC.pvc\_id | **ICORE**.NETWORK\_CONNECTION.pvc\_id | |
| PVC.pvc\_lsite\_id <or> PVC.pvc\_rsite\_id < take only the one for the “customer” side; see logic under:  Logic for identifying a “customer” side in the ICORE.PVC table> | | SITE.site\_id |
| SERVICE\_ASGMT.site\_id | | SITE.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | | SERVICE.serv\_id |
| GDB.SERVICE\_TYPE\_NOTATION.service\_name | | SERVICE.serv\_name |
| GDB.SERVICE\_TYPE\_NOTATION.id\_service\_type | | GDB.SERVICE\_TYPE.id |
| GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM.id\_service\_type | | GDB.SERVICE\_TYPE.id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| SERVICE\_TYPE\_BLOCKED\_SYSTEM | BLOCKED\_SYSTEM | 13609 |

*Note:*

*The BLOCKED\_SYSTEM constraint value is the MOTS-ID of the system.*

**Data Blocking (2)**

Do not process asset records where the special logic for blocking INSTAR items hold.

The “<input site id variable (dep. on current scenario)>” here is “PVC.pvc\_lsite\_id <or> PVC.pvc\_rsite\_id”.

See “Logic for excluding INSTAR items”.

Do not process asset records (VPN) where **ALL** records returned by the logic under “Data Blocking (2)” will be blocked *(if at least one record is NOT “blocked”, then include that VPN record !)*:

<END MVL-VPN-FIX.2015-06-26 Defect 52461>

#### HLD-254035-GCP-FLOW-ICORE-103-B [Icore Data Flow into Asset PVC (as VPN) (Layer-2 "Customer Network")]

The following section is about “Customer Network (Layer-2)” assets. These types of assets are identified by not having any entry in the ICORE.IPFR table (and therefore no entry in the ICORE.VPN table) and having a non-AT&T customer id value at both sides of the PVC table’s entry. In that way that record in the ICORE.PVC table is considered a very simple “Customer Network” object. When the data is checked for selectability, both sides from the ICORE.PVC table must be “customer” sides. (Details to be found below and at other appropriate places.)

Implement the following:

Process the following Icore records into the GDB asset object and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

*NOTE: Only include PVC records where for the given PVC.pvc\_id no record exists in IPFR for IPFR.pvc\_id* ***AND*** *where* ***both*** *the “PVC.pvc\_lcust\_id” and the “PVC.pvc\_rcust\_id” value point to a “real” customer* ***AND*** *where* ***both*** *“PORT\_ASGMT.port\_stat” (for “l” and “r”) have the value “IN-SERVICE”.*

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET column** | **Processing comment** |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘ASSET’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| PVC.pvc\_id | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| - | id\_asset\_type | Referencing ASSET\_TYPE ‘CUSTOMER\_NETWORK’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘PVC\_ID\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_asset\_identifier | ASSET\_IDENTIFIER.id |
| - | id\_change\_tracking | ASSET\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘PVC\_ID\_IDENTIFIER’ |
| PVC.pvc\_id | Value | *Note that there is no “real” identifier (like the “vpn name” for “layer-3” vpns) for these simple “Customer Network” objects, so we use the “pvc id” to make sure we have at least one identifier here.* |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_EXT\_CUSTOMER\_NETWORK column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| PVC.pvc\_id | network\_id |  |
| - | network\_name | Set to the database “NULL” value, as a reasonable “name” value does not exist here ! |
|  | id\_network\_type | Referencing NETWORK\_TYPE ‘NT\_PVC’ |
| <BEGIN 259118> |  |  |
| - | id\_network\_sub\_type | (1) referencing NETWORK\_TYPE ‘**NST\_ANY\_TO\_ANY**’ |
| <BEGIN 259118> |  |  |

Database traversals:

*(PVC “l” side)*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| PVC.pvc\_id | Not exists(IPFR.pvc\_id) |
| PVC.pvc\_lsite\_id | PORT\_ASGMT.site\_id |

Database constraints:

*(PVC “l” side)*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| PVC | PVC\_ID | Not exists(IPFR.pvc\_id) |
| PVC | PVC\_LCUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |

<see also logic under:   
Logic for identifying a “customer” side in the ICORE.PVC table>

Database traversals:

*(PVC “r” side)*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| PVC.pvc\_id | Not exists(IPFR.pvc\_id) |
| PVC.pvc\_rsite\_id | PORT\_ASGMT.site\_id |

Database constraints:

*(PVC “r” side)*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| PVC | PVC\_ID | Not exists(IPFR.pvc\_id) |
| PVC | PVC\_RCUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |

<see also logic under:   
Logic for identifying a “customer” side in the ICORE.PVC table>

**Data Blocking (1)**

Do not process asset records (PVC) where **ALL** records returned by the logic below are linked to a GDB.SERVICE\_TYPE having a record in GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM for Icore *(if at least one record does NOT have a “blocked” service type, then include that PVC record !)*:

Database traversals:

*(PVC “l” side)*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| PVC.pvc\_lsite\_id | SITE.site\_id |
| SERVICE\_ASGMT.site\_id | SITE.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |
| GDB.SERVICE\_TYPE\_NOTATION.service\_name | SERVICE.serv\_name |
| GDB.SERVICE\_TYPE\_NOTATION.id\_service\_type | GDB.SERVICE\_TYPE.id |
| GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM.id\_service\_type | GDB.SERVICE\_TYPE.id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| SERVICE\_TYPE\_BLOCKED\_SYSTEM | BLOCKED\_SYSTEM | 13609 |

Database traversals:

*(PVC “r” side)*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| PVC.pvc\_rsite\_id | SITE.site\_id |
| SERVICE\_ASGMT.site\_id | SITE.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |
| GDB.SERVICE\_TYPE\_NOTATION.service\_name | SERVICE.serv\_name |
| GDB.SERVICE\_TYPE\_NOTATION.id\_service\_type | GDB.SERVICE\_TYPE.id |
| GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM.id\_service\_type | GDB.SERVICE\_TYPE.id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| SERVICE\_TYPE\_BLOCKED\_SYSTEM | BLOCKED\_SYSTEM | 13609 |

*Note:*

*The BLOCKED\_SYSTEM constraint value is the MOTS-ID of the system.*

**Data Blocking (2)**

Do not process asset records where the special logic for blocking INSTAR items hold.

The “<input site id variable (dep. on current scenario)>” here is “PVC.pvc\_lsite\_id <or> PVC.pvc\_rsite\_id”.

See “Logic for excluding INSTAR items”.

Do not process asset records (PVC Layer-2 Customer Network) where **BOTH** sides (pvc\_lsite\_id and pvc\_rsite\_id) returned by the logic under “Data Blocking (2)” will be blocked *(if at least one side is NOT “blocked”, then include that* PVC Layer-2 Customer Network *record !)*:

#### HLD-254035-GCP-FLOW-ICORE-103-C [Icore Data Flow into Asset VPN (“Internet VLAN” “Customer Network”)]

The following section is about “Customer Network (“Internet VLAN”)” assets. These types of assets represent an “Internet VLAN” “Customer Network” which is just a place holder for a VPN in the “internet”. These types of assets are identified by -\*NOT\*- having an entry in the ICORE.VPN table and ICORE.IPFR table, but having a record in the ICORE.PVC table that is visible via the ICORE.CUSTOMER\_NETWORK\_INET\_VLAN view.

There ia a 1-to-1 relationship between an “Internet VLAN” ICORE.PVC record and the associated “Internet VLAN” “Customer Network” (In fact in ICORE no such “Internet VLAN” “Customer Network” data exists; it is derived from the corresponding ICORE.PVC record !)

When the data is checked for selectability, only the “customer” side data from the ICORE.PVC table must be taken into account. (Details to be found below and at other appropriate places.)

Implement the following:

Process the following Icore records into the GDB asset object and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

*NOTE: Exclude records where (under “Database traversals”) for an associated PVC* ***both*** *the “PVC.pvc\_lcust\_id” and the “PVC.pvc\_rcust\_id” value point to a “real” customer. (This is a “degenerated” case.)*

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET column** | **Processing comment** |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘ASSET’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| CUSTOMER\_NETWORK\_INET\_VLAN.vpn\_id + CUSTOMER\_NETWORK\_INET\_VLAN.pvc\_id + CUSTOMER\_NETWORK\_INET\_VLAN.site\_id | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| - | id\_asset\_type | Referencing ASSET\_TYPE ‘CUSTOMER\_NETWORK’ |

“Internet VLAN” “Customer Network” assets do not have valid identifiers, so no entries will be created in respect to identifiers (GDB.ASSET\_IDENTIFIER and GDB.ASSET\_IDENTIFIER\_VALUE).

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_EXT\_CUSTOMER\_NETWORK column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| CUSTOMER\_NETWORK\_INET\_VLAN.vpn\_id | network\_id |  |
| - | network\_name | “Internet VLAN” |
|  | id\_network\_type | Referencing NETWORK\_TYPE ‘NT\_INTERNET’ |
| <BEGIN 259118> |  |  |
| - | id\_network\_sub\_type | (1) referencing NETWORK\_TYPE ‘**NST\_ANY\_TO\_ANY**’ |
| <BEGIN 259118> |  |  |

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| PVC.pvc\_id | CUSTOMER\_NETWORK\_INET\_VLAN.pvc\_id |
| SITE.site\_id | CUSTOMER\_NETWORK\_INET\_VLAN.site\_id |
| SITE.site\_id | PORT\_ASGMT.site\_id |

Database constraints:

*(PVC “l” side [“customer/site” on the “l” side])*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| CUSTOMER\_NETWORK\_INET\_VLAN | SIDE | ‘local’ |
| CUSTOMER\_NETWORK\_INET\_VLAN | CUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) |
| PVC | PVC\_RCUST\_ID | = 395 (i.e. is “AT&T” customer id) |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |

Database constraints:

*(PVC “r” side [“customer/site” on the “r” side])*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| CUSTOMER\_NETWORK\_INET\_VLAN | SIDE | ‘remote’ |
| CUSTOMER\_NETWORK\_INET\_VLAN | CUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) |
| PVC | PVC\_LCUST\_ID | = 395 (i.e. is “AT&T” customer id) |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |

**Data Blocking (1)**

Do not process asset records (VPN) where records returned by the logic below are linked to a GDB.SERVICE\_TYPE having a record in GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM for Icore:

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| CUSTOMER\_NETWORK\_INET\_VLAN.pvc\_id | PVC.pvc\_id |
| CUSTOMER\_NETWORK\_INET\_VLAN.site\_id | SITE.site\_id |
| SERVICE\_ASGMT.site\_id | SITE.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |
| GDB.SERVICE\_TYPE\_NOTATION.service\_name | SERVICE.serv\_name |
| GDB.SERVICE\_TYPE\_NOTATION.id\_service\_type | GDB.SERVICE\_TYPE.id |
| GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM.id\_service\_type | GDB.SERVICE\_TYPE.id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| SERVICE\_TYPE\_BLOCKED\_SYSTEM | BLOCKED\_SYSTEM | 13609 |

*Note:*

*The BLOCKED\_SYSTEM constraint value is the MOTS-ID of the system.*

**Data Blocking (2)**

Do not process asset records where the special logic for blocking INSTAR items hold.

The “<input site id variable (dep. on current scenario)>” here is “PVC.pvc\_lsite\_id <or> PVC.pvc\_rsite\_id”.

See “Logic for excluding INSTAR items”.

Do not process asset records (VPN) where records returned by the logic under “Data Blocking (2)” will be blocked.

#### HLD-254035-GCP-FLOW-ICORE-104-A [Icore Data Flow into Asset PVC (Layer-3 “Network Connection”)]

The following section is about “Network Connections (Layer-3)” assets. These types of assets are identified by having an entry in the ICORE IPFR table. The ICORE.PVC table contains so-called “local (l)” and “remote (r)” side data, and each of those two sides is considered a “Network Connection” (or sometimes called a network connection “leg”). When the data is checked for selectability, only the “customer” side data from the ICORE.PVC table must be taken into account. (Details to be found below and at other appropriate places.)

*NOTE: For the following we have to do with data based on a PVC record and either a PVC.pvc\_lsite\_id or a PVC.pvc\_rsite\_id. For the Enterprise Key Translation (EKT) this is “abstracted away” by making use of the “NETWORK\_CONNECTION” view.*

Implement the following:

Process the following Icore records into the GDB asset object and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

*NOTE: Include only records where for a PVC a PVC.pvc\_id value exists in the IPFR table under IPFR.pvc\_id.*

*NOTE: Include only records where for a PVC only one “cust\_id” value (either “PVC.pvc\_lcust\_id” or “PVC.pvc\_rcust\_id”) points to a “real” customer.*

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET column** | **Processing comment** |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘ASSET’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| *[“customer” on the “l” side]* PVC.pvc\_id + PVC.pvc\_lsite\_id;  *[“customer” on the “r” side]* PVC.pvc\_id + PVC.pvc\_rsite\_id;  *NOTE: For the EKT use the following corresponding values:*  NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| *-* | id\_asset\_type | Referencing ASSET\_TYPE ‘NETWORK\_CONNECTION’ |

*~~NOTE: There is no data item that could be used as an “asset identifier” here, so we do not add that type of data here !~~*

|  |  |  |
| --- | --- | --- |
| **~~Icore table.column~~** | **~~GDB ASSET\_IDENTIFIER column~~** | **~~Processing comment~~** |
| ~~-~~ | ~~id\_asset~~ | ~~<n/a>~~ |
| ~~-~~ | ~~id\_change\_tracking~~ | ~~<n/a>~~ |
| ~~-~~ | ~~id\_identifier\_type~~ | ~~<n/a>~~ |

|  |  |  |
| --- | --- | --- |
| **~~Icore table.column~~** | **~~GDB ASSET\_IDENTIFIER\_VALUE column~~** | **~~Processing comment~~** |
| ~~-~~ | ~~id\_asset\_identifier~~ | ~~<n/a>~~ |
| ~~-~~ | ~~id\_change\_tracking~~ | ~~<n/a>~~ |
| ~~-~~ | ~~id\_identifier\_type~~ | ~~<n/a>~~ |
| ~~<n/a>~~ | ~~Value~~ | ~~<n/a>~~ |

<BEGIN 259118>

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘PVC\_ID\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_asset\_identifier | ASSET\_IDENTIFIER.id |
| - | id\_change\_tracking | ASSET\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘PVC\_ID\_IDENTIFIER’ |
| NETWORK\_CONNECTION.pvc\_id | Value |  |

<END 259118>

<286278><REL1602>

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘E2E\_SERVICE\_CONN\_KEY\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_asset\_identifier | ASSET\_IDENTIFIER.id |
| - | id\_change\_tracking | ASSET\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘E2E\_SERVICE\_CONN\_KEY\_IDENTIFIER’ |
| - | Value | <286284-US600815-US600161>  See End-to-End Key Search and Generation [ Icore Data Flow ]  ~~SELECT external\_key\_value AS ete\_service\_connection\_key, ‘LPP’ as ete\_key\_source~~  ~~FROM service\_user.service\_connection\_mapping~~  ~~WHERE internal\_key\_type LIKE 'Icore Pvc Id'~~  ~~AND internal\_key\_value = <pvc\_id>~~  ~~or~~  ~~select ete\_svc\_conn\_key, ‘USRP’ as ete\_key\_source~~  ~~from usrp.pvc~~  ~~where icore\_pvc\_id =<pvc\_id>~~  ~~and ete\_svc\_conn\_key is not null~~  ~~or~~  ~~select ete\_service\_connection\_key, ‘IEOD’ as ete\_key\_source~~  ~~from grid.service\_connection~~  ~~where icore\_pvc\_id = <pvc\_id>~~  ~~<270198g>~~  ~~remove the same asset identifier (same ete\_svc\_conn\_key) for other Network Connection assets.~~ |

</286278>

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_EXT\_NETWORK\_CONNECTION column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| NETWORK\_CONNECTION.pvc\_id | network\_connection\_id | (to be converted to VARCHAR2) |
| NETWORK\_CONNECTION.contrcir | Speed |  |
|  | id\_network\_subtype | (1)  IPFR.vpn\_site\_type = ‘hub’: Referencing NETWORK\_TYPE ‘**NST\_HUB**’  (2)  IPFR.vpn\_site\_type = ‘spoke’: Referencing NETWORK\_TYPE ‘**NST\_SPOKE**’  (3)  any other value in IPFR.vpn\_site\_type:  Referencing NETWORK\_TYPE ‘**NST\_ANY\_TO\_ANY**’ |
| <286278> |  |  |
| NETWORK\_TYPE.id | Id\_network\_type | Use the NETWORK\_CONNECTION.id query usrp.pvc on icore\_pvc\_id, and retrieve usrp.pvc.pvc\_type:  If pvc\_type = 33 or 13, referencing NETWORK\_TYPE ‘NT\_VPN’  If pvc\_type = 36, referencing NETWORK\_TYPE ‘NT\_INTERNET’  If pvc\_type = 37, referencing NETWORK\_TYPE ‘NT\_TUNNEL’,  Or  Use the NETWORK\_CONNECTION.id query grid.service\_connection on icore\_pvc\_id, and retrieve grid.service\_connection.service\_connection\_type:  If service\_connection\_type = ‘VPN’, referencing NETWORK\_TYPE ‘NT\_VPN’  If service\_connection\_type = ‘INTERNET’, referencing NETWORK\_TYPE ‘NT\_INTERNET’  If pvc\_type = ‘TUNNEL’, referencing NETWORK\_TYPE ‘NT\_TUNNEL’,  Else, set NULL. |
|  | ETE\_SERVICE\_CONNECTION\_KEY <REL1602> | <286284-US600815-US600161>  See End-to-End Key Search and Generation [ Icore Data Flow ]  ~~ete\_svc\_conn\_key retrieved above.~~  ~~<270198g>~~  ~~If there is any other asset\_ext\_network\_connection records with same ete\_service\_connection\_key, wipe out those ete\_service\_connection\_key values to blank~~ |
|  | ete\_key\_source | <286284-US600815-US600161>  See End-to-End Key Search and Generation [ Icore Data Flow ]  ~~See above. ete\_key\_source should be blank/null when ete\_svc\_conn\_key asset identifier is removed.~~ |
| </286278> |  |  |
| IPFR.DHV\_INDICATOR <292589-US847778> | Dhv\_ind |  |
| PVC\_EXTENSION.mvl | MVL\_IND | <304329-US10770> |
|  | Region\_flag <302503> | Set this to ‘OOR’, if icore.service.serv\_name = ‘OPT-E-WAN’ and icore.service\_option.serv\_opt = ‘ASEoD’ |

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| PVC.pvc\_id | NETWORK\_CONNECTION.pvc\_id |
| PVC.pvc\_id | IPFR.pvc\_id |
| NETWORK\_CONNECTION.site\_id | PORT\_ASGMT.site\_id |
| PVC.pvc\_id | PVC\_EXTENSION.pvc\_id (outer join)  <304329-US10770> |

Database constraints:

*(PVC “l” side [“customer/site” on the “l” side])*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| NETWORK\_CONNECTION | SIDE | ‘local’ |
| NETWORK\_CONNECTION | CUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) |
| PVC | PVC\_RCUST\_ID | = 395 (i.e. is “AT&T” customer id) |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |

Database constraints:

*(PVC “r” side [“customer/site” on the “r” side])*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| NETWORK\_CONNECTION | SIDE | ‘remote’ |
| NETWORK\_CONNECTION | CUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) |
| PVC | PVC\_LCUST\_ID | = 395 (i.e. is “AT&T” customer id) |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |

**Data Blocking (1)**

Do not process asset records (PVC) where the record returned by the logic below is linked to a GDB.SERVICE\_TYPE having a record in GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM for Icore:

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| PVC.pvc\_id | NETWORK\_CONNECTION.pvc\_id |
| PVC.pvc\_id | IPFR.pvc\_id |
| NETWORK\_CONNECTION.site\_id | SITE.site\_id |
| SERVICE\_ASGMT.site\_id | SITE.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |
| GDB.SERVICE\_TYPE\_NOTATION.service\_name | SERVICE.serv\_name |
| GDB.SERVICE\_TYPE\_NOTATION.id\_service\_type | GDB.SERVICE\_TYPE.id |
| GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM.id\_service\_type | GDB.SERVICE\_TYPE.id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| SERVICE\_TYPE\_BLOCKED\_SYSTEM | BLOCKED\_SYSTEM | 13609 |

*Note:*

*The BLOCKED\_SYSTEM constraint value is the MOTS-ID of the system.*

**Data Blocking (2)**

Do not process asset records where the special logic for blocking INSTAR items hold.

The “<input site id variable (dep. on current scenario)>” here is “NETWORK\_CONNECTION.site\_id”.

See “Logic for excluding INSTAR items”.

#### HLD-254035-GCP-FLOW-ICORE-104-B [Icore Data Flow into Asset PVC (Layer-2 "Network Connection")]

The following section is about “Network Connection (Layer-2)” assets. These types of assets are identified by not having any entry in the ICORE.IPFR table (and therefore no entry in the ICORE.VPN table) and having a non-AT&T customer id value at both sides. In that way that record in the ICORE.PVC table is considered a very simple “Customer Network” object. The ICORE.PVC table contains so-called “local (l)” and “remote (r)” side data, and each of those two sides is considered a “Network Connection” (or sometimes called a network connection “leg”). When the data is checked for selectability, both sides from the ICORE.PVC table must be “customer” sides. (Details to be found below and at other appropriate places.)

*NOTE: For the following we have to do with data based on a PVC record and either a PVC.pvc\_lsite\_id or a PVC.pvc\_rsite\_id. For the Enterprise Key Translation (EKT) this is “abstracted away” by making use of the “NETWORK\_CONNECTION” view.*

Implement the following:

Process the following Icore records into the GDB asset object and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

*NOTE: Only include PVC records where for the given PVC.pvc\_id no record exists in IPFR for IPFR.pvc\_id* ***AND*** *where* ***both*** *the “PVC.pvc\_lcust\_id” and the “PVC.pvc\_rcust\_id” value point to a “real” customer.*

*NOTE: For the following always 2 records are processed: The one for the “l” side (Record 1) and the one for the “r” side (Record 2). However, this is transparently handled by the two different “Database constraints” (one for the “l” side and one for the “r” side) below.*

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET column** | **Processing comment** |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘ASSET’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| *[“Record 1” (on the “l” side)]* PVC.pvc\_id + PVC.pvc\_lsite\_id;  *[“Record 2” (on the “r” side)]* PVC.pvc\_id + PVC.pvc\_rsite\_id;   *NOTE: For the EKT use the following corresponding values:*  NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| *-* | id\_asset\_type | Referencing ASSET\_TYPE ‘NETWORK\_CONNECTION’ |

*~~NOTE: There is no data item that could be used as an “asset identifier” here, so we do not add that type of data here !~~*

|  |  |  |
| --- | --- | --- |
| **~~Icore table.column~~** | **~~GDB ASSET\_IDENTIFIER column~~** | **~~Processing comment~~** |
| ~~-~~ | ~~id\_asset~~ | ~~<n/a>~~ |
| ~~-~~ | ~~id\_change\_tracking~~ | ~~<n/a>~~ |
| ~~-~~ | ~~id\_identifier\_type~~ | ~~<n/a>~~ |

|  |  |  |
| --- | --- | --- |
| **~~Icore table.column~~** | **~~GDB ASSET\_IDENTIFIER\_VALUE column~~** | **~~Processing comment~~** |
| ~~-~~ | ~~id\_asset\_identifier~~ | ~~<n/a>~~ |
| ~~-~~ | ~~id\_change\_tracking~~ | ~~<n/a>~~ |
| ~~-~~ | ~~id\_identifier\_type~~ | ~~<n/a>~~ |
| ~~<n/a>~~ | ~~Value~~ | ~~<n/a>~~ |

<BEGIN 259118>

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘PVC\_ID\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_asset\_identifier | ASSET\_IDENTIFIER.id |
| - | id\_change\_tracking | ASSET\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘PVC\_ID\_IDENTIFIER’ |
| NETWORK\_CONNECTION.pvc\_id | Value |  |

<END 259118>

<286278><REL1602>

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘E2E\_SERVICE\_CONN\_KEY\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_asset\_identifier | ASSET\_IDENTIFIER.id |
| - | id\_change\_tracking | ASSET\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘E2E\_SERVICE\_CONN\_KEY\_IDENTIFIER’ |
| - | Value | <286284-US600815-US600161>  See End-to-End Key Search and Generation [ Icore Data Flow ]  ~~SELECT external\_key\_value AS ete\_service\_connection\_key, ‘LPP’ as ete\_key\_source~~  ~~FROM service\_user.service\_connection\_mapping~~  ~~WHERE internal\_key\_type LIKE 'Icore Pvc Id'~~  ~~AND internal\_key\_value = <pvc\_id>~~  ~~or~~  ~~select ete\_svc\_conn\_key, ‘USRP’ as ete\_key\_source~~  ~~from usrp.pvc~~  ~~where icore\_pvc\_id =<pvc\_id>~~  ~~and ete\_svc\_conn\_key is not null~~  ~~or~~  ~~select ete\_service\_connection\_key, ‘IEOD’ as ete\_key\_source~~  ~~from grid.service\_connection~~  ~~where icore\_pvc\_id = <pvc\_id>~~  ~~<270198g>~~  ~~remove the same asset identifier (same ete\_svc\_conn\_key) for other Network Connection assets.~~ |

</286278>

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_EXT\_NETWORK\_CONNECTION column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| NETWORK\_CONNECTION.pvc\_id | network\_connection\_id | (to be converted to VARCHAR2) |
| NETWORK\_CONNECTION.contrcir | Speed |  |
|  | id\_network\_subtype | Referencing NETWORK\_TYPE ‘**NST\_POINT\_TO\_POINT**’ |
| <286278> |  |  |
| NETWORK\_TYPE.id | Id\_network\_type | Use the NETWORK\_CONNECTION.id query usrp.pvc on icore\_pvc\_id, and retrieve usrp.pvc.pvc\_type:  If pvc\_type = 33 or 13, referencing NETWORK\_TYPE ‘NT\_VPN’  If pvc\_type = 36, referencing NETWORK\_TYPE ‘NT\_INTERNET’  If pvc\_type = 37, referencing NETWORK\_TYPE ‘NT\_TUNNEL’,  Or  Use the NETWORK\_CONNECTION.id query grid.service\_connection on icore\_pvc\_id, and retrieve grid.service\_connection.service\_connection\_type:  If service\_connection\_type = ‘VPN’, referencing NETWORK\_TYPE ‘NT\_VPN’  If service\_connection\_type = ‘INTERNET’, referencing NETWORK\_TYPE ‘NT\_INTERNET’  If pvc\_type = ‘TUNNEL’, referencing NETWORK\_TYPE ‘NT\_TUNNEL’,  Else, set NULL. |
|  | ETE\_SERVICE\_CONNECTION\_KEY <REL1602> | <286284-US600815-US600161>  See End-to-End Key Search and Generation [ Icore Data Flow ]  ~~ete\_svc\_conn\_key retrieved above.~~  ~~<270198g>~~  ~~If there is any other asset\_ext\_network\_connection records with same ete\_service\_connection\_key, wipe out those ete\_service\_connection\_key values to blank~~ |
|  | ete\_key\_source | <286284-US600815-US600161>  See End-to-End Key Search and Generation [ Icore Data Flow ]  ~~See above. ete\_key\_source should be blank/null when ete\_svc\_conn\_key asset identifier is removed.~~ |
| </286278> |  |  |
| IPFR.DHV\_INDICATOR <292589-US847778> | Dhv\_ind |  |
| PVC\_EXTENSION.mvl | MVL\_IND | <304329-US10770> |
|  | Region\_flag <302503> | Set this to ‘OOR’, if icore.service.serv\_name = ‘OPT-E-WAN’ and icore.service\_option.serv\_opt = ‘ASEoD’ |

Database traversals:

*(generic, i.e for both sides)*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| PVC.pvc\_id | NETWORK\_CONNECTION.pvc\_id |
| PVC.pvc\_id | Not exists(IPFR.pvc\_id) |
| NETWORK\_CONNECTION.site\_id | PORT\_ASGMT.site\_id |
| PVC.pvc\_id | PVC\_EXTENSION.pvc\_id (outer join)  <304329-US10770> |

Database constraints:

*(PVC “l” side)*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| NETWORK\_CONNECTION | SIDE | ‘local’ |
| NETWORK\_CONNECTION | CUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) |
| PVC | PVC\_**R**CUST\_ID | != 395 (i.e. “**R**” side is “non-AT&T/real customer” customer id, too) |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |

Database constraints:

*(PVC “r” side)*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| NETWORK\_CONNECTION | SIDE | ‘remote’ |
| NETWORK\_CONNECTION | CUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) |
| PVC | PVC\_**L**CUST\_ID | != 395 (i.e. “**L**” side is “non-AT&T/real customer” customer id, too) |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |

<see also logic under:   
Logic for identifying a “customer” side in the ICORE.PVC table>

**Data Blocking (1)**

Do not process asset records (PVC) where the record returned by the logic below is linked to a GDB.SERVICE\_TYPE having a record in GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM for Icore:

Database traversals:

*(generic, i.e for both sides)*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| NETWORK\_CONNECTION.site\_id | SITE.site\_id |
| SERVICE\_ASGMT.site\_id | SITE.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |
| GDB.SERVICE\_TYPE\_NOTATION.service\_name | SERVICE.serv\_name |
| GDB.SERVICE\_TYPE\_NOTATION.id\_service\_type | GDB.SERVICE\_TYPE.id |
| GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM.id\_service\_type | GDB.SERVICE\_TYPE.id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| SERVICE\_TYPE\_BLOCKED\_SYSTEM | BLOCKED\_SYSTEM | 13609 |

*Note:*

*The BLOCKED\_SYSTEM constraint value is the MOTS-ID of the system.*

**Data Blocking (2)**

Do not process asset records where the special logic for blocking INSTAR items hold.

The “<input site id variable (dep. on current scenario)>” here is “NETWORK\_CONNECTION.site\_id”.

See “Logic for excluding INSTAR items”.

#### HLD-254035-GCP-FLOW-ICORE-104-C [Icore Data Flow into Asset PVC (“Internet VLAN” “Network Connection”)]

The following section is about “Internet VLAN” “Network Connections” assets. These types of assets are identified by -\*NOT\*- having an entry in the ICORE.VPN table and ICORE.IPFR table, but having a record in the ICORE.PVC table that is visible via the ICORE.NETWORK\_CONNECTION\_INET\_VLAN view.

The ICORE.PVC table contains so-called “local (l)” and “remote (r)” side data, and each of those two sides is considered a “Network Connection” (or sometimes called a network connection “leg”). When the data is checked for selectability, only the “customer” side data from the ICORE.PVC table must be taken into account. (Details to be found below and at other appropriate places.)

*NOTE: For the following we have to do with data based on a PVC record and either a PVC.pvc\_lsite\_id or a PVC.pvc\_rsite\_id. For the Enterprise Key Translation (EKT) this is “abstracted away” by making use of the “NETWORK\_CONNECTION\_INET\_VLAN” view.*

When the data is checked for selectability, only the “customer” side data from the ICORE.PVC table must be taken into account. (Details to be found below and at other appropriate places.)

Implement the following:

Process the following Icore records into the GDB asset object and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

*NOTE: Include only records where for a PVC only one “cust\_id” value (either “PVC.pvc\_lcust\_id” or “PVC.pvc\_rcust\_id”) points to a “real” customer.*

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET column** | **Processing comment** |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘ASSET’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| *[“customer” on the “l” side]* PVC.pvc\_id + PVC.pvc\_lsite\_id;  *[“customer” on the “r” side]* PVC.pvc\_id + PVC.pvc\_rsite\_id;  *NOTE: For the EKT use the following corresponding values:*  NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id; | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| *-* | id\_asset\_type | Referencing ASSET\_TYPE ‘NETWORK\_CONNECTION’ |

*NOTE: There is no data item that could be used as an “asset identifier” here, so we do not add that type of data here !*

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER column** | **Processing comment** |
| - | id\_asset | <n/a> |
| - | id\_change\_tracking | <n/a> |
| - | id\_identifier\_type | <n/a> |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_asset\_identifier | <n/a> |
| - | id\_change\_tracking | <n/a> |
| - | id\_identifier\_type | <n/a> |
| <n/a> | Value | <n/a> |

<BEGIN 259118>

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘PVC\_ID\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_asset\_identifier | ASSET\_IDENTIFIER.id |
| - | id\_change\_tracking | ASSET\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘PVC\_ID\_IDENTIFIER’ |
| NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id | Value |  |

<END 259118>

<286278><REL1602>

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘E2E\_SERVICE\_CONN\_KEY\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_asset\_identifier | ASSET\_IDENTIFIER.id |
| - | id\_change\_tracking | ASSET\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘E2E\_SERVICE\_CONN\_KEY\_IDENTIFIER’ |
| - | Value | <286284-US600815-US600161>  See End-to-End Key Search and Generation [ Icore Data Flow ]  ~~SELECT external\_key\_value AS ete\_service\_connection\_key~~  ~~FROM service\_user.service\_connection\_mapping~~  ~~WHERE internal\_key\_type LIKE 'Icore Pvc Id'~~  ~~AND internal\_key\_value = <pvc\_id>~~  ~~select ete\_svc\_conn\_key~~  ~~from usrp.pvc~~  ~~where icore\_pvc\_id =<pvc\_id>~~  ~~and ete\_svc\_conn\_key is not null~~  ~~or~~  ~~select ete\_service\_connection\_key~~  ~~from grid.service\_connection~~  ~~where icore\_pvc\_id = <pvc\_id>~~  ~~<270198g>~~  ~~remove the same asset identifier (same ete\_svc\_conn\_key) for other Network Connection assets.~~ |

</286278>

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_EXT\_NETWORK\_CONNECTION column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id | network\_connection\_id | (to be converted to VARCHAR2) |
| NETWORK\_CONNECTION\_INET\_VLAN.contrcir | speed |  |
|  | id\_network\_subtype | Referencing NETWORK\_TYPE ‘**NST\_INTERNET**’ |
| <286278> |  |  |
| NETWORK\_TYPE.id | Id\_network\_type | Use the NETWORK\_CONNECTION.id query usrp.pvc on icore\_pvc\_id, and retrieve usrp.pvc.pvc\_type:  If pvc\_type = 33 or 13, referencing NETWORK\_TYPE ‘NT\_VPN’  If pvc\_type = 36, referencing NETWORK\_TYPE ‘NT\_INTERNET’  If pvc\_type = 37, referencing NETWORK\_TYPE ‘NT\_TUNNEL’,  Or  Use the NETWORK\_CONNECTION.id query grid.service\_connection on icore\_pvc\_id, and retrieve grid.service\_connection.service\_connection\_type:  If service\_connection\_type = ‘VPN’, referencing NETWORK\_TYPE ‘NT\_VPN’  If service\_connection\_type = ‘INTERNET’, referencing NETWORK\_TYPE ‘NT\_INTERNET’  If pvc\_type = ‘TUNNEL’, referencing NETWORK\_TYPE ‘NT\_TUNNEL’,  Else, set NULL. |
|  | ETE\_SERVICE\_CONNECTION\_KEY <REL1602> | <286284-US600815-US600161>  See End-to-End Key Search and Generation [ Icore Data Flow ]  ~~ete\_svc\_conn\_key retrieved above.~~  ~~<270198g>~~  ~~If there is any other asset\_ext\_network\_connection records with same ete\_service\_connection\_key, wipe out those ete\_service\_connection\_key values to blank~~ |
|  | ete\_key\_source | <286284-US600815-US600161>  See End-to-End Key Search and Generation [ Icore Data Flow ]  ~~See above. ete\_key\_source should be blank/null when ete\_svc\_conn\_key asset identifier is removed.~~ |
| </286278> |  |  |
| PVC\_EXTENSION.mvl | MVL\_IND | <304329-US10770> |

Database traversals:

*(generic, i.e for both sides, if applicable for the corr. side)*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id | PVC.pvc\_id |
| NETWORK\_CONNECTION\_INET\_VLAN.site\_id | PORT\_ASGMT.site\_id |
| PVC.pvc\_id | PVC\_EXTENSION.pvc\_id (outer join)  <304329-US10770> |

Database constraints:

*(PVC “l” side [“customer/site” on the “l” side])*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| NETWORK\_CONNECTION\_INET\_VLAN | SIDE | ‘local’ |
| NETWORK\_CONNECTION\_INET\_VLAN | CUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) |
| PVC | PVC\_**R**CUST\_ID | = 395 (i.e. “**R**” side is “AT&T” customer id) |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |

Database constraints:

*(PVC “r” side [“customer/site” on the “r” side])*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| NETWORK\_CONNECTION\_INET\_VLAN | SIDE | ‘remote’ |
| NETWORK\_CONNECTION\_INET\_VLAN | CUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) |
| PVC | PVC\_**L**CUST\_ID | = 395 (i.e. “**L**” side is “AT&T” customer id) |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |

**Data Blocking (1)**

Do not process asset records (PVC) where the record returned by the logic below is linked to a GDB.SERVICE\_TYPE having a record in GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM for Icore:

Database traversals:

*(generic, i.e for both sides, if applicable for the corr. side)*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id | PVC.pvc\_id |
| NETWORK\_CONNECTION\_INET\_VLAN.site\_id | SITE.site\_id |
| SERVICE\_ASGMT.site\_id | SITE.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |
| GDB.SERVICE\_TYPE\_NOTATION.service\_name | SERVICE.serv\_name |
| GDB.SERVICE\_TYPE\_NOTATION.id\_service\_type | GDB.SERVICE\_TYPE.id |
| GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM.id\_service\_type | GDB.SERVICE\_TYPE.id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| SERVICE\_TYPE\_BLOCKED\_SYSTEM | BLOCKED\_SYSTEM | 13609 |

*Note:*

*The BLOCKED\_SYSTEM constraint value is the MOTS-ID of the system.*

**Data Blocking (2)**

Do not process asset records where the special logic for blocking INSTAR items hold.

The “<input site id variable (dep. on current scenario)>” here is “NETWORK\_CONNECTION\_INET\_VLAN.site\_id”.

See “Logic for excluding INSTAR items”.

#### HLD-293772-CR160298-GCP-FLOW-USRP-104-D [Usrp Data Flow into Asset PVC (“Tunnel Vlan” “Network Connection”)]

The following section is about “Tunnel Service Connection” “Network Connections” assets. These types of assets are identified by usrp.pvc\_type =37.

Implement the following:

Process the following USRP records into the GDB asset object and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **USRP table.column** | **GDB ASSET column** | **Processing comment** |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘ASSET’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| *usrp.tunnel\_vlan\_vw.icore\_pvc\_id* | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| *-* | id\_asset\_type | Referencing ASSET\_TYPE ‘NETWORK\_CONNECTION’ |

286278911818nel\_vlan\_vw.icore\_pvc\_idn USRP replication schema:tion as below: or create new address\_notation records individual

Only Create asset\_identifier ‘E2E\_SERVICE\_CONNECTION\_KEY\_IDENTIFIER’

|  |  |  |
| --- | --- | --- |
| **USRP table.column** | **GDB ASSET\_IDENTIFIER column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘E2E\_SERVICE\_CONN\_KEY\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **USRP table.column** | **GDB ASSET\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_asset\_identifier | ASSET\_IDENTIFIER.id |
| - | id\_change\_tracking | ASSET\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘E2E\_SERVICE\_CONN\_KEY\_IDENTIFIER’ |
| ~~usrp.tunnel\_vlan\_vw.ete\_svc\_conn\_key~~ | Value | See End-to-End Key Search and Generation [ Icore Data Flow ] |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSET\_EXT\_NETWORK\_CONNECTION column** | **Processing comment** |
| - | id\_asset | ASSET.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| usrp. tunnel\_vlan\_vw.icore\_pvc\_id | network\_connection\_id | (to be converted to VARCHAR2) |
| usrp.tunnel\_vlan\_vw.cir | Speed |  |
|  | id\_network\_subtype | Referencing NETWORK\_TYPE ‘**NST\_TUNNEL**’ |
|  | Id\_network\_type | Referencing NETWORK\_TYPE ‘**NT\_TUNNEL**’ |
|  | ETE\_SERVICE\_CONNECTION\_KEY | End-to-End Key Search and Generation [ Icore Data Flow ]  ete\_svc\_conn\_key retrieved above. |
|  | ete\_key\_source | End-to-End Key Search and Generation [ Icore Data Flow ] |

<BEGIN: Port Load Corrections>

#### HLD-254035-GCP-FLOW-ICORE-110-A [Icore Data Flow into Phy Port SITE (Layer-3 “Phy Port”)]

and

#### HLD-254035-GCP-FLOW-ICORE-110-B [Icore Data Flow into Phy Port SITE (Layer-2 “Phy Port”)]

have been merged into the following requirement.

<END: Port Load Corrections>

#### HLD-254035-GCP-FLOW-ICORE-110-A [Icore Data Flow into Phy Port SITE (“Phy Port”)]

This section is on loading physical port data from ICORE. Physical ports are an object type of their own in GDB and are not classified under the “Asset” object type. In the context of physical port data the ICORE CUST\_ACCESS table plays a central role for identifying the physical port data via the “site\_id” data element. The base table for CE port data is the ICORE SITE table; the base table for PE ports is the ICORE PORT\_ASGMT table.

For physical ports the requirement is that they are associated to “Access Circuits”, i.e. ICORE.cust\_access. As we use ICORE.site.site\_id, we can ignore the table ICORE.cust\_access\_ckt, as it will have the same “site\_id” entry as any corresponding row from ICORE.cust\_access.

*NOTE: In respect to the “****Enterprise Key Translation (EKT)****” we use (for CE ports) the ICORE.SITE.site\_id value a second time here, so care needs to be taken given this special situation.*

*NOTE: This requirement needs to be executed after requirement “HLD-254035-GCP-FLOW-ICORE-101 [Icore Data Flow into Site]” has been completed, so the value for “id\_site\_inventory\_site\_rep” will already be available.*

Implement the following:

Process the following Icore records into the GDB phy port object:

**(A) CE Phy Port**

The “CE Port” data is created starting from a record of the ICORE.SITE table.

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB PHY\_PORT column** | **Processing comment** |
| SITE.site\_id | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘PHY\_PORT’ |
| - | id\_port\_type | Referencing PORT\_TYPE ‘CE’ |
| - | is\_read\_only | ‘Y’ |
| - | instar\_site\_id\_value | Set to NULL |
| SITE.site\_id | icore\_site\_id\_value |  |
| SITE.site\_id | icore\_ce\_port\_site\_id\_value |  |
| SITE.site\_id | id\_site\_inventory\_site\_rep  [via Enterprise Key Translation for “SITE.site\_id”] | Use SOURCE\_KEY.id into GDB |
| Asset.ID  <Redesign Managed Equipment> | id\_asset\_equipment [via Eneterprise Key Translation for “EQP.ODBID” with meta-system as ‘LPP-CPE’] | (~~NULL; will be set later~~)  Note: Logic for populating the value from NC3 schema is given below.  Else, NULL. |
| <BEGIN: Port Load Corrections> |  |  |
| - | Protocol | <to be derived; see “Logic for setting the “protocol” value for SITE records (Physical Port)”> |
| <END: Port Load Corrections> |  |  |
| SITE.contr\_port\_speed | Speed |  |
| - | Name | Set to NULL |
| - | mac\_address | Set to NULL |
| <BEGIN: Port Load Corrections> |  |  |
| - | cos\_profile | Set to NULL (no longer used; 2014-06-24) |
| <END: Port Load Corrections> |  |  |
| - | id\_phy\_port\_aggregator | Set to NULL; not used from within ICORE |
| <286278> |  |  |
|  | ete\_key\_source | <286284-US600815-US600161>  See End-to-End Key Search and Generation [ Icore Data Flow ]  ~~SELECT external\_key\_value AS ete\_port\_key~~  ~~FROM access\_connection\_mapping~~  ~~WHERE internal\_key\_type = 'Icore Site Id' AND internal\_key\_value = <icore.site.id>~~  ~~select ete\_port\_key, ‘USRP’ as ete\_key\_source~~  ~~from usrp.logical\_port~~  ~~where icore\_site\_id = <icore.site.id>~~  ~~and ete\_port\_key is not null~~  ~~and valid\_ind = ‘Y’~~  ~~or~~  ~~select ete\_port\_key, ‘IEOD’ as ete\_key\_source~~  ~~from grid.port~~  ~~where icore\_site\_id = <icore.site.id>~~  ~~Ete\_key\_source should be blank/null when the ete\_port\_key port identifier is removed.~~ |
| </286278> |  |  |
| <290789a-CR154491-Upd-2017-02-08>  ETH\_INTF\_TYPE.eth\_intface\_name | wan\_interface\_name | Use Join:  ETH\_INTF\_TYPE.eth\_intface\_id = SITE.eth\_intface\_id |

<Redesign Managed Equip>

SELECT DISTINCT er.odbid,er.common\_name,

p2.value,

cu3.point\_type

FROM NC3\_MAT.PARM p1,

NC3\_MAT.PHY\_CIRCUIT pc,

NC3\_MAT.CMP\_USAGE cu1,

NC3\_MAT.CMP\_USAGE cu2,

NC3\_MAT.PORT po,

NC3\_MAT.EQP ec,

NC3\_MAT.EQP er,

NC3\_MAT.EQP\_REF err,

NC3\_MAT.PARM p2,

NC3\_MAT.CMP\_USAGE cu3,

NC3\_MAT.CONFIG c

WHERE p1.value = <ICORE SITE ID>

AND p1.internal\_label = 'ICORE\_SITE\_ID'

AND p1.stop\_date > CURRENT\_DATE

AND pc.odbid = p1.odbid\_cmponent

AND pc.odbid = cu1.odbid\_cmponent

AND cu1.stop\_date > CURRENT\_DATE

AND cu1.odbid\_config = cu2.odbid\_config

AND cu2.stop\_date > CURRENT\_DATE

AND cu2.odbid\_cmponent = po.odbid

AND po.odbid\_eqp = ec.odbid

AND ec.odbid\_main\_eqp = er.odbid

AND ec.contain\_stop\_date > CURRENT\_DATE

AND er.odbid\_eqp\_ref = err.odbid

AND err.eqp\_category = 'Router'

AND cu3.odbid\_cmponent = er.odbid

AND cu3.stop\_date > CURRENT\_DATE

AND c.odbid =cu3.odbid\_config

AND p2.odbid\_cmponent =c.odbid

AND p2.internal\_label ='SITE\_BACKUP\_TYPE'

AND p2.stop\_date > CURRENT\_DATE

AND pc.stop\_date > CURRENT\_DATE

AND po.stop\_date > CURRENT\_DATE

AND ec.stop\_date > CURRENT\_DATE

AND er.stop\_date > CURRENT\_DATE

AND err.stop\_date > CURRENT\_DATE

AND c.stop\_date > CURRENT\_DATE;

</Redesign Managed Equip>

**Port Identifier (1): DLCI value:**

<BEGIN: Port Load Corrections>

(only if a value exists and is applicable; see “Logic for setting the “port identifier” value(s) for SITE records (Physical Port)” for detail)

<END: Port Load Corrections>

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB PHY\_PORT\_IDENTIFIER column** | **Processing comment** |
| - | id\_phy\_port | PHY\_PORT.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘DLCI\_PORT\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB PHY\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_phy\_port\_identifier | PHY\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | PHY\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘DLCI\_PORT\_IDENTIFIER’ |
| SITE.site\_glbl\_dlci | Value |  |

**Port Identifier (2): Vpi Vci value pair:**<BEGIN: Port Load Corrections>

(only if applicable and a full value pair exists; see “Logic for setting the “port identifier” value(s) for SITE records (Physical Port)” for detail)

<END: Port Load Corrections>

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB PHY\_PORT\_IDENTIFIER column** | **Processing comment** |
| - | id\_phy\_port | PHY\_PORT.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘VPI\_VCI\_PORT\_IDENTIFIER’ |

Vpi value:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB PHY\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_phy\_port\_identifier | PHY\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | PHY\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘VPI\_PORT\_IDENTIFIER’ |
| SITE.glbl\_vpi | Value |  |

Vci value:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB PHY\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_phy\_port\_identifier | PHY\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | PHY\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘VCI\_PORT\_IDENTIFIER’ |
| SITE.glbl\_vci | Value |  |

<286278>

**Port Identifier (3): ETE\_PORT\_KEY** <REL1602>

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB PHY\_PORT\_IDENTIFIER column** | **Processing comment** |
| - | id\_phy\_port | PHY\_PORT.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘E2E\_PORT\_KEY\_PORT\_IDENTIFIER’ |

Vpi value:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB PHY\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_phy\_port\_identifier | PHY\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | PHY\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘E2E\_PORT\_KEY\_PORT\_IDENTIFIER’ |
|  | Value | <286284-US600815-US600161>  See End-to-End Key Search and Generation [ Icore Data Flow ]  ~~See above.~~  ~~<REL1604>~~  ~~remove the same port identifier (same ete\_port\_key) for other ce port.~~ |

</286278>

<BEGIN: Port Load Corrections>

{All other port identifiers have been removed}

<END: Port Load Corrections>

<BEGIN: Port Load Corrections>

Database traversals:

*NOTE: We do not need to join “CUST\_ACCESS\_CKT”, as its “site\_id” is also present in “CUST\_ACCESS” !*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SITE.site\_id | CUST\_ACCESS.site\_id |
| SITE.site\_id | PORT\_ASGMT.site\_id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |

<END: Port Load Corrections>

<BEGIN: Port Load Corrections>

**<no longer used; 2014-06-25; hc2164>**

~~Database traversals (for getting “cos\_profile” data):~~

|  |  |
| --- | --- |
| **~~From table.column~~** | **~~To table.column~~** |
| ~~SITE.site\_id~~ | ~~SERVICE\_ASGMT.site\_id~~ |
| ~~SERVICE\_OPTION.serv\_opt\_id~~ | ~~SERVICE\_ASGMT.serv\_opt\_id~~ |
| ~~SERVICE\_OPTION.serv\_id~~ | ~~SERVICE.serv\_id~~ |

~~Database constraints (for getting “cos\_profile” data):~~

|  |  |  |
| --- | --- | --- |
| **~~Table name~~** | **~~Column name~~** | **~~Constraint(s)~~** |
| ~~SERVICE~~ | ~~SERV\_NAME~~ | ~~COS PACKAGE~~ |

~~(See also “Logic for pulling “cos profile” values for “Port” data” for more details on the detailed logic.)~~

~~<END: Port Load Corrections>~~

**(B) PE Phy Port**

The “PE Port” data is created starting from the same record of the ICORE.SITE table that corresponds to the record from ICORE.SITE for the “CE Port”.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Icore table.column** | **GDB PHY\_PORT column** | | **Processing comment** | |
| PORT\_ASGMT.cust\_port | id [via Enterprise Key Translation] | | Use SOURCE\_KEY.id into GDB | |
| - | id\_change\_tracking | | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] | |
| - | id\_object\_type | | Referencing OBJECT\_TYPE ‘PHY\_PORT’ | |
| - | id\_port\_type | | Referencing PORT\_TYPE ‘PE’ | |
| - | is\_read\_only | | ‘Y’ | |
| - | instar\_site\_id\_value | | Set to NULL | |
| EQUIPMENT.site\_id | icore\_site\_id\_value | |  | |
| SITE.site\_id | icore\_ce\_port\_site\_id\_value | |  | |
| EQUIPMENT.site\_id | id\_site\_inventory\_site\_rep  [via Enterprise Key Translation for “ICORE.SITE.site\_id” (== “ICORE.equipment.site\_id”)] | | Use SOURCE\_KEY.id into GDB  (if existing; otherwise set to NULL) | |
| - | id\_asset\_equipment | | (NULL; will be set later) | |
| - | protocol | | <to be derived; see “Logic for setting the “protocol” value for SITE records (Physical Port)”> | |
| PORT\_ASGMT.port\_speed | speed | |  | |
| - | name | | Set to NULL | |
| - | mac\_address | | Set to NULL | |
| <BEGIN: Port Load Corrections> |  |  | |
| - | cos\_profile | Set to NULL (no longer used; 2014-06-24) | |
| <END: Port Load Corrections> |  |  | |
| - | id\_phy\_port\_aggregator | | Set to NULL; not used from within ICORE | |
| <290789a.154491>  See Processing Comment | Port\_level\_cos\_ind | | (1) SITE.port\_level\_cos  (2) If the above query returns NULL, then see if GRID.PORT record is found first using below joins:  GRID.PORT.ETE\_PORT\_KEY = GRID.ACCESS\_CIRCUIT.ETE\_PORT\_KEY;  GRID.ACCESS\_CIRCUIT.ETE\_ACCESS\_KEY = GRID.CIRCUIT.ETE\_ACCESS\_KEY;  GRID.CIRCUIT.ETE\_ACCESS\_KEY = GDB.ASSET\_EXT\_ACCESS\_CIRCUIT.ETE\_ACCESS\_KEY  and  GRID.CIRCUIT.CLCI = GDB.ASSET\_EXT\_ACCESS\_CIRCUIT.CLCI  If the GRID.PORT record is found then use GRID.PORT.COS\_INDICATOR;  If the GRID.PORT record is not found, then set the value to ‘N’ | |
| <290789a-CR154491-Upd-2017-02-08>  ETH\_INTF\_TYPE.eth\_intface\_name | wan\_interface\_name | | Use Join:  ETH\_INTF\_TYPE.eth\_intface\_id = SITE.eth\_intface\_id | |
| ~~<296528.161673>~~ | ~~tdm\_indicator~~ | | ~~Set to ‘Y’ if the derived ‘protocol’ value above is ‘MLPPP’ or ‘PPP’ else set to ‘N’~~ | |

NOTE that no identifier data is processed here, as for PE ports no “identifier” values exist !

<290789a.154491> A One time update may need to be done for the existing PHY\_PORT records of type ‘PE’ to update the port\_level\_cos\_ind attribute

~~<296528.161673> A One time update may need to be done for the existing PHY\_PORT records of type ‘PE’ to update the tdm\_idnicator attribute~~

<BEGIN: Port Load Corrections>

Database traversals:

*NOTE: We do not need to join “CUST\_ACCESS\_CKT”, as its “site\_id” is also present in “CUST\_ACCESS” !*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SITE.site\_id | CUST\_ACCESS.site\_id |
| SITE.site\_id | PORT\_ASGMT.site\_id |
| PORT\_ASGMT.equip\_id | EQUIPMENT.equip\_id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |

<END: Port Load Corrections>

<BEGIN: Port Load Corrections>

**<no longer used; 2014-06-25; hc2164>**

~~Database traversals (for getting “cos\_profile” data):~~

|  |  |
| --- | --- |
| **~~From table.column~~** | **~~To table.column~~** |
| ~~SITE.site\_id~~ | ~~SERVICE\_ASGMT.site\_id~~ |
| ~~SERVICE\_OPTION.serv\_opt\_id~~ | ~~SERVICE\_ASGMT.serv\_opt\_id~~ |
| ~~SERVICE\_OPTION.serv\_id~~ | ~~SERVICE.serv\_id~~ |

~~Database constraints (for getting “cos\_profile” data):~~

|  |  |  |
| --- | --- | --- |
| **~~Table name~~** | **~~Column name~~** | **~~Constraint(s)~~** |
| ~~SERVICE~~ | ~~SERV\_NAME~~ | ~~COS PACKAGE~~ |

~~(See also “Logic for pulling “cos profile” values for “Port” data” for more details on the detailed logic.)~~

<END: Port Load Corrections>

<BEGIN: Port Load Corrections>

"Logic for setting the “protocol” value for SITE records (Physical Port)" and "Logic for setting the “port identifier” value(s) for SITE records (Physical Port)" have been renamed and repositioned; the logic itself has remained unchanged.

<BEGIN: Port Load Corrections>

##### Logic for setting the “protocol” value for SITE records (Physical Port)

The logic for setting the “protocol” value for SITE records (Physical Port) is as follows:

(1) Check whether a value exists in the table “DSL\_PORTTYPE.dp\_typedesc” using “SITE.dptype\_id = DSL\_PORTTYPE.dptype\_id” of the current SITE record.  
If it exists, then use the value from “DSL\_PORTTYPE.dp\_typedesc” up to and including the “DSL” sub-string  
(e.g.: “G.SHDSL\_Router” => “G.SHDSL”; “aDSL2+ Annex M\_Router” => “aDSL”).  
If in addition the “SITE.protocol” value is “FR”, then append “/Frame” to the current “protocol” value.

If in addition the “SITE.protocol” value is “ATM”, then append “/ATM” to the current “protocol” value.

(2) If the above does not hold, then check the “SITE.protocol” value.  
If the value is “FR”, then use “Frame”.

If the value is “ATM”, then use “ATM”.

If the value is “ETH”, then use “Ethernet”.

If the value is “POS” and “SITE.pos\_option” is null or blank, then use “Sonet”.

If the value is “POS” and “SITE.pos\_option” contains a value, then

* use “Frame Encap” if the “SITE.pos\_option” value is “FR Encap”
* use the “SITE.pos\_option” value otherwise

(3) If none of the above holds, leave the “protocol” element empty.

##### Logic for setting the “port identifier” value(s) for SITE records (Physical Port)

(This does not apply to PE ports, as for PE ports no “identifier” values exist !)

The logic for setting the “port identifier” value(s) for SITE records (Physical Port) is as follows:

(1) Check the “SITE.protocol” value.  
If the value is “FR”, then use the value from “SITE.site\_glbl\_dlci”.

If the value is “POS” and the “SITE.pos\_option” value is “FR Encap”, then use the value from “SITE.site\_glbl\_dlci”.

If the value is “ATM”, then use the values from “SITE.glbl\_vpi” and “SITE.glbl\_vci”.

(2) If none of the above holds, then do not create “port identifier” records.

<BEGIN: Port Load Corrections>

**<no longer used; 2014-06-25; hc2164>**

##### ~~Logic for pulling “cos profile” values for “Port” data~~

~~The logic for pulling “cos profile” values is given in the embedded file below.~~

~~Note that it returns “<<NO COS>>” if the join as given above does not work for a given “site\_id” value.~~

~~~~

<END: Port Load Corrections>

**Data Blocking (1)**

Do not process port records (SITE) where the record returned by the logic below is linked to a GDB.SERVICE\_TYPE having a record in GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM for Icore:

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SERVICE\_ASGMT.site\_id | SITE.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |
| GDB.SERVICE\_TYPE\_NOTATION.service\_name | SERVICE.serv\_name |
| GDB.SERVICE\_TYPE\_NOTATION.id\_service\_type | GDB.SERVICE\_TYPE.id |
| GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM.id\_service\_type | GDB.SERVICE\_TYPE.id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| SERVICE\_TYPE\_BLOCKED\_SYSTEM | BLOCKED\_SYSTEM | 13609 |

*Note:*

*The BLOCKED\_SYSTEM constraint value is the MOTS-ID of the system.*

**Data Blocking (2)**

Do not process ICORE physical port records where the special logic for blocking INSTAR items hold.

The “<input site id variable (dep. on current scenario)>” here is “SITE.site\_id”.

See “Logic for excluding INSTAR items”.

#### HLD-254035-GCP-FLOW-ICORE-110-D [Icore Data Flow into Phy Port SITE (“Interface Name” data)]

**NOTE: This is a “pseudo-requirement” not to be included in any traceability files.**

NOTE: This requirement will no longer be implemented, as we plan to logically replace ICORE ports by their “master” ports from other port data source systems (like NC3) via “port unification”.

~~This section is on loading “Interface Name” data for physical port data from ICORE where the “Interface Name” is stored in NC3 (for AVPN & EVPN). This is done by mapping via the ICORE SITE.site\_id which implies that it is limited to “CE” ports.~~

~~Before executing this step the two steps “HLD-254035-GCP-FLOW-ICORE-120-A [Icore Data Flow into Phy Port SITE (Layer-3 “Phy Port”)]” and “HLD-254035-GCP-FLOW-ICORE-120-B [Icore Data Flow into Phy Port SITE (Layer-2 Phy Port”)]” must have been completed.~~

~~The data loading needs to be done by “updating” already existing rows of the “GDB.phy\_port” table.~~

~~Implement the following:~~

~~Use the following GDB physical port object as having been loaded from ICORE:~~

|  |  |  |
| --- | --- | --- |
| **~~ICORE table.column~~** | **~~GDB PHY\_PORT column~~** | **~~Processing comment~~** |
| ~~SITE.site\_id~~ | ~~id~~ | ~~Use “GDB.PHY\_PORT.id” (where GDB.PHY\_PORT.id\_port\_type references “~~**~~CE~~**~~”) to identify the appropriate record [via Enterprise Key Translation]~~ |

~~Update the GDB port object using the following NC3 records:~~

|  |  |  |
| --- | --- | --- |
| **~~NC3 table.column~~** | **~~GDB PHY\_PORT column~~** | **~~Processing comment~~** |
| ~~NC3.port.common\_name~~ | ~~name~~ | ~~See “Logic for associating NC3 “Interface Name” data to ICORE and GDB “port” data” for details~~ |

#### HLD-254035-GCP-FLOW-ICORE-110-E [Icore Data Flow into Phy Port SITE (“id\_asset Equipment” data)]

**NOTE: This is a “pseudo-requirement” not to be included in any traceability files.**

NOTE: This requirement will no longer be implemented, as we plan to logically replace ICORE ports by their “master” ports from other port data source systems (like NC3) via “port unification”.

~~This requirement is for covering the relationship between (managed) equipment assets (from “NC3” resp. “LPP-CPE”) and CE physical ports associated with them.~~

~~Before executing this step the two steps “HLD-254035-GCP-FLOW-ICORE-120-A [Icore Data Flow into Phy Port SITE (Layer-3 “Phy Port”)]” and “HLD-254035-GCP-FLOW-ICORE-120-B [Icore Data Flow into Phy Port SITE (Layer-2 Phy Port”)]” must have been completed.~~

~~The data loading needs to be done by “updating” already existing rows of the “GDB.phy\_port” table.~~

~~Implement the following:~~

~~Use the following GDB port object as having been loaded from ICORE:~~

|  |  |  |
| --- | --- | --- |
| **~~ICORE table.column~~** | **~~GDB PHY\_PORT column~~** | **~~Processing comment~~** |
| ~~SITE.site\_id~~ | ~~id~~ | ~~Use “GDB.PHY\_PORT.id” (where GDB.PHY\_PORT.id\_port\_type references “~~**~~CE~~**~~”) to identify the appropriate record [via Enterprise Key Translation]~~ |

~~Update the GDB port object using the following NC3 record:~~

|  |  |  |
| --- | --- | --- |
| **~~NC3 table.column~~** | **~~GDB PHY\_PORT column~~** | **~~Processing comment~~** |
| ~~NC3.eqp.odbid~~ | ~~id\_asset\_equipment~~  ~~[via Enterprise Key Translation; see “Logic for mapping ports, pvcs and (managed NC3) assets” below for obtaining the value for “NC3.eqp.odbid”]~~ | ~~See “Logic for mapping ports, pvcs and (managed NC3) assets” below for details~~ |

#### HLD-254035-GCP-FLOW-ICORE-120-A [Icore Data Flow into Log Port SITE (Layer-3 “Log Port”)]

This section is on loading logical port data from ICORE. Ports are an object type of their own in GDB and are not classified under the “Asset” object type. In the context of port data the ICORE PVC table plays a central role for identifying the PE port that corresponds to a certain CE port. The base table for CE port data is the ICORE SITE table; the base table for PE ports is the ICORE PORT\_ASGMT table.

<BEGIN: Port Load Corrections>

The following section is about “Layer-3” logical ports. (The “Layer-3” terminus is given here in order to indicate how these ports are identified, i.e. via checking for an existing “Layer-3” network connection.) These types of ports are identified in the same way as for “Network Connections (Layer-3)” assets, i.e. by having an entry in the ICORE IPFR table.

<END: Port Load Corrections>

When the data is checked for selectability, only the “customer” side data from the ICORE.PVC table must be taken into account. (Details to be found below and at other appropriate places.)

*NOTE: This requirement needs to be executed after requirement “HLD-254035-GCP-FLOW-ICORE-101 [Icore Data Flow into Site]” has been completed, so the value for “id\_site\_inventory\_site\_rep” will already be available.*

*NOTE: This requirement needs to be executed after all “phy port” related requirements have been completed, so “log ports” can be associated with their “phy ports”.*

Implement the following:

Process the following Icore records into the GDB log\_port object:

*NOTE: Include only records where for the associated PVC a PVC.pvc\_id value exists in the IPFR table under IPFR.pvc\_id.*

*NOTE: Include only records where for the associated PVC only one “cust\_id” value (either “PVC.pvc\_lcust\_id” or “PVC.pvc\_rcust\_id”) points to a “real” customer.*

**(A) CE Logical Port**

The “CE Port” data is created starting from a record of the ICORE.SITE table.

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT column** | **Processing comment** |
| <BEGIN: Port Load Corrections> |  |  |
| *[“customer” on the “l” side]* PVC.pvc\_id + PVC.pvc\_lsite\_id;  *[“customer” on the “r” side]* PVC.pvc\_id + PVC.pvc\_rsite\_id;  *NOTE: For the EKT use the following corresponding values:*  <LOG-PORT-EKT-FIX> LOG\_PORT\_CE\_VW.pvc\_id + LOG\_PORT\_CE\_VW.site\_id; | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| <END: Port Load Corrections> |  |  |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘LOG\_PORT’ |
| - | id\_port\_type | Referencing PORT\_TYPE ‘CE’ |
| - | is\_read\_only | ‘Y’ |
| - | instar\_site\_id\_value | Set to NULL |
| SITE.site\_id | icore\_site\_id\_value |  |
| PVC.pvc\_id | icore\_pvc\_id\_value |  |
| SITE.site\_id | icore\_ce\_port\_site\_id\_value |  |
| SITE.site\_id | id\_site\_inventory\_site\_rep  [via Enterprise Key Translation for “SITE.site\_id”] | Use SOURCE\_KEY.id into GDB |
| - | id\_asset\_equipment | (NULL; will be set later) |
| <BEGIN: Port Load Corrections> |  |  |
| *[“customer/site” on the “l” side]* PVC.pvc\_lproto;  *[“customer/site” on the “r” side]* PVC.pvc\_rproto; | protocol |  |
| *[“customer/site” on the “l” side]* PVC.pvc\_lcontrcir;  *[“customer/site” on the “r” side]* PVC.pvc\_rcontrcir; | speed |  |
| <END: Port Load Corrections> |  |  |
| - | name | Set to NULL |
| - | mac\_address | Set to NULL |
| <BEGIN: Port Load Corrections> |  |  |
| - | cos\_profile | Set to NULL (no longer used; 2014-06-24) |
| <END: Port Load Corrections> |  |  |
| SITE.site\_id | id\_phy\_port | Use SOURCE\_KEY.id into GDB which populated “GDB.PHY\_PORT.id” with “id\_port\_type” referencing “CE” (which must exist; otherwise set to NULL !) |

<BEGIN: Port Load Corrections>

**Port Identifier (1): DLCI value:**   
 Only execute if a DLCI value exists !

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER column** | **Processing comment** |
| - | id\_log\_port | LOG\_PORT.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘DLCI\_PORT\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘DLCI\_PORT\_IDENTIFIER’ |
| *[“customer/site” on the “l” side]* PVC.pvc\_ldlci;  *[“customer/site” on the “r” side]* PVC.pvc\_rdlci; | Value |  |

<END: Port Load Corrections>

<BEGIN: Port Load Corrections>

**Port Identifier (2): Vpi Vci value pair:**   
 Only execute if a full VPI/VCI value pair exists !

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER column** | **Processing comment** |
| - | id\_log\_port | LOG\_PORT.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘VPI\_VCI\_PORT\_IDENTIFIER’ |

Vpi value:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘VPI\_PORT\_IDENTIFIER’ |
| *[“customer/site” on the “l” side]* PVC.pvc\_lvpi;  *[“customer/site” on the “r” side]* PVC.pvc\_rvpi; | Value |  |

Vci value:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘VCI\_PORT\_IDENTIFIER’ |
| *[“customer/site” on the “l” side]* PVC.pvc\_lvci;  *[“customer/site” on the “r” side]* PVC.pvc\_rvci; | Value |  |

<END: Port Load Corrections>

**Port Identifier (3): IP Address (version 4):**   
 Only execute if an IP Address value (version 4) exists !

<BEGIN: Port Load Corrections>

{Text removed}

<END: Port Load Corrections>

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER column** | **Processing comment** |
| - | id\_log\_port | LOG\_PORT.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘IPV4\_PORT\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘IPV4\_IP\_ADDR\_PORT\_IDENTIFIER’ |
| IPFR.cpe\_ip\_address | Value |  |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘IPV4\_SUBNET\_MASK\_PORT\_IDENTIFIER’ |
| IPFR.address\_mask | Value | (If no value, then set to ‘<UNKNOWN>’) |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘IPV4\_ROUTING\_PROTOCOL\_PORT\_IDENTIFIER’ |
| ROUTING\_TYPE.routing\_type | Value | (If no value – due to outer join – then set to ‘<UNKNOWN>’)  Note: remove trailing spaces if any |

**Port Identifier (4): IP Address (version 6):**   
 Only execute if an IP Address value (version 6) exists !

<BEGIN: Port Load Corrections>

{Text removed}

<END: Port Load Corrections>

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER column** | **Processing comment** |
| - | id\_log\_port | LOG\_PORT.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘IPV6\_PORT\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘IPV6\_IP\_ADDR\_PORT\_IDENTIFIER’ |
| IPFR.v6\_cer\_ip | Value |  |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘IPV6\_SUBNET\_MASK\_PORT\_IDENTIFIER’ |
|  | <BEGIN: Port Load Corrections> |  |
| IPFR.v6\_ip\_blocksize | Value | (If no value, then set to ‘<UNKNOWN>’) |
|  | <END: Port Load Corrections> |  |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘IPV6\_ROUTING\_PROTOCOL\_PORT\_IDENTIFIER’ |
| V6\_RTG\_PROTOCOL.protocol | Value | (If no value – due to outer join – then set to ‘<UNKNOWN>’)  Note: remove trailing spaces if any |

**Port Identifier (5): VLAN:**   
 Only execute if a VLAN value exists !

<BEGIN: Port Load Corrections>

{Text removed}

<END: Port Load Corrections>

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER column** | **Processing comment** |
| - | id\_log\_port | LOG\_PORT.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘VLAN\_PORT\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘VLAN\_PORT\_IDENTIFIER’ |
| PVC.vlan\_id | Value |  |

**Port Identifier (6): ETH Bottom:**   
 Only execute if a ETH Bottom value exists !

<BEGIN: Port Load Corrections>

{Text removed}

<END: Port Load Corrections>

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER column** | **Processing comment** |
| - | id\_log\_port | LOG\_PORT.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘ETH\_BOTTOM\_PORT\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘ETH\_BOTTOM\_PORT\_IDENTIFIER’ |
| PVC\_EXTENSION.c\_vlanid\_bot | value |  |

**Port Identifier (7): ETH Top:**   
 Only execute if a ETH Top value exists !

<BEGIN: Port Load Corrections>

{Text removed}

<END: Port Load Corrections>

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER column** | **Processing comment** |
| - | id\_log\_port | LOG\_PORT.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘ETH\_TOP\_PORT\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘ETH\_TOP\_PORT\_IDENTIFIER’ |
| PVC\_EXTENSION.c\_vlanid\_top | Value |  |

<286278>

**Port Identifier (8): ete\_port\_key:**

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER column** | **Processing comment** |
| - | id\_log\_port | LOG\_PORT.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘E2E\_PORT\_KEY\_PORT\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘E2E\_PORT\_KEY\_PORT\_IDENTIFIER’ |
| - | Value | Take the ‘E2E\_PORT\_KEY\_PORT\_IDENTIFIER’ from corresponding phy\_port (via id\_phy\_port)  Note: don’t create the identifier if it is not populated |

</286278>

<BEGIN: Port Load Corrections>

Database traversals:

*(PVC “l” side [“customer/site” on the “l” side])*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SITE.site\_id | <LOG-PORT-EKT-FIX> LOG\_PORT\_CE\_VW.site\_id |
| PVC.pvc\_id | <LOG-PORT-EKT-FIX> LOG\_PORT\_CE\_VW.pvc\_id |
| SITE.site\_id | PVC.pvc\_lsite\_id |
| SITE.site\_id | PORT\_ASGMT.site\_id |
| PVC.pvc\_id | IPFR.pvc\_id |
| PVC.pvc\_id | PVC\_EXTENSION.pvc\_id (use “outer” join here at this side !) |
| IPFR.routing\_type\_id | ROUTING\_TYPE.routing\_type\_id (use “outer” join here at this side !) |
| IPFR.ipfr\_id | V6\_RTG\_PROTOCOL.ipfr\_id (use “outer” join here at this side !) |

<END: Port Load Corrections>

Database constraints:

*(PVC “l” side [“customer/site” on the “l” side])*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| PVC | PVC\_**L**CUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) |
| PVC | PVC\_**R**CUST\_ID | = 395 (i.e. is “AT&T” customer id) |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |

<see also logic under:   
Logic for identifying a “customer” side in the ICORE.PVC table>

<BEGIN: Port Load Corrections>

Database traversals:

*(PVC “r” side [“customer/site” on the “r” side])*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SITE.site\_id | <LOG-PORT-EKT-FIX> LOG\_PORT\_CE\_VW.site\_id |
| PVC.pvc\_id | <LOG-PORT-EKT-FIX> LOG\_PORT\_CE\_VW.pvc\_id |
| SITE.site\_id | PVC.pvc\_rsite\_id |
| SITE.site\_id | PORT\_ASGMT.site\_id |
| PVC.pvc\_id | IPFR.pvc\_id |
| PVC.pvc\_id | PVC\_EXTENSION.pvc\_id (use “outer” join here at this side !) |
| IPFR.routing\_type\_id | ROUTING\_TYPE.routing\_type\_id (use “outer” join here at this side !) |
| IPFR.ipfr\_id | V6\_RTG\_PROTOCOL.ipfr\_id (use “outer” join here at this side !) |

<END: Port Load Corrections>

Database constraints:

*(PVC “r” side [“customer/site” on the “r” side])*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| PVC | PVC\_**R**CUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) |
| PVC | PVC\_**L**CUST\_ID | = 395 (i.e. is “AT&T” customer id) |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |

<see also logic under:   
Logic for identifying a “customer” side in the ICORE.PVC table>

<BEGIN: Port Load Corrections>

**<no longer used; 2014-06-25; hc2164>**

~~For the “cos\_profile”-related “Database traversals” and “Database constraints” make sure you do not simply do a standard “equi” join (because that may suppress too many rows from “ICORE.SITE”), but use some logic as e.g. given below under “Logic for pulling “cos profile” values for “Port” data”.~~

~~Database traversals (for getting “cos\_profile” data):~~

|  |  |
| --- | --- |
| **~~From table.column~~** | **~~To table.column~~** |
| ~~SITE.site\_id~~ | ~~SERVICE\_ASGMT.site\_id~~ |
| ~~SERVICE\_OPTION.serv\_opt\_id~~ | ~~SERVICE\_ASGMT.serv\_opt\_id~~ |
| ~~SERVICE\_OPTION.serv\_id~~ | ~~SERVICE.serv\_id~~ |

~~Database constraints (for getting “cos\_profile” data):~~

|  |  |  |
| --- | --- | --- |
| **~~Table name~~** | **~~Column name~~** | **~~Constraint(s)~~** |
| ~~SERVICE~~ | ~~SERV\_NAME~~ | ~~COS PACKAGE~~ |

<END: Port Load Corrections>

**(B) PE Logical Port**

The “PE Port” data is created by using the “non-customer” (i.e. “provider”) data from the PVC record.  
**NOTE**: For the “site\_id” data the “customer” data is still used !

|  |  |  |  |
| --- | --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT column** | **Processing comment** | |
| <BEGIN: Port Load Corrections> |  |  | |
| *[“customer” on the “l” side]* PVC.pvc\_id + PVC.pvc\_lsite\_id + PVC.pvc\_lcust\_port;  *[“customer” on the “r” side]* PVC.pvc\_id + PVC.pvc\_rsite\_id + PVC.pvc\_rcust\_port;  *NOTE: For the EKT use the following corresponding values:*  LOG\_PORT\_PE\_VW.pvc\_id + LOG\_PORT\_PE\_VW.site\_id + LOG\_PORT\_PE\_VW.cust\_port; | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB | |
| <END: Port Load Corrections> |  |  | |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] | |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘LOG\_PORT’ | |
| - | id\_port\_type | Referencing PORT\_TYPE ‘PE’ | |
| - | is\_read\_only | ‘Y’ | |
| - | instar\_site\_id\_value | Set to NULL | |
| EQUIPMENT.site\_id | icore\_site\_id\_value |  | |
| PVC.pvc\_id | icore\_pvc\_id\_value |  |
| SITE.site\_id | icore\_ce\_port\_site\_id\_value |  | |
| EQUIPMENT.site\_id | id\_site\_inventory\_site\_rep  [via Enterprise Key Translation for “ICORE.SITE.site\_id” (== “ICORE.equipment.site\_id”)] | Use SOURCE\_KEY.id into GDB  (if existing; otherwise set to NULL) | |
| - | id\_asset\_equipment | (NULL; will be set later) | |
| <BEGIN: Port Load Corrections> |  |  | |
| *[“customer/site” on the “l” side]* PVC.pvc\_**r**proto;  *[“customer/site” on the “r” side]* PVC.pvc\_**l**proto; | protocol | **NOTE**: “non-customer” (i.e. “provider”) side is used ! | |
| *[“customer/site” on the “l” side]* PVC.pvc\_**r**contrcir;  *[“customer/site” on the “r” side]* PVC.pvc\_**l**contrcir; | speed | **NOTE**: “non-customer” (i.e. “provider”) side is used ! | |
| <END: Port Load Corrections> |  |  | |
| - | name | Set to NULL | |
| - | mac\_address | Set to NULL | |
| <BEGIN: Port Load Corrections> |  |  | |
| - | cos\_profile | Set to NULL (no longer used; 2014-06-24) | |
| <END: Port Load Corrections> |  |  | |
| PORT\_ASGMT.cust\_port | id\_phy\_port | Use SOURCE\_KEY.id into GDB which populated “GDB.PHY\_PORT.id” with “id\_port\_type” referencing “PE” (which must exist; otherwise set to NULL !) | |

<BEGIN: Port Load Corrections>

**Port Identifier (1): IP Address (version 4):**   
 Only execute if an IP Address value (version 4) exists !

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER column** | **Processing comment** |
| - | id\_log\_port | LOG\_PORT.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘IPV4\_PORT\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘IPV4\_IP\_ADDR\_PORT\_IDENTIFIER’ |
| IPFR.per\_ip\_address | Value |  |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘IPV4\_SUBNET\_MASK\_PORT\_IDENTIFIER’ |
| IPFR.address\_mask | Value | (If no value, then set to ‘<UNKNOWN>’) |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘IPV4\_ROUTING\_PROTOCOL\_PORT\_IDENTIFIER’ |
| ROUTING\_TYPE.routing\_type | Value | (If no value – due to outer join – then set to ‘<UNKNOWN>’)  Note: remove trailing spaces if any |

<END: Port Load Corrections>

<BEGIN: Port Load Corrections>

**Port Identifier (2): IP Address (version 6):**   
 Only execute if an IP Address value (version 6) exists !

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER column** | **Processing comment** |
| - | id\_log\_port | LOG\_PORT.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘IPV6\_PORT\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘IPV6\_IP\_ADDR\_PORT\_IDENTIFIER’ |
| IPFR.v6\_per\_ip | Value |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** | |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id | |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking | |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘IPV6\_SUBNET\_MASK\_PORT\_IDENTIFIER’ | |
|  | <BEGIN: Port Load Corrections> | |  |
| IPFR.v6\_ip\_blocksize | Value | | (If no value, then set to ‘<UNKNOWN>’) |
|  | <END: Port Load Corrections> | |  |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘IPV6\_ROUTING\_PROTOCOL\_PORT\_IDENTIFIER’ |
| V6\_RTG\_PROTOCOL.protocol | Value | (If no value – due to outer join – then set to ‘<UNKNOWN>’)  Note: remove trailing spaces if any |

<END: Port Load Corrections>

<BEGIN: Port Load Corrections>

**Port Identifier (3): DLCI value:**   
 Only execute if a DLCI value exists !

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER column** | **Processing comment** |
| - | id\_log\_port | LOG\_PORT.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘DLCI\_PORT\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘DLCI\_PORT\_IDENTIFIER’ |
| *[“customer/site” on the “l” side]* PVC.pvc\_**r**dlci;  *[“customer/site” on the “r” side]* PVC.pvc\_**l**dlci; | Value | **NOTE**: “non-customer” (i.e. “provider”) side is used ! |

<END: Port Load Corrections>

<BEGIN: Port Load Corrections>

**Port Identifier (4): Vpi Vci value pair:**   
 Only execute if a full VPI/VCI value pair exists !

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER column** | **Processing comment** |
| - | id\_log\_port | LOG\_PORT.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘VPI\_VCI\_PORT\_IDENTIFIER’ |

Vpi value:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘VPI\_PORT\_IDENTIFIER’ |
| *[“customer/site” on the “l” side]* PVC.pvc\_**r**vpi;  *[“customer/site” on the “r” side]* PVC.pvc\_**l**vpi; | Value | **NOTE**: “non-customer” (i.e. “provider”) side is used ! |

Vci value:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘VCI\_PORT\_IDENTIFIER’ |
| *[“customer/site” on the “l” side]* PVC.pvc\_**r**vci;  *[“customer/site” on the “r” side]* PVC.pvc\_**l**vci; | Value | **NOTE**: “non-customer” (i.e. “provider”) side is used ! |

**~~Port Identifier (5): VLAN:~~** ~~Not applicable here !~~

**~~Port Identifier (6): ETH Bottom:~~** ~~Not applicable here !~~

**~~Port Identifier (7): ETH Top:~~** ~~Not applicable here !~~

<END: Port Load Corrections>

The following sections contain data access logic for these steps:

1. Identifying the “customer” side of a PVC
2. Identifying the “PE” port
3. ~~Pulling “cos profile” data~~

Step 1 and 2 belong together, as for step 2 the “customer” side of a PVC is used, i.e. the result from step 1.

The next two “Database Traversal/Constraint” blocks are for idfentifying the “customer” side of the PVC record.

Database traversals (1):

*(PVC “l” side [“customer/site” on the “l” side])*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SITE.site\_id | PVC.pvc\_lsite\_id |
| PVC.pvc\_id | IPFR.pvc\_id |

Database constraints (1):

*(PVC “l” side [“customer/site” on the “l” side])*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| PVC | PVC\_**L**CUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) |
| PVC | PVC\_**R**CUST\_ID | = 395 (i.e. is “AT&T customer” customer id) |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |

<see also logic under:   
Logic for identifying a “customer” side in the ICORE.PVC table>

Database traversals (2):

*(PVC “r” side [“customer/site” on the “r” side])*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SITE.site\_id | PVC.pvc\_rsite\_id |
| PVC.pvc\_id | IPFR.pvc\_id |

Database constraints (2):

*(PVC “r” side [“customer/site” on the “r” side])*

|  |  |  |  |
| --- | --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** | |
| PVC | PVC\_**R**CUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) | |
| PVC | PVC\_**L**CUST\_ID | = 395 (i.e. is “AT&T customer” customer id) |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE | |

<see also logic under:   
Logic for identifying a “customer” side in the ICORE.PVC table>

The next two “Database Traversal/Constraint” blocks are for identifying the “PE port” side of the PVC record.

<BEGIN: Port Load Corrections>

**The basic rule here is:**

***When the “customer” side is on “l”, then the “PE” (“provider”) side is on “r”;***

***When the “customer” side is on “r”, then the “PE” (“provider”) side is on “l”.***

Database traversals (3):

*(PVC PE port “r” side [“PE port” on the “r” side])*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| PVC.pvc\_id | LOG\_PORT\_PE\_VW.pvc\_id |
| PVC.pvc\_rsite\_id | LOG\_PORT\_PE\_VW.site\_id |
| PVC.pvc\_rcust\_port | LOG\_PORT\_PE\_VW.cust\_port |
| PVC.pvc\_id | IPFR.pvc\_id |
| PVC.pvc\_rcust\_port | PORT\_ASGMT.cust\_port |
| PORT\_ASGMT.equip\_id | EQUIPMENT.equip\_id |
| IPFR.routing\_type\_id | ROUTING\_TYPE.routing\_type\_id (use “outer” join here at this side !) |
| IPFR.ipfr\_id | V6\_RTG\_PROTOCOL.ipfr\_id (use “outer” join here at this side !) |

Database traversals (4):

*(PVC PE port “l” side [“PE port” on the “l” side])*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| PVC.pvc\_id | LOG\_PORT\_PE\_VW.pvc\_id |
| PVC.pvc\_lsite\_id | LOG\_PORT\_PE\_VW.site\_id |
| PVC.pvc\_lcust\_port | LOG\_PORT\_PE\_VW.cust\_port |
| PVC.pvc\_id | IPFR.pvc\_id |
| PVC.pvc\_lcust\_port | PORT\_ASGMT.cust\_port |
| PORT\_ASGMT.equip\_id | EQUIPMENT.equip\_id |
| IPFR.routing\_type\_id | ROUTING\_TYPE.routing\_type\_id (use “outer” join here at this side !) |
| IPFR.ipfr\_id | V6\_RTG\_PROTOCOL.ipfr\_id (use “outer” join here at this side !) |

<END: Port Load Corrections>

<BEGIN: Port Load Corrections>

**<no longer used; 2014-06-25; hc2164>**

~~Database traversals (for getting “cos\_profile” data):~~

|  |  |
| --- | --- |
| **~~From table.column~~** | **~~To table.column~~** |
| ~~SITE.site\_id~~ | ~~SERVICE\_ASGMT.site\_id~~ |
| ~~SERVICE\_OPTION.serv\_opt\_id~~ | ~~SERVICE\_ASGMT.serv\_opt\_id~~ |
| ~~SERVICE\_OPTION.serv\_id~~ | ~~SERVICE.serv\_id~~ |

~~Database constraints (for getting “cos\_profile” data):~~

|  |  |  |
| --- | --- | --- |
| **~~Table name~~** | **~~Column name~~** | **~~Constraint(s)~~** |
| ~~SERVICE~~ | ~~SERV\_NAME~~ | ~~COS PACKAGE~~ |

~~(See also “Logic for pulling “cos profile” values for “Port” data” for more details on the detailed logic.)~~

<END: Port Load Corrections>

**Data Blocking (1)**

Do not process port records (SITE) where the record returned by the logic below is linked to a GDB.SERVICE\_TYPE having a record in GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM for Icore:

Database traversals:

*(PVC “l” side [“custome/site r” on the “l” side])*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| PVC.pvc\_lsite\_id | SITE.site\_id |
| SERVICE\_ASGMT.site\_id | SITE.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |
| GDB.SERVICE\_TYPE\_NOTATION.service\_name | SERVICE.serv\_name |
| GDB.SERVICE\_TYPE\_NOTATION.id\_service\_type | GDB.SERVICE\_TYPE.id |
| GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM.id\_service\_type | GDB.SERVICE\_TYPE.id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| SERVICE\_TYPE\_BLOCKED\_SYSTEM | BLOCKED\_SYSTEM | 13609 |

Database traversals:

*(PVC “r” side [“customer/site” on the “r” side])*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| PVC.pvc\_rsite\_id | SITE.site\_id |
| SERVICE\_ASGMT.site\_id | SITE.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |
| GDB.SERVICE\_TYPE\_NOTATION.service\_name | SERVICE.serv\_name |
| GDB.SERVICE\_TYPE\_NOTATION.id\_service\_type | GDB.SERVICE\_TYPE.id |
| GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM.id\_service\_type | GDB.SERVICE\_TYPE.id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| SERVICE\_TYPE\_BLOCKED\_SYSTEM | BLOCKED\_SYSTEM | 13609 |

*Note:*

*The BLOCKED\_SYSTEM constraint value is the MOTS-ID of the system.*

**Data Blocking (2)**

Do not process ICORE logical port records where the special logic for blocking INSTAR items hold.

The “<input site id variable (dep. on current scenario)>” here is “SITE. site\_id”.

See “Logic for excluding INSTAR items”.

#### HLD-254035-GCP-FLOW-ICORE-120-B [Icore Data Flow into Log Port SITE (Layer-2 “Log Port”)]

This section is on loading logical port data from ICORE. Logical ports are an object type of their own in GDB and are not classified under the “Asset” object type. In the context of port data the ICORE PVC table plays a central role for identifying the PE port that corresponds to a certain CE port. The base table for CE port data is the ICORE SITE table; the base table for PE ports is the ICORE PORT\_ASGMT table.

<BEGIN: Port Load Corrections>

The following section is about “Layer-2” ports. (The “Layer-2” terminus is given here in order to indicate how these ports are identified, i.e. via checking for an existing “Layer-2” network connection.) These types of ports are identified in the same way as for “Network Connections (Layer-2)” assets, i.e. by -\*NOT\*- having an entry in the ICORE IPFR table and having a non-AT&T customer id value at both sides. When the data is checked for selectability, both sides from the ICORE.PVC table must be “customer” sides. (Details to be found below and at other appropriate places.)

<END: Port Load Corrections>

*NOTE: This requirement needs to be executed after requirement “HLD-254035-GCP-FLOW-ICORE-101 [Icore Data Flow into Site]” has been completed, so the value for “id\_site\_inventory\_site\_rep” will already be available.*

*NOTE: This requirement needs to be executed after all “phy port” related requirements have been completed, so “log ports” can be associated with their “phy ports”.*

Implement the following:

Process the following Icore records into the GDB log port object:

*NOTE: Include only records where for the associated PVC* ***-\*NO\*-*** *PVC.pvc\_id value exists in the IPFR table under IPFR.pvc\_id* ***AND*** *where for the associated PVC* ***both*** *the “PVC.pvc\_lcust\_id” and the “PVC.pvc\_rcust\_id” value point to a “real” customer.*

**(A) CE Log Port**

The “CE Port” data is created starting from a record of the ICORE.SITE table.

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT column** | **Processing comment** |
| <BEGIN: Port Load Corrections> |  |  |
| *[Using “l” side]* PVC.pvc\_id + PVC.pvc\_lsite\_id;  *[Using “r” side]* PVC.pvc\_id + PVC.pvc\_rsite\_id;  *NOTE: For the EKT use the following corresponding values:*  <LOG-PORT-EKT-FIX> LOG\_PORT\_CE\_VW.pvc\_id + LOG\_PORT\_CE\_VW.site\_id; | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| <END: Port Load Corrections> |  |  |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘LOG\_PORT’ |
| - | id\_port\_type | Referencing PORT\_TYPE ‘CE’ |
| - | is\_read\_only | ‘Y’ |
| - | instar\_site\_id\_value | Set to NULL |
| SITE.site\_id | icore\_site\_id\_value |  |
| PVC.pvc\_id | icore\_pvc\_id\_value |  |
| SITE.site\_id | icore\_ce\_port\_site\_id\_value |  |
| SITE.site\_id | id\_site\_inventory\_site\_rep  [via Enterprise Key Translation for “SITE.site\_id”] | Use SOURCE\_KEY.id into GDB |
| - | id\_asset\_equipment | (NULL; will be set later) |
| <BEGIN: Port Load Corrections> |  |  |
| *[Using “l” side]* PVC.pvc\_lproto;  *[Using “r” side]* PVC.pvc\_rproto; | Protocol |  |
| *[Using “l” side]* PVC.pvc\_lcontrcir;  *[Using “r” side]* PVC.pvc\_rcontrcir; | speed |  |
| <END: Port Load Corrections> |  |  |
| - | name | Set to NULL |
| - | mac\_address | Set to NULL |
| <BEGIN: Port Load Corrections> |  |  |
| - | cos\_profile | Set to NULL (no longer used; 2014-06-24) |
| <END: Port Load Corrections> |  |  |
| SITE.site\_id | id\_phy\_port | Use SOURCE\_KEY.id into GDB which populated “GDB.PHY\_PORT.id” with “id\_port\_type” referencing “CE” (which must exist; otherwise set to NULL !) |

**Port Identifier (1): DLCI value:**   
 Only execute if a DLCI value exists !

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER column** | **Processing comment** |
| - | id\_log\_port | LOG\_PORT.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘DLCI\_PORT\_IDENTIFIER’ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | | **Processing comment** | | |
| - | id\_log\_port\_identifier | | LOG\_PORT\_IDENTIFIER.id | | |
| - | id\_change\_tracking | | LOG\_PORT\_IDENTIFIER.id\_change\_tracking | | |
| - | id\_identifier\_type | | Referencing IDENTIFIER\_TYPE ‘DLCI\_PORT\_IDENTIFIER’ | | |
| <BEGIN: Port Load Corrections> | |  | |  |
| *[Using “l” side]* PVC.pvc\_ldlci;  *[Using “r” side]* PVC.pvc\_rdlci; | Value | |  | | |
| <END: Port Load Corrections> | |  | |  |

**Port Identifier (2): Vpi Vci value pair:**   
 Only execute if a full VPI/VCI value pair exists !

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER column** | **Processing comment** |
| - | id\_log\_port | LOG\_PORT.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘VPI\_VCI\_PORT\_IDENTIFIER’ |

Vpi value:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | | **Processing comment** | | |
| - | id\_log\_port\_identifier | | LOG\_PORT\_IDENTIFIER.id | | |
| - | id\_change\_tracking | | LOG\_PORT\_IDENTIFIER.id\_change\_tracking | | |
| - | id\_identifier\_type | | Referencing IDENTIFIER\_TYPE ‘VPI\_PORT\_IDENTIFIER’ | | |
| <BEGIN: Port Load Corrections> | |  | |  |
| *[Using “l” side]* PVC.pvc\_lvpi;  *[Using “r” side]* PVC.pvc\_rvpi; | Value | |  | | |
| <END: Port Load Corrections> | |  | |  |

Vci value:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | | **Processing comment** | | |
| - | id\_log\_port\_identifier | | LOG\_PORT\_IDENTIFIER.id | | |
| - | id\_change\_tracking | | LOG\_PORT\_IDENTIFIER.id\_change\_tracking | | |
| - | id\_identifier\_type | | Referencing IDENTIFIER\_TYPE ‘VCI\_PORT\_IDENTIFIER’ | | |
| <BEGIN: Port Load Corrections> | |  | |  |
| *[Using “l” side]* PVC.pvc\_lvci;  *[Using “r” side]* PVC.pvc\_rvci; | Value | |  | | |
| <END: Port Load Corrections> | |  | |  |

<286278>

**Port Identifier (2): ete\_port\_key:**

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER column** | **Processing comment** |
| - | id\_log\_port | LOG\_PORT.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘E2E\_PORT\_KEY\_PORT\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘E2E\_PORT\_KEY\_PORT\_IDENTIFIER’ |
| - | Value | Take the ‘E2E\_PORT\_KEY\_PORT\_IDENTIFIER’ from corresponding phy\_port (via id\_phy\_port)  Note: don’t create the identifier if it is not populated |

</286278>

Database traversals:

*(PVC “l” side [“site” record located on the “l” PVC side])*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SITE.site\_id | PVC.pvc\_lsite\_id |

*(PVC “r” side [“site” record located on the “r” PVC side])*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SITE.site\_id | PVC.pvc\_rsite\_id |

<BEGIN: Port Load Corrections>

*(generic, i.e. independently of where “site” record located in the PVC record)*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SITE.site\_id | <LOG-PORT-EKT-FIX> LOG\_PORT\_CE\_VW.site\_id |
| PVC.pvc\_id | <LOG-PORT-EKT-FIX> LOG\_PORT\_CE\_VW.pvc\_id |
| SITE.site\_id | PORT\_ASGMT.site\_id |
| PVC.pvc\_id | Not exists(IPFR.pvc\_id) |

<END: Port Load Corrections>

Database constraints:

*(generic; PVC “l” side or PVC “r” side)*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| PVC | PVC\_LCUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) |
| PVC | PVC\_RCUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) |
| PORT\_ASGMT *(via “PVC.pvc\_lsite\_id”)* | PORT\_STAT | IN-SERVICE |

<see also logic under:   
Logic for identifying a “customer” side in the ICORE.PVC table>

<BEGIN: Port Load Corrections>

**<no longer used; 2014-06-25; hc2164>**

~~Database traversals (for getting “cos\_profile” data):~~

|  |  |
| --- | --- |
| **~~From table.column~~** | **~~To table.column~~** |
| ~~SITE.site\_id~~ | ~~SERVICE\_ASGMT.site\_id~~ |
| ~~SERVICE\_OPTION.serv\_opt\_id~~ | ~~SERVICE\_ASGMT.serv\_opt\_id~~ |
| ~~SERVICE\_OPTION.serv\_id~~ | ~~SERVICE.serv\_id~~ |

~~Database constraints (for getting “cos\_profile” data):~~

|  |  |  |
| --- | --- | --- |
| **~~Table name~~** | **~~Column name~~** | **~~Constraint(s)~~** |
| ~~SERVICE~~ | ~~SERV\_NAME~~ | ~~COS PACKAGE~~ |

~~(See also “Logic for pulling “cos profile” values for “Port” data” for more details on the detailed logic.)~~

<END: Port Load Corrections>

**(B) PE Log Port**

The “PE Port” data is created starting from the same record of the ICORE.SITE table that corresponds to the record from ICORE.SITE for the “CE Port”.

**NOTE**: As here no “non-customer” PVC side (either “R” or “l”) exists, we will create some type of “dummy” PE record using some of the “CE” data.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT column** | | **Processing comment** | |
| <BEGIN: Port Load Corrections> |  |  | |
| *[Using “l” side]* PVC.pvc\_id + PVC.pvc\_lsite\_id + PVC.pvc\_lcust\_port;  *[Using “r” side]* PVC.pvc\_id + PVC.pvc\_rsite\_id + PVC.pvc\_rcust\_port;  *NOTE: For the EKT use the following corresponding values:*  LOG\_PORT\_PE\_VW.pvc\_id + LOG\_PORT\_PE\_VW.site\_id + LOG\_PORT\_PE\_VW.cust\_port; | id [via Enterprise Key Translation] | | Use SOURCE\_KEY.id into GDB | |
| <END: Port Load Corrections> |  |  | |
| - | id\_change\_tracking | | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] | |
| - | id\_object\_type | | Referencing OBJECT\_TYPE ‘LOG\_PORT’ | |
| - | id\_port\_type | | Referencing PORT\_TYPE ‘PE’ | |
| - | is\_read\_only | | ‘Y’ | |
| - | instar\_site\_id\_value | | Set to NULL | |
| EQUIPMENT.site\_id | icore\_site\_id\_value | |  | |
| PVC.pvc\_id | icore\_pvc\_id\_value | |  |
| SITE.site\_id | icore\_ce\_port\_site\_id\_value | |  | |
| EQUIPMENT.site\_id | id\_site\_inventory\_site\_rep  [via Enterprise Key Translation for “ICORE.SITE.site\_id” (== “ICORE.equipment.site\_id”)] | | Use SOURCE\_KEY.id into GDB  (if existing; otherwise set to NULL) | |
| - | id\_asset\_equipment | | (NULL; will be set later) | |
| <BEGIN: Port Load Corrections> |  |  | |
| *[Using “l” side]* PVC.pvc\_lproto;  *[Using “r” side]* PVC.pvc\_rproto; | Protocol | |  | |
| *[Using “l” side]* PVC.pvc\_lcontrcir;  *[Using “r” side]* PVC.pvc\_rcontrcir; | speed | |  | |
| <END: Port Load Corrections> |  | |  | |
| - | name | | Set to NULL | |
| - | mac\_address | | Set to NULL | |
| <BEGIN: Port Load Corrections> |  | |  | |
| - | cos\_profile | | Set to NULL (no longer used; 2014-06-24) | |
| <END: Port Load Corrections> |  | |  | |
| PORT\_ASGMT.cust\_port | id\_phy\_port | | Use SOURCE\_KEY.id into GDB which populated “GDB.PHY\_PORT.id” with “id\_port\_type” referencing “PE” (which must exist; otherwise set to NULL !) | |

NOTE that no identifier data is processed here, as for “Layer-2” logical PE ports no “identifier” values can be identified !

**~~Port Identifier (1): DLCI value:~~**  ~~Not applicable here !~~

**~~Port Identifier (2): Vpi Vci value pair:~~**  ~~Not applicable here !~~

Database traversals:

*(PVC “l” side [“site” record located on the “l” PVC side])*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SITE.site\_id | PVC.pvc\_lsite\_id |

*(PVC “r” side [“site” record located on the “r” PVC side])*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SITE.site\_id | PVC.pvc\_rsite\_id |

<BEGIN: Port Load Corrections>

*(generic, i.e. independently of where “site” record located in the PVC record)*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| PVC.pvc\_id | LOG\_PORT\_PE\_VW.pvc\_id |
| SITE.site\_id | LOG\_PORT\_PE\_VW.site\_id |
| PORT\_ASGMT.cust\_port | LOG\_PORT\_PE\_VW.cust\_port |
| SITE.site\_id | PORT\_ASGMT.site\_id |
| PORT\_ASGMT.equip\_id | EQUIPMENT.equip\_id |
| PVC.pvc\_id | Not exists(IPFR.pvc\_id) |

<END: Port Load Corrections>

Database constraints:

*(generic; PVC “l” side or PVC “r” side)*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| PVC | PVC\_LCUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) |
| PVC | PVC\_RCUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) |
| PORT\_ASGMT *(via “PVC.pvc\_lsite\_id”)* | PORT\_STAT | IN-SERVICE |

<see also logic under:   
Logic for identifying a “customer” side in the ICORE.PVC table>

<BEGIN: Port Load Corrections>

**<no longer used; 2014-06-25; hc2164>**

~~Database traversals (for getting “cos\_profile” data):~~

|  |  |
| --- | --- |
| **~~From table.column~~** | **~~To table.column~~** |
| ~~SITE.site\_id~~ | ~~SERVICE\_ASGMT.site\_id~~ |
| ~~SERVICE\_OPTION.serv\_opt\_id~~ | ~~SERVICE\_ASGMT.serv\_opt\_id~~ |
| ~~SERVICE\_OPTION.serv\_id~~ | ~~SERVICE.serv\_id~~ |

~~Database constraints (for getting “cos\_profile” data):~~

|  |  |  |
| --- | --- | --- |
| **~~Table name~~** | **~~Column name~~** | **~~Constraint(s)~~** |
| ~~SERVICE~~ | ~~SERV\_NAME~~ | ~~COS PACKAGE~~ |

~~(See also “Logic for pulling “cos profile” values for “Port” data” for more details on the detailed logic.)~~

<END: Port Load Corrections>

**Data Blocking (1)**

Do not process port records (SITE) where the record returned by the logic below is linked to a GDB.SERVICE\_TYPE having a record in GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM for Icore:

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SERVICE\_ASGMT.site\_id | SITE.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |
| GDB.SERVICE\_TYPE\_NOTATION.service\_name | SERVICE.serv\_name |
| GDB.SERVICE\_TYPE\_NOTATION.id\_service\_type | GDB.SERVICE\_TYPE.id |
| GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM.id\_service\_type | GDB.SERVICE\_TYPE.id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| SERVICE\_TYPE\_BLOCKED\_SYSTEM | BLOCKED\_SYSTEM | 13609 |

*Note:*

*The BLOCKED\_SYSTEM constraint value is the MOTS-ID of the system.*

**Data Blocking (2)**

Do not process ICORE logical port records where the special logic for blocking INSTAR items hold.

The “<input site id variable (dep. on current scenario)>” here is “SITE.site\_id”.

See “Logic for excluding INSTAR items”.

#### HLD-254035-GCP-FLOW-ICORE-120-C [Icore Data Flow into Log Port SITE (Internet VLAN “Log Port”)]

This section is on loading logical port data from ICORE. Ports are an object type of their own in GDB and are not classified under the “Asset” object type. In the context of port data the ICORE PVC table plays a central role for identifying the PE port that corresponds to a certain CE port. The base table for CE port data is the ICORE SITE table; the base table for PE ports is the ICORE PORT\_ASGMT table.

<BEGIN: Port Load Corrections>

The following section is about “Internet VLAN” logical ports. These types of ports are identified in the same way as for “Network Connections (Internet VLAN)” assets, i.e. by -\*NOT\*- having an entry in the ICORE.VPN table and ICORE.IPFR table, but having a record in the ICORE.PVC table that is visible via the ICORE.NETWORK\_CONNECTION\_INET\_VLAN view.

<END: Port Load Corrections>

When the data is checked for selectability, only the “customer” side data from the ICORE.PVC table must be taken into account. (Details to be found below and at other appropriate places.)

*NOTE: This requirement needs to be executed after requirement “HLD-254035-GCP-FLOW-ICORE-101 [Icore Data Flow into Site]” has been completed, so the value for “id\_site\_inventory\_site\_rep” will already be available.*

*NOTE: This requirement needs to be executed after all “phy port” related requirements have been completed, so “log ports” can be associated with their “phy ports”.*

Implement the following:

Process the following Icore records into the GDB log\_port object:

*NOTE: Include only records where for the associated PVC only one “cust\_id” value (either “PVC.pvc\_lcust\_id” or “PVC.pvc\_rcust\_id”) points to a “real” customer.*

**(A) CE Logical Port**

The “CE Port” data is created starting from a record of the ICORE.SITE table.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Icore table.column** | | **GDB LOG\_PORT column** | | | **Processing comment** | | | |
| <BEGIN: Port Load Corrections> |  | |  | | |
| *[“customer” on the “l” side]* PVC.pvc\_id + PVC.pvc\_lsite\_id;  *[“customer” on the “r” side]* PVC.pvc\_id + PVC.pvc\_rsite\_id;  *NOTE: For the EKT use the following corresponding values:*  <LOG-PORT-EKT-FIX> LOG\_PORT\_CE\_INET\_VLAN\_VW.vpn\_id + LOG\_PORT\_CE\_INET\_VLAN\_VW.pvc\_id + LOG\_PORT\_CE\_INET\_VLAN\_VW.site\_id; | | id [via Enterprise Key Translation] | | | Use SOURCE\_KEY.id into GDB | | | |
| <END: Port Load Corrections> |  | |  | | |
| - | | id\_change\_tracking | | | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] | | | |
| - | | id\_object\_type | | | Referencing OBJECT\_TYPE ‘LOG\_PORT’ | | | |
| - | | id\_port\_type | | | Referencing PORT\_TYPE ‘CE’ | | | |
| - | | is\_read\_only | | | ‘Y’ | | | |
| - | | instar\_site\_id\_value | | | Set to NULL | | | |
| SITE.site\_id | | icore\_site\_id\_value | | |  | | | |
| PVC.pvc\_id | | icore\_pvc\_id\_value | | |  | |
| SITE.site\_id | | icore\_ce\_port\_site\_id\_value | | |  | | | |
| SITE.site\_id | | id\_site\_inventory\_site\_rep  [via Enterprise Key Translation for “SITE.site\_id”] | | | Use SOURCE\_KEY.id into GDB | | | |
| - | | id\_asset\_equipment | | | (NULL; will be set later) | | | |
| LOG\_PORT\_CE\_INET\_VLAN\_VW.proto; | | Protocol | | |  | | | |
| <BEGIN: Port Load Corrections> | |  | |  | | | |
| LOG\_PORT\_CE\_INET\_VLAN\_VW.contrcir; | | speed | | |  | | | |
| <END: Port Load Corrections> | |  | |  | | | |
| - | | name | | | Set to NULL | | | |
| - | | mac\_address | | | Set to NULL | | | |
| <BEGIN: Port Load Corrections> | |  | |  | | | |
| - | | cos\_profile | | Set to NULL (no longer used; 2014-06-24) | | | |
| <END: Port Load Corrections> | |  | |  | | | |
| SITE.site\_id | | id\_phy\_port | | | Use SOURCE\_KEY.id into GDB which populated “GDB.PHY\_PORT.id” with “id\_port\_type” referencing “CE” (which must exist; otherwise set to NULL !) | | | |

<BEGIN: Port Load Corrections>

**Port Identifier (1): IP Address (version 4):**   
 Only execute if an IP Address value (version 4) exists !

*(NOTE:   
At the moment it seems that ip v4 address data is not available for “Internet VLAN” items,   
so the approach here is to try to pull ip-address-related identifier data from INSTAR)*

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER column** | **Processing comment** |
| - | id\_log\_port | LOG\_PORT.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘IPV4\_PORT\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘IPV4\_IP\_ADDR\_PORT\_IDENTIFIER’ |
| **INSTAR**.SERIAL\_IP\_ADDR.ip\_address | Value |  |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘IPV4\_SUBNET\_MASK\_PORT\_IDENTIFIER’ |
| **INSTAR**.SERIAL\_IP\_ADDR.subnet\_mask | Value |  |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘IPV4\_ROUTING\_PROTOCOL\_PORT\_IDENTIFIER’ |
| - | Value | set to ‘<UNKNOWN>’ |

<END: Port Load Corrections>

<BEGIN: Port Load Corrections>

**Port Identifier (2): IP Address (version 6):**   
 Only execute if an IP Address value (version 6) exists !

*(NOTE:   
At the moment it seems that ip v6 address data is not available for “Internet VLAN” items,   
so the approach here is to try to pull ip-address-related identifier data from INSTAR)*

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER column** | **Processing comment** |
| - | id\_log\_port | LOG\_PORT.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘IPV6\_PORT\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘IPV6\_IP\_ADDR\_PORT\_IDENTIFIER’ |
| First attempt: **INSTAR**.IPV6\_ASSIGNED\_LINK\_IPS.ipv6\_address  Second attempt:  **INSTAR**.IPV6\_CUST\_LINK\_IPS.ipv6\_address | Value | First attempt:  Via **INSTAR**.IP\_PORT\_ASGMT, **INSTAR**.IPV6\_PORT\_ASGMT\_MAP using **INSTAR**.IPV6\_ADDRESS\_TYPE ‘IPv6\_CR Address’.  Second attempt (if first doesn’t return an ip address):  Via **INSTAR**.IP\_PORT\_ASGMT, **INSTAR**.IPV6\_PORT\_ASGMT\_MAP using **INSTAR**.IPV6\_ADDRESS\_TYPE ‘IPv6\_Cust\_CER\_Address’. |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘IPV6\_SUBNET\_MASK\_PORT\_IDENTIFIER’ |
| First attempt: **INSTAR**.IPV6\_ASSIGNED\_LINK\_IPS.length  Second attempt:  **INSTAR**.IPV6\_CUST\_LINK\_IPS.length | Value | First attempt:  Via **INSTAR**.IP\_PORT\_ASGMT, **INSTAR**.IPV6\_PORT\_ASGMT\_MAP using **INSTAR**.IPV6\_ADDRESS\_TYPE ‘IPv6\_CR Address’.  Second attempt (if first doesn’t return an ip address):  Via **INSTAR**.IP\_PORT\_ASGMT, **INSTAR**.IPV6\_PORT\_ASGMT\_MAP using **INSTAR**.IPV6\_ADDRESS\_TYPE ‘IPv6\_Cust\_CER\_Address’. |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘IPV6\_ROUTING\_PROTOCOL\_PORT\_IDENTIFIER’ |
| - | Value | set to ‘<UNKNOWN>’ |

<END: Port Load Corrections>

**Port Identifier (3): VLAN:**   
 Execute even if a VLAN value does -\*NOT\*- exist (see below) !

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER column** | **Processing comment** |
| - | id\_log\_port | LOG\_PORT.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘VLAN\_PORT\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘VLAN\_PORT\_IDENTIFIER’ |
| PVC.vlan\_id | Value | If none exists (i.e. PVC.vlan\_id is empty or NULL), then set to “Internet VLAN”) |

**Port Identifier (4): ETH Bottom:**   
 Only execute if a ETH Bottom value exists !

<BEGIN: Port Load Corrections>

{Text removed}

<END: Port Load Corrections>

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER column** | **Processing comment** |
| - | id\_log\_port | LOG\_PORT.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘ETH\_BOTTOM\_PORT\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘ETH\_BOTTOM\_PORT\_IDENTIFIER’ |
| PVC\_EXTENSION.c\_vlanid\_bot | Value |  |

**Port Identifier (5): ETH Top:**   
 Only execute if a ETH Top value exists !

<BEGIN: Port Load Corrections>

{Text removed}

<END: Port Load Corrections>

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER column** | **Processing comment** |
| - | id\_log\_port | LOG\_PORT.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘ETH\_TOP\_PORT\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘ETH\_TOP\_PORT\_IDENTIFIER’ |
| PVC\_EXTENSION.c\_vlanid\_top | Value |  |

<286278>

**Port Identifier (6): ete\_port\_key:**

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER column** | **Processing comment** |
| - | id\_log\_port | LOG\_PORT.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘E2E\_PORT\_KEY\_PORT\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘E2E\_PORT\_KEY\_PORT\_IDENTIFIER’ |
| - | Value | Take the ‘E2E\_PORT\_KEY\_PORT\_IDENTIFIER’ from corresponding phy\_port (via id\_phy\_port)  Note: don’t create the identifier if it is not populated |

</286278>

Database traversals:

*(PVC “l” side [“customer/site” on the “l” side])*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| PVC.pvc\_id | <LOG-PORT-EKT-FIX> LOG\_PORT\_CE\_INET\_VLAN\_VW.pvc\_id |
| SITE.site\_id | <LOG-PORT-EKT-FIX> LOG\_PORT\_CE\_INET\_VLAN\_VW.site\_id |
| SITE.site\_id | PORT\_ASGMT.site\_id |
| <LOG-PORT-EKT-FIX> LOG\_PORT\_CE\_INET\_VLAN\_VW.pvc\_id | PVC\_EXTENSION.pvc\_id (use “outer” join here at this side !) |

Database constraints:

*(PVC “l” side [“customer/site” on the “l” side])*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| <LOG-PORT-EKT-FIX> LOG\_PORT\_CE\_INET\_VLAN\_VW | SIDE | ‘local’ |
| <LOG-PORT-EKT-FIX> LOG\_PORT\_CE\_INET\_VLAN\_VW | CUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) |
| PVC | PVC\_RCUST\_ID | = 395 (i.e. is “AT&T” customer id) |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |

<see also logic under:   
Logic for identifying a “customer” side in the ICORE.PVC table>

Database traversals:

*(PVC “r” side [“customer/site” on the “r” side])*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| PVC.pvc\_id | <LOG-PORT-EKT-FIX> LOG\_PORT\_CE\_INET\_VLAN\_VW.pvc\_id |
| SITE.site\_id | <LOG-PORT-EKT-FIX> LOG\_PORT\_CE\_INET\_VLAN\_VW.site\_id |
| SITE.site\_id | PORT\_ASGMT.site\_id |
| PVC.pvc\_id | PVC\_EXTENSION.pvc\_id (use “outer” join here at this side !) |

Database constraints:

*(PVC “r” side [“customer/site” on the “r” side])*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| <LOG-PORT-EKT-FIX> LOG\_PORT\_CE\_INET\_VLAN\_VW | SIDE | ‘remote’ |
| <LOG-PORT-EKT-FIX> LOG\_PORT\_CE\_INET\_VLAN\_VW | CUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) |
| PVC | PVC\_LCUST\_ID | = 395 (i.e. is “AT&T” customer id) |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |

<see also logic under:   
Logic for identifying a “customer” side in the ICORE.PVC table>

<BEGIN: Port Load Corrections>

**<no longer used; 2014-06-25; hc2164>**

~~Database traversals (for getting “cos\_profile” data):~~

|  |  |
| --- | --- |
| **~~From table.column~~** | **~~To table.column~~** |
| ~~SITE.site\_id~~ | ~~SERVICE\_ASGMT.site\_id~~ |
| ~~SERVICE\_OPTION.serv\_opt\_id~~ | ~~SERVICE\_ASGMT.serv\_opt\_id~~ |
| ~~SERVICE\_OPTION.serv\_id~~ | ~~SERVICE.serv\_id~~ |

~~Database constraints (for getting “cos\_profile” data):~~

|  |  |  |
| --- | --- | --- |
| **~~Table name~~** | **~~Column name~~** | **~~Constraint(s)~~** |
| ~~SERVICE~~ | ~~SERV\_NAME~~ | ~~COS PACKAGE~~ |

<END: Port Load Corrections>

<BEGIN: Port Load Corrections>

**The following is on pulling ip address data from INSTAR via the ICORE table “ICORE.map\_instar\_port”.**

Database traversals (for ipv4 address data):

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| **ICORE**.MAP\_INSTAR\_PORT.pvc\_id | **ICORE**.PVC.pvc\_id |
| **ICORE**.MAP\_INSTAR\_PORT.instar\_port\_asgmt | **INSTAR**.IP\_PORT\_ASGMT.ip\_port\_asgmt\_id |
| **INSTAR**.IP\_PORT\_ASGMT.cr\_addr\_id | **INSTAR**.SERIAL\_IP\_ADDR.serial\_ip\_addr\_id |

Database traversals (for ipv6 address data, *first attempt*):

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| **ICORE**.MAP\_INSTAR\_PORT.pvc\_id | **ICORE**.PVC.pvc\_id |
| **ICORE**.MAP\_INSTAR\_PORT.instar\_port\_asgmt | **INSTAR**.IP\_PORT\_ASGMT.ip\_port\_asgmt\_id |
| **INSTAR.**IP\_PORT\_ASGMT.sdid | **INSTAR.**IPV6\_PORT\_ASGMT\_MAP.sdid |
| **INSTAR.**IPV6\_PORT\_ASGMT\_MAP.ipv6\_link\_ip\_id | **INSTAR.**IPV6\_ASSIGNED\_LINK\_IPS.ipv6\_link\_ip\_id |
| **INSTAR.**IPV6\_PORT\_ASGMT\_MAP.ipv6\_address\_type\_id | **INSTAR.**IPV6\_ADDRESS\_TYPE.ipv6\_address\_type\_id |

Database traversals (for ipv6 address data, *second attempt*):

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| **ICORE**.MAP\_INSTAR\_PORT.pvc\_id | **ICORE**.PVC.pvc\_id |
| **ICORE**.MAP\_INSTAR\_PORT.instar\_port\_asgmt | **INSTAR**.IP\_PORT\_ASGMT.ip\_port\_asgmt\_id |
| **INSTAR.**IP\_PORT\_ASGMT.sdid | **INSTAR.**IPV6\_PORT\_ASGMT\_MAP.sdid |
| **INSTAR.**IPV6\_PORT\_ASGMT\_MAP.ipv6\_link\_ip\_id | **INSTAR.**IPV6\_CUST\_LINK\_IPS.ipv6\_cust\_link\_ip\_id |
| **INSTAR.**IPV6\_PORT\_ASGMT\_MAP.ipv6\_address\_type\_id | **INSTAR.**IPV6\_ADDRESS\_TYPE.ipv6\_address\_type\_id |

<END: Port Load Corrections>

**(B) PE Logical Port**

The “PE Port” data is created starting from the same record of the ICORE.SITE table that corresponds to the record from ICORE.SITE for the “CE Port”.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Icore table.column** | | **GDB LOG\_PORT column** | | **Processing comment** | |
| <BEGIN: Port Load Corrections> |  | |  | |
| *[Using “l” side]* PVC.pvc\_id + PVC.pvc\_lsite\_id + PVC.pvc\_lcust\_port;  *[Using “r” side]* PVC.pvc\_id + PVC.pvc\_rsite\_id + PVC.pvc\_rcust\_port;  *NOTE: For the EKT use the following corresponding values:*  LOG\_PORT\_PE\_VW.pvc\_id + LOG\_PORT\_PE\_VW.site\_id + LOG\_PORT\_PE\_VW.cust\_port; | | id [via Enterprise Key Translation] | | Use SOURCE\_KEY.id into GDB | |
| <END: Port Load Corrections> |  | |  | |
| - | | id\_change\_tracking | | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] | |
| - | | id\_object\_type | | Referencing OBJECT\_TYPE ‘LOG\_PORT’ | |
| - | | id\_port\_type | | Referencing PORT\_TYPE ‘PE’ | |
| - | | is\_read\_only | | ‘Y’ | |
| - | | instar\_site\_id\_value | | Set to NULL | |
| EQUIPMENT.site\_id | | icore\_site\_id\_value | |  | |
| PVC.pvc\_id | | icore\_pvc\_id\_value | |  |
| SITE.site\_id | | icore\_ce\_port\_site\_id\_value | |  | |
| EQUIPMENT.site\_id | | id\_site\_inventory\_site\_rep  [via Enterprise Key Translation for “ICORE.SITE.site\_id” (== “ICORE.equipment.site\_id”)] | | Use SOURCE\_KEY.id into GDB  (if existing; otherwise set to NULL) | |
| - | | id\_asset\_equipment | | (NULL; will be set later) | |
| <BEGIN: Port Load Corrections> |  | |  | |
| *[“customer/site” on the “l” side]* PVC.pvc\_**r**proto;  *[“customer/site” on the “r” side]* PVC.pvc\_**l**proto; | | protocol | | **NOTE**: “non-customer” (i.e. “provider”) side is used ! | |
| *[“customer/site” on the “l” side]* PVC.pvc\_**r**contrcir;  *[“customer/site” on the “r” side]* PVC.pvc\_**l**contrcir; | | speed | | **NOTE**: “non-customer” (i.e. “provider”) side is used ! | |
| <END: Port Load Corrections> | |  | |  | |
| - | | mac\_address | | Set to NULL | |
| <BEGIN: Port Load Corrections> | |  | |  | |
| - | | cos\_profile | | Set to NULL (no longer used; 2014-06-24) | |
| <END: Port Load Corrections> | |  | |  | |
| PORT\_ASGMT.cust\_port | | id\_phy\_port | | Use SOURCE\_KEY.id into GDB which populated “GDB.PHY\_PORT.id” with “id\_port\_type” referencing “PE” (which must exist; otherwise set to NULL !) | |

NOTE that no identifier data is processed here, as for “Internet VLAN” logical PE ports no “identifier” values can be identified !

<BEGIN: Port Load Corrections>

Database traversals:

*(PVC “l” side [“customer/site” on the “l” side])*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| PVC.pvc\_id | LOG\_PORT\_PE\_VW.pvc\_id |
| SITE.site\_id | LOG\_PORT\_PE\_VW.site\_id |
| PORT\_ASGMT.cust\_port | LOG\_PORT\_PE\_VW.cust\_port |
| PVC.pvc\_id | NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id |
| SITE.site\_id | NETWORK\_CONNECTION\_INET\_VLAN.site\_id |
| SITE.site\_id | PORT\_ASGMT.site\_id |
| PORT\_ASGMT.equip\_id | EQUIPMENT.equip\_id |

<END: Port Load Corrections>

Database constraints:

*(PVC “l” side [“customer/site” on the “l” side])*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| NETWORK\_CONNECTION\_INET\_VLAN | SIDE | ‘local’ |
| NETWORK\_CONNECTION\_INET\_VLAN | CUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) |
| PVC | PVC\_RCUST\_ID | = 395 (i.e. is “AT&T” customer id) |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |

<BEGIN: Port Load Corrections>

Database traversals:

*(PVC “r” side [“customer/site” on the “r” side])*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| PVC.pvc\_id | LOG\_PORT\_PE\_VW.pvc\_id |
| SITE.site\_id | LOG\_PORT\_PE\_VW.site\_id |
| PORT\_ASGMT.cust\_port | LOG\_PORT\_PE\_VW.cust\_port |
| PVC.pvc\_id | NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id |
| SITE.site\_id | NETWORK\_CONNECTION\_INET\_VLAN.site\_id |
| SITE.site\_id | PORT\_ASGMT.site\_id |
| PORT\_ASGMT.equip\_id | EQUIPMENT.equip\_id |

<END: Port Load Corrections>

Database constraints:

*(PVC “r” side [“customer/site” on the “r” side])*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| NETWORK\_CONNECTION\_INET\_VLAN | SIDE | ‘remote’ |
| NETWORK\_CONNECTION\_INET\_VLAN | CUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) |
| PVC | PVC\_LCUST\_ID | = 395 (i.e. is “AT&T” customer id) |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |

<see also logic under:   
Logic for identifying a “customer” side in the ICORE.PVC table>

**Data Blocking (1)**

Do not process port records (SITE) where the record returned by the logic below is linked to a GDB.SERVICE\_TYPE having a record in GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM for Icore:

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| NETWORK\_CONNECTION\_INET\_VLAN.site\_id | SITE.site\_id |
| SERVICE\_ASGMT.site\_id | SITE.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |
| GDB.SERVICE\_TYPE\_NOTATION.service\_name | SERVICE.serv\_name |
| GDB.SERVICE\_TYPE\_NOTATION.id\_service\_type | GDB.SERVICE\_TYPE.id |
| GDB.SERVICE\_TYPE\_BLOCKED\_SYSTEM.id\_service\_type | GDB.SERVICE\_TYPE.id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| SERVICE\_TYPE\_BLOCKED\_SYSTEM | BLOCKED\_SYSTEM | 13609 |

*Note:*

*The BLOCKED\_SYSTEM constraint value is the MOTS-ID of the system.*

**Data Blocking (2)**

Do not process ICORE logical port records where the special logic for blocking INSTAR items hold.

The “<input site id variable (dep. on current scenario)>” here is “SITE. site\_id”.

See “Logic for excluding INSTAR items”.

#### HLD-254035-GCP-FLOW-ICORE-120-D [Icore Data Flow into Log Port SITE (“Interface Name” data)]

**NOTE: This is a “pseudo-requirement” not to be included in any traceability files.**

NOTE: This requirement will no longer be implemented, as we plan to logically replace ICORE ports by their “master” ports from other port data source systems (like NC3) via “port unification”.

~~This section is on loading “Interface Name” data for logical port data from ICORE where the “Interface Name” is stored in NC3 (for AVPN & EVPN). This is done by mapping via the ICORE SITE.site\_id which implies that it is limited to “CE” ports.~~

~~Before executing this step the three steps “HLD-254035-GCP-FLOW-ICORE-110-A [Icore Data Flow into Log Port SITE (Layer-3 “Log Port”)]”, “HLD-254035-GCP-FLOW-ICORE-110-B [Icore Data Flow into Log Port SITE (Layer-2 “Log Port”)]” and “HLD-254035-GCP-FLOW-ICORE-110-C [Icore Data Flow into Log Port SITE (Internet VLAN “Log Port”)]” must have been completed.~~

~~The data loading needs to be done by “updating” already existing rows of the “GDB.log\_port” table.~~

~~Implement the following:~~

~~Use the following GDB logical port object as having been loaded from ICORE:~~

|  |  |  |
| --- | --- | --- |
| **~~ICORE table.column~~** | **~~GDB LOG\_PORT column~~** | **~~Processing comment~~** |
| ~~SITE.site\_id~~ | ~~id [via Enterprise Key Translation]~~ | ~~Use “GDB.LOG\_PORT.id” (where GDB.LOG\_PORT.id\_port\_type references “~~**~~CE~~**~~”) to identify the appropriate record [via Enterprise Key Translation]~~ |

~~Update the GDB port object using the following NC3 records:~~

|  |  |  |
| --- | --- | --- |
| **~~NC3 table.column~~** | **~~GDB PORT column~~** | **~~Processing comment~~** |
| ~~NC3.port.common\_name~~ | ~~name~~ | ~~See “Logic for associating NC3 “Interface Name” data to ICORE and GDB “port” data” for details~~ |

##### ~~Logic for associating NC3 “Interface Name” data to ICORE and GDB “port” data~~

~~The logic for associating NC3 “Interface Name” data to ICORE and GDB “port” data is the same as the one used under “Logic for mapping ports, pvcs and (managed NC3) assets”.~~

~~Therefore that logic should be used for retrieving the “Interface Name” data. The logic works in the same way, but instead of creating an association when the two searched items have been found, here the NC3 port interface name (“port\_interface\_name” in the SQL statement) is used to set the column “name” for the corresponding port object.~~

***~~HINT~~****~~: For performance reasons the implementation for both this item and the item creating the association (see “~~*~~HLD-254035-GCP-FLOW-ICORE-110-D [Icore Data Flow into Log Port SITE (“Interface Name” data)]~~*~~”)can be combined, so that both items are executed in one implementation step.~~*

#### HLD-254035-GCP-FLOW-ICORE-120-E [Icore Data Flow into Log Port SITE (“id\_asset Equipment” data)]

**NOTE: This is a “pseudo-requirement” not to be included in any traceability files.**

NOTE: This requirement will no longer be implemented, as we plan to logically replace ICORE ports by their “master” ports from other port data source systems (like NC3) via “port unification”.

~~This requirement is for covering the relationship between (managed) equipment assets (from “NC3” resp. “LPP-CPE”) and CE logical ports associated with them.~~

~~Before executing this step the three steps “HLD-254035-GCP-FLOW-ICORE-110-A [Icore Data Flow into Log Port SITE (Layer-3 “Log Port”)]”, “HLD-254035-GCP-FLOW-ICORE-110-B [Icore Data Flow into Log Port SITE (Layer-2 “Log Port”)]” and “HLD-254035-GCP-FLOW-ICORE-110-C [Icore Data Flow into Log Port SITE (Internet VLAN “Log Port”)]” must have been completed.~~

~~The data loading needs to be done by “updating” already existing rows of the “GDB.log\_port” table.~~

~~Implement the following:~~

~~Use the following GDB logical port object as having been loaded from ICORE:~~

|  |  |  |
| --- | --- | --- |
| **~~ICORE table.column~~** | **~~GDB LOG\_PORT column~~** | **~~Processing comment~~** |
| ~~SITE.site\_id~~ | ~~id [via Enterprise Key Translation]~~ | ~~Use “GDB.LOG\_PORT.id” (where GDB.LOG\_PORT.id\_port\_type references “~~**~~CE~~**~~”) to identify the appropriate record [via Enterprise Key Translation]~~ |

~~Update the GDB logical port object using the following NC3 record:~~

|  |  |  |
| --- | --- | --- |
| **~~NC3 table.column~~** | **~~GDB LOG\_PORT column~~** | **~~Processing comment~~** |
| ~~NC3.eqp.odbid~~ | ~~id\_asset\_equipment~~  ~~[via Enterprise Key Translation; see “Logic for mapping ports, pvcs and (managed NC3) assets” for obtaining the value for “NC3.eqp.odbid”]~~ | ~~See “Logic for mapping ports, pvcs and (managed NC3) assets” for details~~ |

##### ~~Logic for mapping ports, pvcs and (managed NC3) assets~~

~~In order to be able to map NC3 (managed) equipment data to ICORE port data, the PVC data available in both ICORE and NC3 are needed as a “bridge”.~~

~~The logic to be implemented will provide these data items:~~

* ~~the NC3 equipment “odbid” value   
  (“device\_odbid\_eqp” in the SQL statement below)~~
* ~~the NC3 and ICORE pvc id  
  (“pvc\_id” in the SQL statement below)~~
* ~~the NC3 port interface name  
  (“port\_interface\_name” in the SQL statement below)  
  [NOTE:   
  This item is not needed for “mapping”, but is easily available by this logic and can be reused for setting the “GDB.phy/log\_port.name”]~~

~~With the data from above, two items need to be found:~~

* ~~the CE port (no PE port data in NC3 exists) belonging to the “pvc id”;  
  this can be done using the “ICORE.network\_connection” view with “pvc id”  
  checking whether the port’s site\_id (SITE.site\_id) value is among the returned site id values;~~
* ~~the NC3 asset belonging to the “odbid” value;  
  this can be done using the Enterprise Key Translation (EKT) for the given “odbid”;~~

~~If the two items have been found, then the association can be created (see the table above).~~

~~The logic to be applied for getting the “bridge” data is contained in the embedded SQL file. The SQL statement is complex and executes quite slowly, so it may be necessary to “stage” that “bridge” data or “decompose” the SQL statement for appropriate performance.~~

***~~NOTE~~*** *~~that the SQL statement must not be taken “as is”, as it contains timing statements and some PVC ID values for testing purposes !~~*

~~~~

<LOG-PORT-EKT-FIX> (BEGIN)

#### HLD-LOG-PORT-EKT-FIX-GCP-FLOW-ICORE-120-F [Icore Data Flow into “CE” Log Port <STATIC DATA CLEANSING and RELOADING>]

This requirement is for cleansing ICORE “CE” “log\_port” port objects and for reloading them, as their EKT value needs to be created via the corr. “LOG\_PORT\_CE\_\*” view (and no longer via the corr. “NETWORK\_CONNECTION\*” view).

This is a one-time-only requirement.

The execution of this process requires the maintenance of a change tracking event into GDB.CHANGE\_TRACKING, CHANGE\_SYSTEM and CHANGE\_USER. The change tracking event for the service delivery data flow must use CHANGE\_SYSTEM.name = ’ICORE\_TO\_GDB’ and CHANGE\_USER.name = ’ICORE\_TO\_GDB’. It is sufficient to have one change tracking event for all affected records during a single execution of this process.

NOTE that this requirement is a “one-time” delete and (then) load.

(1) Deleting ICORE “CE” log port objects from GDB

The set of GDB log port objects for which this delete applies is as follows  
(all “CE” log port objects loaded from ICORE):

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| GDB.PORT\_TYPE.ID | GDB.LOG\_PORT.ID\_PORT\_TYPE |
| METADATA.SOURCE\_KEY.id | GDB.LOG\_PORT.id |
| METADATA.SOURCE\_KEY.id\_meta\_table | METADATA.META\_TABLE.id |
| METADATA.META\_SYSTEM.id | METADATA.META\_TABLE.id\_meta\_system |

Database constraints (1):

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| GDB.PORT\_TYPE | TYPE | CE |

Database constraints (2):

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| METADATA.META\_SYSTEM | NAME | ICORE |

All “CE” log port objects as identified by the logic given above shall be deleted.

(2) Reloading ICORE “CE” log port objects into GDB

After the ICORE “CE” log port objects have been deleted from GDB, they shall be reloaded again, but this time by making use of the new EKT views “LOG\_PORT\_CE\_\*”, as described under the updated requirements:

HLD-254035-GCP-FLOW-ICORE-120-A [Icore Data Flow into Log Port SITE (Layer-3 “Log Port”)]

HLD-254035-GCP-FLOW-ICORE-120-B [Icore Data Flow into Log Port SITE (Layer-2 “Log Port”)]

HLD-254035-GCP-FLOW-ICORE-120-C [Icore Data Flow into Log Port SITE (Internet VLAN “Log Port”)]

(3) Reloading ICORE “CE” log port object associations to network connections

After the ICORE “CE” log port objects have been reloaded into GDB, they shall be associated with their corresponding network connections again, but this time by making use of the new EKT views “LOG\_PORT\_CE\_\*”, as described under the updated requirements:

HLD-254035-GCP-FLOW-ICORE-207 [Icore Data Flow into Log Port (“Network Connection Port”) to Asset]

(4) Recreating unifications into ICORE “CE” log port objects

After the 3 steps above have been executed, all log port unifications into the newly reloaded ICORE “CE” log port objects shall be recreated.

<LOG-PORT-EKT-FIX> (END)

#### HLD-293772-CR160298-GCP-FLOW-USRP-120-G [Icore Data Flow into Log Port SITE (Tunnel Vlan “Log Port”)]

This section is loading logical port data from USRP for Tunnel Vlan. Ports are an object type of their own in GDB and are not classified under the “Asset” object type.

*NOTE: This requirement needs to be executed after requirement “HLD-254035-GCP-FLOW-ICORE-101 [Icore Data Flow into Site]” has been completed, so the value for “id\_site\_inventory\_site\_rep” will already be available.*

*NOTE: This requirement needs to be executed after all “phy port” related requirements have been completed, so “log ports” can be associated with their “phy ports”.*

Implement the following:

Process the following Icore records into the GDB log\_port object:

**(A) CE Logical Port**

The “CE Port” data is created starting from a record of the USRP.logical\_port table. Ensure the corresponding ete\_svc\_conn\_key from USRP.pvc table satisfies:

ete\_svc\_conn\_key is null or not exists ( --- check if same ete\_svc\_conn\_key is loaded from IEOD or not.

select 1

from gdb.ete\_key\_map

where internal\_key\_value = a.ete\_svc\_conn\_key --- only if it is populated

and ete\_key\_source = 'IEOD' and rownum<2

)

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT column** | **Processing comment** |
| usrp.logical\_port.log\_port\_id | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘LOG\_PORT’ |
| - | id\_port\_type | Referencing PORT\_TYPE ‘CE’ |
| - | is\_read\_only | ‘Y’ |
| - | instar\_site\_id\_value | Set to NULL |
| usrp.logical\_port.icore\_site\_id | icore\_site\_id\_value |  |
| usrp.pvc.icore\_pvc\_id | icore\_pvc\_id\_value |  |
| usrp.logical\_port.icore\_site\_id | icore\_ce\_port\_site\_id\_value |  |
| usrp.logical\_port.icore\_site\_id | id\_site\_inventory\_site\_rep  [via Enterprise Key Translation for “SITE.site\_id”] | Use SOURCE\_KEY.id into GDB |
| - | id\_asset\_equipment | (NULL; will be set later) |
| *-* | protocol | Set to NULL |
| usrp.logical\_port.port\_speed | speed |  |
| - | name | Set to NULL |
| - | mac\_address | Set to NULL |
| - | cos\_profile | Set to NULL (no longer used; 2014-06-24) |
| usrp.logical\_port.icore\_site\_id  (same as icore.site.site\_id) | id\_phy\_port | Use SOURCE\_KEY.id into GDB which populated “GDB.PHY\_PORT.id” with “id\_port\_type” referencing “CE” (which must exist; otherwise set to NULL !) |

**Port Identifier (1): ete\_port\_key:** only if it is populated

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER column** | **Processing comment** |
| - | id\_log\_port | LOG\_PORT.id |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘E2E\_PORT\_KEY\_PORT\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOG\_PORT\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_log\_port\_identifier | LOG\_PORT\_IDENTIFIER.id |
| - | id\_change\_tracking | LOG\_PORT\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘E2E\_PORT\_KEY\_PORT\_IDENTIFIER’ |
| - | Value | Take the ‘E2E\_PORT\_KEY\_PORT\_IDENTIFIER’ from corresponding phy\_port (via id\_phy\_port)  Note: don’t create the identifier if it is not populated |

**(B) PE Logical Port**

|  |  |  |  |
| --- | --- | --- | --- |
| **USRP table.column** | **GDB LOG\_PORT column** | **Processing comment** | |
| usrp.TUNNEL\_VLAN\_PE\_PORT\_VW.log\_port\_id | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB | |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] | |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘LOG\_PORT’ | |
| - | id\_port\_type | Referencing PORT\_TYPE ‘PE’ | |
| - | is\_read\_only | ‘Y’ | |
| - | instar\_site\_id\_value | Set to NULL | |
| - | icore\_site\_id\_value | Set to NULL | |
| usrp.pvc.pvc\_id | icore\_pvc\_id\_value |  |
| Usrp.logical\_port.icore\_site\_id | icore\_ce\_port\_site\_id\_value |  | |
| EQUIPMENT.site\_id | id\_site\_inventory\_site\_rep  [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB  (if existing; otherwise set to NULL)  Set to NULL | |
| - | id\_asset\_equipment | (NULL; will be set later) | |
| *-* | protocol | **NULL** | |
| usrp.TUNNEL\_VLAN\_PE\_PORT\_VW.port\_speed | speed | **NOTE**: “non-customer” (i.e. “provider”) side is used ! | |
| - | name | Set to NULL | |
| - | mac\_address | Set to NULL | |
| - | cos\_profile | Set to NULL (no longer used; 2014-06-24) | |
| - | id\_phy\_port | Set to NULL | |

#### HLD-PHY-PORT-FIX-GCP-FLOW-ICORE-120-H [ICORE Data Flow into PHY-PORT]

~~<Redesign Managed Equip>~~

~~In order to load the right value into GDB.PHY\_PORT table under ID\_ASSET\_EQUIPMENT column, we should create a batch process which will follow the below steps:~~

* 1. ~~Once the port unification is complete by the existing process, we need to identify the records in GDB.PHY\_PORT table which has the serv\_opt\_name as ‘SO\_MANAGED\_INDICATOR’ in SERV\_OPT\_TYPE table from following association and PHY\_PORT.ID\_ASSET\_EQUIPMENT is NULL:~~

~~Asset (Access\_Circuit) -> Service\_Option~~

* 1. ~~For all the found records, we need to use the below logic to identify whether the equipment value is present in NC3 or not. If present, we need to update the same in GDB.PHY\_PORT.ID\_ASSET\_EQUIPMENT.~~

~~SELECT DISTINCT er.odbid,er.common\_name,~~

~~p2.value,~~

~~cu3.point\_type~~

~~FROM NC3\_MAT.PARM p1,~~

~~NC3\_MAT.PHY\_CIRCUIT pc,~~

~~NC3\_MAT.CMP\_USAGE cu1,~~

~~NC3\_MAT.CMP\_USAGE cu2,~~

~~NC3\_MAT.PORT po,~~

~~NC3\_MAT.EQP ec,~~

~~NC3\_MAT.EQP er,~~

~~NC3\_MAT.EQP\_REF err,~~

~~NC3\_MAT.PARM p2,~~

~~NC3\_MAT.CMP\_USAGE cu3,~~

~~NC3\_MAT.CONFIG c~~

~~WHERE p1.value IN <ICORE SITE ID>~~

~~AND p1.internal\_label = 'ICORE\_SITE\_ID'~~

~~AND p1.stop\_date > CURRENT\_DATE~~

~~AND pc.odbid = p1.odbid\_cmponent~~

~~AND pc.odbid = cu1.odbid\_cmponent~~

~~AND cu1.stop\_date > CURRENT\_DATE~~

~~AND cu1.odbid\_config = cu2.odbid\_config~~

~~AND cu2.stop\_date > CURRENT\_DATE~~

~~AND cu2.odbid\_cmponent = po.odbid~~

~~AND po.odbid\_eqp = ec.odbid~~

~~AND ec.odbid\_main\_eqp = er.odbid~~

~~AND ec.contain\_stop\_date > CURRENT\_DATE~~

~~AND er.odbid\_eqp\_ref = err.odbid~~

~~AND err.eqp\_category = 'Router'~~

~~AND cu3.odbid\_cmponent = er.odbid~~

~~AND cu3.stop\_date > CURRENT\_DATE~~

~~AND c.odbid =cu3.odbid\_config~~

~~AND p2.odbid\_cmponent =c.odbid~~

~~AND p2.internal\_label ='SITE\_BACKUP\_TYPE'~~

~~AND p2.stop\_date > CURRENT\_DATE~~

~~AND pc.stop\_date > CURRENT\_DATE~~

~~AND po.stop\_date > CURRENT\_DATE~~

~~AND ec.stop\_date > CURRENT\_DATE~~

~~AND er.stop\_date > CURRENT\_DATE~~

~~AND err.stop\_date > CURRENT\_DATE~~

~~AND c.stop\_date > CURRENT\_DATE ;~~

* 1. ~~Pick the corresponding ODBID and identify the Asset.ID via Enterprise Key Translation and load that value in PHY\_PORT.ID\_ASSET\_EQUIPMENT.~~

~~</Redesign Managed Equip>~~

#### HLD-254035-GCP-FLOW-ICORE-130 [Icore Data Flow into Service]

Implement the following:

Process the following Icore records into the GDB service object and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB SERVICE\_TYPE\_NOTATION column** | **Processing comment** |
| - | id | Create a new unique “id” value using the "SERVICE\_TYPE\_NOTATION\_ID" sequence |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| SERVICE.serv\_name | Service\_name |  |
| - | Id\_service\_type | NULL if not already pre-populated (ref. HLD-254035-GCP-GDB-for-ICORE-001 [SERVICE\_TYPE Content and Mapping]) |

Make sure no duplicate “service\_name” entries are created in “GDB.SERVICE\_TYPE\_NOTATION”.

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB SERVICE column** | **Processing comment** |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘SERVICE’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
|  | Id [via Enterprise Key Translation] | New primary key from Enterprise Key Translation.  Use SOURCE\_KEY.id into GDB  {More details found below} |
| - | Id\_service\_type\_notation | SERVICE\_TYPE\_NOTATION.id |
| - | Id\_service\_type | SERVICE\_TYPE\_NOTATION.id\_service\_type  (presents reference to SERVICE\_TYPE.id (if existing), or NULL) |

For the above MS-WORD table, for each row inserted into “GDB.SERVICE\_TYPE\_NOTATION” (and for cases where the corresponding data already exists in GDB.SERVICE\_TYPE\_NOTATION, because it has been prepopulated !) a corresponding row in “GDB.SERVICE” needs to be created, unless it already exists. The latter (it already exists) is the case, if a row exists in GDB.SERVICE   
where (Id\_service\_type\_notation = SERVICE\_TYPE\_NOTATION.id)  
and (<current source system service name> = SERVICE\_TYPE\_NOTATION.service\_name)  
and ((Id\_service\_type = SERVICE\_TYPE\_NOTATION.id\_service\_type)  
 or (Id\_service\_type = NULL))

For setting “GDB.SERVICE.id” there is a special setup in the EKT system (METADATA schema) for creating EKT values for “GDB.SERVICE.id”. It works as follows:

So use

* meta\_system.name = ‘GDB’
* meta\_table.name = ‘SERVICE’
* source\_key.num\_source\_key\_values = 0

Use that as the input into the EKT system to get a new Enterprise Key Value

(here: as a new primary key value for “GDB.SERVICE.id”).

**Hint**: For the above the same process already exists for INSTAR (“HLD-232213e-GCP-FLOW-INSTAR-103 [Instar Data Flow into Service]”, implemented by the GRDB ETL team); that already existing process can be used for all needed details.

#### HLD-254035-GCP-FLOW-ICORE-200-A [Icore Data Flow into Site to Organization (via “CUSTOMER”)]

Implement the following:

Process the following Icore records into the GDB site to organization association and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**SITE**’, OBJECT\_TYPE\_TO ‘**ORGANIZATION**’, ROLE with FUNCTION\_TYPE ‘**CONTRACTED\_BY**’ and FUNCTION\_ROLE NULL |
| SITE.site\_id | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective SITE.id |
| SITE.cust\_id (=CUSTOMER.cust\_id)  PID : 302503  Or  customer\_ubsubaccount\_vw.site\_id | Id\_object\_to [via Enterprise Key Translation for CUSTOMER.cust\_id] | Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id  Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules] where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id |
| SITE.site\_id | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective SITE.id |
| - | Gdb\_internal\_flags | NULL |

*Note:*

Make sure an association is created only if an entry exists in GDB for both the “id\_object\_what” and the “id\_object\_to” !

*Note:*

*This step requires that the following steps have been executed:*

1. *HLD-254035-GCP-FLOW-ICORE-100 [Icore Data Flow into Organization CUSTOMER]*
2. *HLD-254035-GCP-FLOW-ICORE-101 [Icore Data Flow into Site]*

#### HLD-254035-GCP-FLOW-ICORE-200-B [Icore Data Flow into Site to Organization (via “MRC\_GRC\_SOC\_SOURCE”)]

Implement the following:

Process the following Icore records into the GDB site to organization association and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

NOTE that here more than one association record (having different “MCN\_GRC\_SOC\_SOURCE.id” values) may be possible !

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**SITE**’, OBJECT\_TYPE\_TO ‘**ORGANIZATION**’, ROLE with FUNCTION\_TYPE ‘**CONTRACTED\_BY**’ and FUNCTION\_ROLE NULL |
| SITE.site\_id | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective SITE.id |
| MCN\_GRC\_SOC\_SOURCE.id (via database traversal given below) | Id\_object\_to [via Enterprise Key Translation for MCN\_GRC\_SOC\_SOURCE.id] | Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id  Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules] where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id |
| SITE.site\_id | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective SITE.id |
| - | Gdb\_internal\_flags | NULL |

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SITE.site\_id | MCN\_GRC\_SOC\_SITE\_IDS.site\_id |
| MCN\_GRC\_SOC\_SITE\_IDS.id\_m\_g\_s\_source | MCN\_GRC\_SOC\_SOURCE.id |

**Constraints**:

Only take into account “MCN\_GRC\_SOC\_SOURCE.id” values for which all three components “MCN”, GRC” and “SOC” in the corresponding record are defined, i.e. all three values are not equal to “<<NONE>>”.

*Note:*

Make sure an association is created only if an entry exists in GDB for both the “id\_object\_what” and the “id\_object\_to” !

*Note:*

*This step requires that the following steps have been executed:*

1. *HLD-254035-GCP-FLOW-ICORE-100-B [Icore Data Flow into Organization MCN\_GRC\_SOC\_SOURCE]*
2. *HLD-254035-GCP-FLOW-ICORE-101 [Icore Data Flow into Site]*

#### HLD-254035-GCP-FLOW-ICORE-201-A [Icore Data Flow into Asset (“Access Circuit”) to Site]

Implement the following:

Process the following Icore records into the GDB asset to site association and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**SITE**’, ROLE with FUNCTION\_TYPE ‘**PART\_OF**’ and FUNCTION\_ROLE NULL |
| CUST\_ACCESS.site\_id <or> (CUST\_ACCESS\_CKT.site\_id + CUST\_ACCESS\_CKT.seq\_no) | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id |
| CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id   (=SITE.site\_id) | Id\_object\_to [via Enterprise Key Translation for SITE.site\_id] | Use SOURCE\_KEY.id which populated the respective SITE.id |
| CUST\_ACCESS.site\_id <or> (CUST\_ACCESS\_CKT.site\_id + CUST\_ACCESS\_CKT.seq\_no) | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id |
| - | Gdb\_internal\_flags | NULL |

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

<286284-US600815-US600161>

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB SITE\_IDENTIFIER column** | **Processing comment** |
| - | id\_site | Id\_object\_to from above |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘E2E\_SITE\_KEY\_SITE\_IDENTIFIER’ |

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB SITE\_IDENTIFIER\_VALUE column** | **Processing comment** |
| - | id\_site\_identifier | SITE\_IDENTIFIER.id |
| - | id\_change\_tracking | SITE\_IDENTIFIER.id\_change\_tracking |
| - | id\_identifier\_type | Referencing IDENTIFIER\_TYPE ‘E2E\_SITE\_KEY\_SITE\_IDENTIFIER’ |
|  | Value | <286284-US600815-US600161>  See End-to-End Key Search and Generation [ Icore Data Flow ]  Note: Id\_object\_what is the asset.id(ACCESS\_CIRCUIT) |

Update GDB.SITE\_EXT as below:

|  |  |  |
| --- | --- | --- |
| **GRID table.column** | **GDB SITE\_EXT column** | **Processing comment** |
| - | id\_site | Id\_object\_to |
| - | ETE\_SITE\_KEY | <286284-US600815-US600161>  See End-to-End Key Search and Generation [ Icore Data Flow ] |
|  | ETE\_KEY\_SOURCE | <286284-US600815-US600161>  See End-to-End Key Search and Generation [ Icore Data Flow ] |

</286284-US600815-US600161>

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that the following steps have been executed:*

1. *HLD-254035-GCP-FLOW-ICORE-101 [Icore Data Flow into Site]*
2. *HLD-254035-GCP-FLOW-ICORE-102-A [Icore Data Flow into Asset CUST\_ACCESS (“Access Circuit”)]*
3. *HLD-254035-GCP-FLOW-ICORE-102-B [Icore Data Flow into Asset CUST\_ACCESS\_CKT (“Access Circuit”)]*

#### HLD-254035-GCP-FLOW-ICORE-201-B [Icore Data Flow into Asset (“Customer Network”) to Site]

**NOTE: This is a “pseudo-requirement” not to be included in any traceability files.**

NOTE: For assets of type “Customer Network” a concept of “site” or “location” does not exist, so no corresponding association will be created here.

#### HLD-254035-GCP-FLOW-ICORE-201-C [Icore Data Flow into Asset (“Network Connection”) to Site]

Implement the following:

**(1) “Layer-2” and “Layer-3” “Network Connections”**

For “Layer-2” and “Layer-3” “Network Connections” process the following Icore records into the GDB asset to site association and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**SITE**’, ROLE with FUNCTION\_TYPE ‘**PART\_OF**’ and FUNCTION\_ROLE NULL |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.**site\_id**;  (=SITE.site\_id) | Id\_object\_to [via Enterprise Key Translation for SITE.site\_id] | Use SOURCE\_KEY.id which populated the respective SITE.id |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id |
| - | Gdb\_internal\_flags | NULL |

**(2) “Internet VLAN” “Network Connections”**

For “Internet VLAN” “Network Connections” process the following Icore records into the GDB asset to site association and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**SITE**’, ROLE with FUNCTION\_TYPE ‘**PART\_OF**’ and FUNCTION\_ROLE NULL |
| NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id; | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id |
| NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.**site\_id**;  (=SITE.site\_id) | Id\_object\_to [via Enterprise Key Translation for SITE.site\_id] | Use SOURCE\_KEY.id which populated the respective SITE.id |
| NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id; | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id |
| - | Gdb\_internal\_flags | NULL |

<293772-CR160298>

**(3) “Tunnel Vlan” “Network Connections”**

For “Tunnel Vlan” “Network Connections” process the following usrp records into the GDB asset to site association and into the Data Index (ref. Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**SITE**’, ROLE with FUNCTION\_TYPE ‘**PART\_OF**’ and FUNCTION\_ROLE NULL |
| usrp.tunnel\_vlan\_vw.icore\_pvc\_id | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id |
| usrp.tunnel\_vlan\_vw.icore\_site\_id (same as icore.SITE.site\_id) | Id\_object\_to [via Enterprise Key Translation for SITE.site\_id] | Use SOURCE\_KEY.id which populated the respective SITE.id |
| usrp.tunnel\_vlan\_vw.icore\_pvc\_id | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id |
| - | Gdb\_internal\_flags | NULL |

</293772-CR160298>

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that the following steps have been executed:*

1. *HLD-254035-GCP-FLOW-ICORE-101 [Icore Data Flow into Site]*
2. *HLD-254035-GCP-FLOW-ICORE-104-A [Icore Data Flow into Asset PVC (Layer-3 “Network Connection”)]*
3. *HLD-254035-GCP-FLOW-ICORE-104-B [Icore Data Flow into Asset PVC (Layer-2 "Network Connection")]*
4. *HLD-254035-GCP-FLOW-ICORE-104-C [Icore Data Flow into Asset PVC (“Internet VLAN” “Network Connection”)]*
5. HLD-293772-CR160298-GCP-FLOW-USRP-104-D [Usrp Data Flow into Asset PVC (“Tunnel Vlan” “Network Connection”)]

#### HLD-254035-GCP-FLOW-ICORE-202-A [Icore Data Flow into Asset (“Access Circuit”) to Service]

Implement the following:

Process the following Icore records into the GDB asset to service association and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**SERVICE**’, ROLE with FUNCTION\_TYPE ‘**IMPLEMENTED\_BY**’ and FUNCTION\_ROLE NULL |
| CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id +  CUST\_ACCESS\_CKT.seq\_no | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| SERVICE.serv\_name (see “Database traversals” below) | Id\_object\_to [via lookup in GDB SERVICE and SERVICE\_TYPE\_NOTATION] | Use SERVICE.id which references SERVICE\_TYPE\_NOTATION having service\_name = Icore SERVICE.serv\_name.  Note: If the service.serv\_name = ‘OPT-E-WAN’ then look up for the service\_option.serv\_opt = ‘ASEoD’; if both matches, then set GDB.SERVICE.ID\_SERVICE\_TYPE referencing to ‘SDN-ETHERNET’ <**302503 - tkt 690443** > |
| CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id +  CUST\_ACCESS\_CKT.seq\_no | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SERVICE\_ASGMT.site\_id | CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |
| GDB.SERVICE\_TYPE\_NOTATION.service\_name | SERVICE.serv\_name |

**Hint**: For the above a very similar process already exists for INSTAR (HLD-232213e-GCP-FLOW-INSTAR-202 [Instar Data Flow into Asset to Service], implemented by the GRDB ETL team); that already existing process can be used for all needed details.

##### Additional step for assigning the “FR” service to an asset

For assigning the “FR” service to an asset a dedicated additional step is needed, as the “FR” service is not covered by the database traversal as given above.

The logic is given under “Logic for assigning the “FR” service to an asset”.

Use CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id as the “site id variable”.

<BEGIN 254035c.DEFECT-15114>

**NOTE**: This logic must only be applied, where no “service name” could be found using the logic from “Database traversals” as given above !

(This means that a “service name” gets associated to an “access circuit” asset either by “Database traversals” given above, or by the 2 special sections on assigning “FR” resp. “ATM”.)

<END 254035c.DEFECT-15114>

##### Additional step for assigning the “ATM” service to an asset

For assigning the “ATM” service to an asset a dedicated additional step is needed, as the “ATM” service is not covered by the database traversal as given above.

The logic is given under “Logic for assigning the “ATM” service to an asset”.

Use CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id as the “site id variable”.

<BEGIN 254035c.DEFECT-15114>

**NOTE**: This logic must only be applied, where no “service name” could be found using the logic from “Database traversals” as given above !

(This means that a “service name” gets associated to an “access circuit” asset either by “Database traversals” given above, or by the 2 specail sections on assigning “FR” resp. “ATM”.)

<END 254035c.DEFECT-15114>

*Note that in general an asset may be associated to more than one service, and the above two steps may just increase the number of services an asset may be associated to.*

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that the following steps have been executed:*

1. *HLD-254035-GCP-FLOW-ICORE-102-A [Icore Data Flow into Asset CUST\_ACCESS (“Access Circuit”)]*
2. *HLD-254035-GCP-FLOW-ICORE-102-B [Icore Data Flow into Asset CUST\_ACCESS\_CKT (“Access Circuit”)]*
3. *HLD-254035-GCP-FLOW-ICORE-106 [Icore Data Flow into Service]*

##### Logic for assigning the “FR” service to an asset

The logic for assigning the “FR” service to an asset is as follows:

* use the “site id variable” as given by the context
* execute the following statement:  
  (Note that “icore.network\_connection” is a view used to “abstract away” the “local” and “remote” data blocks from the PVC record)

select

count(\*)

from

icore.site s,

icore.pvc p,

icore.network\_connection n

where (1 = 1)

and (s.site\_id = <site id variable>

and (s.site\_id = n.site\_id)

and (n.pvc\_id = p.pvc\_id)

and (s.protocol = 'FR')

and (p.pvc\_type = 'CUST')

and (p.PVC\_STAT = 'IN-SERVICE')

;

* if that statement returns a value greater “0”, then assign the service “FR”; otherwise do nothing.

**Note** that the execution of the SQL statement is not really needed, it only demonstrates the logic for finding out whether or not to assign the service to an asset item.

##### Logic for assigning the “ATM” service to an asset

The logic for assigning the “ATM” service to an asset is as follows:

* use the “site id variable” as given by the context
* execute the following statement:  
  (Note that “icore.network\_connection” is a view used to “abstract away” the “local” and “remote” data blocks from the PVC record)

select

count(\*)

from

icore.site s,

icore.pvc p,

icore.network\_connection n

where (1 = 1)

and (s.site\_id = <site id variable>

and (s.site\_id = n.site\_id)

and (n.pvc\_id = p.pvc\_id)

and (s.protocol = 'ATM')

and (p.pvc\_type = 'CUST')

and (p.PVC\_STAT = 'IN-SERVICE')

;

* if that statement returns a value greater “0”, then assign the service “ATM”; otherwise do nothing.

**Note** that the execution of the SQL statement is not really needed, it only demonstrates the logic for finding out whether or not to assign the service to an asset item.

<BEGIN 254035c.DEFECT-15114>

#### HLD-254035.DEFECT-15114- GCP-FLOW-ICORE-202-A [Icore Data Flow into Asset (“Access Circuit”) to Service <STATIC DATA CLEANSING>]

This requirement is for cleansing “access-circuit-to-service” associations from “bad” service associations.

This is a one-time-only requirement that has to be used as an “addendum” to the logic fix for defect “15114”.

The execution of this process requires the maintenance of a change tracking event into GDB.CHANGE\_TRACKING, CHANGE\_SYSTEM and CHANGE\_USER. The change tracking event for the service delivery data flow must use CHANGE\_SYSTEM.name = ’ICORE\_TO\_GDB’ and CHANGE\_USER.name = ’ICORE\_TO\_GDB’. It is sufficient to have one change tracking event for all affected records during a single execution of this process.

For all “access circuit” assets as having been loaded from ICORE,  
all associations to service type values of “FR” and/or “ATM” need to be removed,  
if associations to service type values other than “FR” and/or “ATM” exist.

<END 254035c.DEFECT-15114>

#### HLD-254035-GCP-FLOW-ICORE-202-B [Icore Data Flow into Asset (“Customer Network”) to Service]

<BEGIN 254035c.DEFECT-12499>

**~~NOTE: This is a “pseudo-requirement” not to be included in any traceability files.~~**

~~NOTE: For assets of type “Customer Network” a concept of “service” does not exist, so no corresponding association will be created here.~~

**(1) “Layer-3” “Customer Networks”**

Implement the following:

For “Layer-3” “Customer Networks” process the following Icore records into the GDB asset to service association and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**SERVICE**’, ROLE with FUNCTION\_TYPE ‘**IMPLEMENTED\_BY**’ and FUNCTION\_ROLE NULL |
| VPN.vpn\_id | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| SERVICE.serv\_name (see “Database traversals” below) | Id\_object\_to [via lookup in GDB SERVICE and SERVICE\_TYPE\_NOTATION] | Use SERVICE.id which references SERVICE\_TYPE\_NOTATION having service\_name = Icore SERVICE.serv\_name.  Note: If the service.serv\_name = ‘OPT-E-WAN’ then look up for the service\_option.serv\_opt = ‘ASEoD’; if both matches, then set GDB.SERVICE.ID\_SERVICE\_TYPE referencing to ‘SDN-ETHERNET’ <**302503 - tkt 690443>** |
| VPN.vpn\_id | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| VPN.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |
| GDB.SERVICE\_TYPE\_NOTATION.service\_name | SERVICE.serv\_name |

**Hint**: For the above a very similar process already exists for INSTAR (HLD-232213e-GCP-FLOW-INSTAR-202 [Instar Data Flow into Asset to Service], implemented by the GRDB ETL team); that already existing process can be used for all needed details.

**(2) “Layer-2” “Customer Networks”**

For “Layer-2” “Customer Networks” process the following Icore records into the GDB asset to service association and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

*NOTE: Only include PVC records where for the given PVC.pvc\_id no record exists in IPFR for IPFR.pvc\_id* ***AND*** *where* ***both*** *the “PVC.pvc\_lcust\_id” and the “PVC.pvc\_rcust\_id” value point to a “real” customer* ***AND*** *where* ***both*** *“PORT\_ASGMT.port\_stat” (for “l” and “r”) have the value “IN-SERVICE”.*

<BEGIN 254035c.DEFECT-R3S1400015110>

The original design (which is the text below that is stroken through) sometimes creates associations to a service type (via the two “site\_id” links where the corr. “site/port” may carry more than the “layer-2” services) that is not appropriate here. So that logic shall be removed (identified by the text that is stroken through below).

For creating associations with a service the only logic to be applied remains this:

* Additional step for assigning the “FR” service to an asset
* Additional step for assigning the “ATM” service to an asset

*~~NOTE: As here~~* ***~~both~~*** *~~the “PVC.pvc\_lsite\_id” and the “PVC.pvc\_rsite\_id” values are used to create a “customer-network-to-service” association, it is possible to get 2 associations for a “layer-2” “customer network”, if the two “site\_id” values point to different service types ! (If they are pointing to the same service type, then only one association record shall be created.)~~*

|  |  |  |
| --- | --- | --- |
| **~~Icore table.column~~** | **~~GDB ASSOCIATION column~~** | **~~Processing comment~~** |
| ~~-~~ | ~~id\_change\_tracking~~ | ~~See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process]~~ |
| ~~-~~ | ~~is\_read\_only~~ | ~~‘Y’~~ |
| ~~-~~ | ~~Id\_role\_notation~~ | ~~NULL~~ |
| ~~-~~ | ~~Id\_association\_type~~ | ~~Referencing ASSOCIATION\_TYPE.id which represents:~~  ~~OBJECT\_TYPE\_WHAT ‘~~**~~ASSET~~**~~’, OBJECT\_TYPE\_TO ‘~~**~~SERVICE~~**~~’, ROLE with FUNCTION\_TYPE ‘~~**~~IMPLEMENTED\_BY~~**~~’ and FUNCTION\_ROLE NULL~~ |
| ~~PVC.pvc\_id~~ | ~~Id\_object\_what [via Enterprise Key Translation]~~ | ~~Use SOURCE\_KEY.id which populated the respective ASSET.id~~  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| ~~SERVICE.serv\_name (see “Database traversals” below)~~ | ~~Id\_object\_to [via lookup in GDB SERVICE and SERVICE\_TYPE\_NOTATION]~~ | ~~Use SERVICE.id which references SERVICE\_TYPE\_NOTATION having service\_name = Icore SERVICE.serv\_name.~~ |
| ~~PVC.pvc\_id~~ | ~~Id\_source\_key [via Enterprise Key Translation]~~ | ~~Use SOURCE\_KEY.id which populated the respective ASSET.id~~  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| ~~-~~ | ~~Gdb\_internal\_flags~~ | ~~NULL~~ |

~~Database traversals (PVC.pvc\_~~**~~l~~**~~site\_id (“local/left”)):~~

|  |  |
| --- | --- |
| **~~From table.column~~** | **~~To table.column~~** |
| ~~SERVICE\_ASGMT.site\_id~~ | ~~PVC.pvc\_~~**~~l~~**~~site\_id~~ |
| ~~SERVICE\_ASGMT.serv\_opt\_id~~ | ~~SERVICE\_OPTION.serv\_opt\_id~~ |
| ~~SERVICE\_OPTION.serv\_id~~ | ~~SERVICE.serv\_id~~ |
| ~~GDB.SERVICE\_TYPE\_NOTATION.service\_name~~ | ~~SERVICE.serv\_name~~ |

~~Database traversals (PVC.pvc\_~~**~~r~~**~~site\_id (“remote/right”)):~~

|  |  |
| --- | --- |
| **~~From table.column~~** | **~~To table.column~~** |
| ~~SERVICE\_ASGMT.site\_id~~ | ~~PVC.pvc\_~~**~~r~~**~~site\_id~~ |
| ~~SERVICE\_ASGMT.serv\_opt\_id~~ | ~~SERVICE\_OPTION.serv\_opt\_id~~ |
| ~~SERVICE\_OPTION.serv\_id~~ | ~~SERVICE.serv\_id~~ |
| ~~GDB.SERVICE\_TYPE\_NOTATION.service\_name~~ | ~~SERVICE.serv\_name~~ |

**~~NOTE:~~**

~~For a “layer-2” “customer network” a “customer-network-to-service” association may be created for both “PVC.pvc\_~~**~~l~~**~~site\_id” and “PVC.pvc\_~~**~~r~~**~~site\_id”; this should, however, happen only if the service values are different; if they are the same, then only a single association record shall be created !~~

**~~Hint~~**~~: For the above a very similar process already exists for INSTAR (HLD-232213e-GCP-FLOW-INSTAR-202 [Instar Data Flow into Asset to Service], implemented by the GRDB ETL team); that already existing process can be used for all needed details.~~

<END 254035c.DEFECT-R3S1400015110>

**(3) “Internet VLAN” “Customer Networks”**

For “Internet VLAN” “Customer Networks” process the following Icore records into the GDB asset to service association and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**SERVICE**’, ROLE with FUNCTION\_TYPE ‘**IMPLEMENTED\_BY**’ and FUNCTION\_ROLE NULL |
| CUSTOMER\_NETWORK\_INET\_VLAN.vpn\_id + CUSTOMER\_NETWORK\_INET\_VLAN.pvc\_id + CUSTOMER\_NETWORK\_INET\_VLAN.site\_id | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| SERVICE.serv\_name (see “Database traversals” below) | Id\_object\_to [via lookup in GDB SERVICE and SERVICE\_TYPE\_NOTATION] | Use SERVICE.id which references SERVICE\_TYPE\_NOTATION having service\_name = Icore SERVICE.serv\_name. |
| CUSTOMER\_NETWORK\_INET\_VLAN.vpn\_id + CUSTOMER\_NETWORK\_INET\_VLAN.pvc\_id + CUSTOMER\_NETWORK\_INET\_VLAN.site\_id | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SERVICE\_ASGMT.site\_id | CUSTOMER\_NETWORK\_INET\_VLAN.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |
| GDB.SERVICE\_TYPE\_NOTATION.service\_name | SERVICE.serv\_name |

**Hint**: For the above a very similar process already exists for INSTAR (HLD-232213e-GCP-FLOW-INSTAR-202 [Instar Data Flow into Asset to Service], implemented by the GRDB ETL team); that already existing process can be used for all needed details.

##### Additional step for assigning the “FR” service to an asset

For assigning the “FR” service to an asset a dedicated additional step is needed, as the “FR” service is not covered by the database traversal as given above.

The logic is given under “Logic for assigning the “FR” service to an asset”.

Use “PVC.pvc\_**l**site\_id” <or> PVC.pvc\_**r**site\_id <or> “CUSTOMER\_NETWORK\_INET\_VLAN.site\_id” as the “site id variable”. *(****NOTE*** *that this does not apply to “layer-3” “customer networks”, as for them no “site\_id” data exists !)*

<BEGIN 254035c.DEFECT-15114>

**NOTE**: This logic must only be applied to “layer-2” assets !

<END 254035c.DEFECT-15114>

##### Additional step for assigning the “ATM” service to an asset

For assigning the “ATM” service to an asset a dedicated additional step is needed, as the “ATM” service is not covered by the database traversal as given above.

The logic is given under “Logic for assigning the “ATM” service to an asset”.

Use “PVC.pvc\_**l**site\_id” <or> PVC.pvc\_**r**site\_id <or> “CUSTOMER\_NETWORK\_INET\_VLAN.site\_id” as the “site id variable”. *(****NOTE*** *that this does not apply to “layer-3” “customer networks”, as for them no “site\_id” data exists !)*

<BEGIN 254035c.DEFECT-15114>

**NOTE**: This logic must only be applied to “layer-2” assets !

<END 254035c.DEFECT-15114>

*Note that in general an asset may be associated to more than one service, and the above two steps may just increase the number of services an asset may be associated to.*

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that the following steps have been executed:*

1. *HLD-254035-GCP-FLOW-ICORE-103-A [Icore Data Flow into Asset VPN (Layer-3 “Customer Network”)]*
2. *HLD-254035-GCP-FLOW-ICORE-103-B [Icore Data Flow into Asset PVC (as VPN) (Layer-2 "Customer Network")]*
3. *HLD-254035-GCP-FLOW-ICORE-103-C [Icore Data Flow into Asset VPN (“Internet VLAN” “Customer Network”)]*
4. *HLD-254035-GCP-FLOW-ICORE-106 [Icore Data Flow into Service]*

<END 254035c.DEFECT-12499>

<BEGIN 254035c.DEFECT-15114>

#### HLD-254035.DEFECT-15114- GCP-FLOW-ICORE-202-B [Icore Data Flow into Asset (“Customer Network”) to Service <STATIC DATA CLEANSING>]

This requirement is for cleansing “customer-network-to-service” associations from “bad” service associations.

This is a one-time-only requirement that has to be used as an “addendum” to the logic fix for defect “15114”.

The execution of this process requires the maintenance of a change tracking event into GDB.CHANGE\_TRACKING, CHANGE\_SYSTEM and CHANGE\_USER. The change tracking event for the service delivery data flow must use CHANGE\_SYSTEM.name = ’ICORE\_TO\_GDB’ and CHANGE\_USER.name = ’ICORE\_TO\_GDB’. It is sufficient to have one change tracking event for all affected records during a single execution of this process.

**(1) “Layer-3” “Customer Networks”**

“layer-3” “customer network” assets are identified by having a GDB.ASSET\_EXT\_CUSTOMER\_NETWORK.id\_network\_type value that points to a value referencing NETWORK\_TYPE ‘**NT\_VPN**’.

For those assets all associations to service type values of “FR” and/or “ATM” need to be removed.

**(2) “Layer-2” “Customer Networks”**

“layer-2” “customer network” assets are identified by having a GDB.ASSET\_EXT\_CUSTOMER\_NETWORK.id\_network\_type value that points to a value referencing NETWORK\_TYPE ‘**NT\_PVC**’.

For those assets all associations to service type values *that are different from* “FR” and/or “ATM” need to be removed.

**(3) “Internet VLAN” “Customer Networks”**

“Internet VLAN” “customer network” assets are identified by having a GDB.ASSET\_EXT\_CUSTOMER\_NETWORK.id\_network\_type value that points to a value referencing NETWORK\_TYPE ‘**NT\_INTERNET**’.

For those assets all associations to service type values of “FR” and/or “ATM” need to be removed.

<END 254035c.DEFECT-15114>

<BEGIN 254035c.DEFECT-12499>

#### HLD-254035.DEFECT-12499- GCP-FLOW-ICORE-202-B [Icore Data Flow into Asset (“Customer Network”) to Service <STATIC INITIAL LOAD>]

This requirement is for initially creating “customer-network-to-service” associations.

This initial load needs to have been completed before the changes for DEFECT-12499 for the requirement   
“HLD-254035-GCP-FLOW-ICORE-202-B [Icore Data Flow into Asset (“Customer Network”) to Service]”   
can be executed for the first time.

The execution of this process requires the maintenance of a change tracking event into GDB.CHANGE\_TRACKING, CHANGE\_SYSTEM and CHANGE\_USER. The change tracking event for the service delivery data flow must use CHANGE\_SYSTEM.name = ’ICORE\_TO\_GDB’ and CHANGE\_USER.name = ’ICORE\_TO\_GDB’. It is sufficient to have one change tracking event for all affected records during a single execution of this process.

NOTE that this requirement is a “one-time” load.

The set of assets for which this initial load applies is as follows  
(all “customer networks” loaded from ICORE):

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| GDB.ASSET\_EXT\_CUSTOMER\_NETWORK.id\_asset | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id\_meta\_table | METADATA.META\_TABLE.id |
| METADATA.META\_SYSTEM.id | METADATA.META\_TABLE.id\_meta\_system |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| METADATA.META\_SYSTEM | NAME | ICORE |

For the initial load - for the set of assets as identified above - the logic is as follows:

* Use the same logic as is used under   
  “HLD-254035-GCP-FLOW-ICORE-202-B [Icore Data Flow into Asset (“Customer Network”) to Service]”   
  to initially create “customer-network-to-service” associations.

<END 254035c.DEFECT-12499>

#### HLD-254035-GCP-FLOW-ICORE-202-C [Icore Data Flow into Asset (“Network Connection”) to Service]

*NOTE: For the following we have to do with data based on a PVC record and either a PVC.pvc\_lsite\_id or a PVC.pvc\_rsite\_id. For the Enterprise Key Translation (EKT) this is “abstracted away” by making use of the “NETWORK\_CONNECTION” or the “NETWORK\_CONNECTION\_INET\_VLAN” view.*

<BEGIN 254035c.DEFECT-15114>

Because of “DEFECT-15114” the section “(1) “Layer-2” and “Layer-3” “Network Connections”” has been separated into two separate ones, one for “layer-3” and one for “layer-2”.

The logic for “layer-2” has changed, but the logic for “layer-3” and “Internet VLAN” have not changed (except that “FR” and “ATM” service associations must not be applied any longer; see the changes under the corr. sections).

<END 254035c.DEFECT-15114>

Implement the following:

**(1) ~~“Layer-2” and~~ “Layer-3” “Network Connections”**

For ~~“Layer-2” and~~ “Layer-3” “Network Connections” process the following Icore records into the GDB asset to service association and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**SERVICE**’, ROLE with FUNCTION\_TYPE ‘**IMPLEMENTED\_BY**’ and FUNCTION\_ROLE NULL |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| SERVICE.serv\_name (see “Database traversals” below) | Id\_object\_to [via lookup in GDB SERVICE and SERVICE\_TYPE\_NOTATION] | Use SERVICE.id which references SERVICE\_TYPE\_NOTATION having service\_name = Icore SERVICE.serv\_name.  Note: If the service.serv\_name = ‘OPT-E-WAN’ then look up for the service\_option.serv\_opt = ‘ASEoD’; if both matches, then set GDB.SERVICE.ID\_SERVICE\_TYPE referencing to ‘SDN-ETHERNET’ <**302503 - tkt 690443>** |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

<BEGIN 254035c.DEFECT-15114>

**(2) “Layer-2” “Network Connections”**

For “Layer-2” “Network Connections” process the following Icore records into the GDB asset to service association and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**SERVICE**’, ROLE with FUNCTION\_TYPE ‘**IMPLEMENTED\_BY**’ and FUNCTION\_ROLE NULL |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| Only applicable here:  “FR” and/or “ATM” (see “Additional step for assigning the “FR” service to an asset” and “Additional step for assigning the “ATM” service to an asset” below) | Id\_object\_to [via lookup in GDB SERVICE and SERVICE\_TYPE\_NOTATION] | Use SERVICE.id which references SERVICE\_TYPE\_NOTATION having service\_name = (“ATM” or “FR”). |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

<END 254035c.DEFECT-15114>

**(3) “Internet VLAN” “Network Connections”**

For “Internet VLAN” “Network Connections” process the following Icore records into the GDB asset to service association and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**SERVICE**’, ROLE with FUNCTION\_TYPE ‘**IMPLEMENTED\_BY**’ and FUNCTION\_ROLE NULL |
| NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id; | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| SERVICE.serv\_name (see “Database traversals” below) | Id\_object\_to [via lookup in GDB SERVICE and SERVICE\_TYPE\_NOTATION] | Use SERVICE.id which references SERVICE\_TYPE\_NOTATION having service\_name = Icore SERVICE.serv\_name. |
| NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id; | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SERVICE\_ASGMT.site\_id | NETWORK\_CONNECTION.site\_id <or> NETWORK\_CONNECTION\_INET\_VLAN.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |
| GDB.SERVICE\_TYPE\_NOTATION.service\_name | SERVICE.serv\_name |

**Hint**: For the above a very similar process already exists for INSTAR (HLD-232213e-GCP-FLOW-INSTAR-202 [Instar Data Flow into Asset to Service], implemented by the GRDB ETL team); that already existing process can be used for all needed details.

<293772-CR160298>

**(4) “Tunnel Vlan” “Network Connections”**

For “Tunnel Vlan” “Network Connections” process the following usrp records into the GDB asset to service association and into the Data Index (ref. Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**SERVICE**’, ROLE with FUNCTION\_TYPE ‘**IMPLEMENTED\_BY**’ and FUNCTION\_ROLE NULL |
| usrp.tunnel\_vlan\_vw.icore\_pvc\_id | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| - | Id\_object\_to [via lookup in GDB SERVICE and SERVICE\_TYPE\_NOTATION] | Use SERVICE.id which references SERVICE\_TYPE\_NOTATION having service\_name = ‘AVPN’ |
| usrp.tunnel\_vlan\_vw.pvc\_id | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id |
| - | Gdb\_internal\_flags | NULL |

</293772-CR160298>

##### Additional step for assigning the “FR” service to an asset

For assigning the “FR” service to an asset a dedicated additional step is needed, as the “FR” service is not covered by the database traversal as given above.

The logic is given under “Logic for assigning the “FR” service to an asset”.

<BEGIN 254035c.DEFECT-15114>

Use NETWORK\_CONNECTION.site\_id ~~<or> NETWORK\_CONNECTION\_INET\_VLAN.site\_id~~ as the “site id variable”.

**NOTE**: This logic must only be applied to “layer-2” assets !

<END 254035c.DEFECT-15114>

##### Additional step for assigning the “ATM” service to an asset

For assigning the “ATM” service to an asset a dedicated additional step is needed, as the “ATM” service is not covered by the database traversal as given above.

The logic is given under “Logic for assigning the “ATM” service to an asset”.

<BEGIN 254035c.DEFECT-15114>

Use NETWORK\_CONNECTION.site\_id ~~<or> NETWORK\_CONNECTION\_INET\_VLAN.site\_id~~ as the “site id variable”.

**NOTE**: This logic must only be applied to “layer-2” assets !

<END 254035c.DEFECT-15114>

*Note that in general an asset may be associated to more than one service, and the above two steps may just increase the number of services an asset may be associated to.*

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that the following steps have been executed:*

1. *HLD-254035-GCP-FLOW-ICORE-104-A [Icore Data Flow into Asset PVC (Layer-3 “Network Connection”)]*
2. *HLD-254035-GCP-FLOW-ICORE-104-B [Icore Data Flow into Asset PVC (Layer-2 "Network Connection")]*
3. *HLD-254035-GCP-FLOW-ICORE-104-C [Icore Data Flow into Asset PVC (“Internet VLAN” “Network Connection”)]*
4. *HLD-293772-CR160298-GCP-FLOW-USRP-104-D [Usrp Data Flow into Asset PVC (“Tunnel Vlan” “Network Connection”)]*
5. *HLD-254035-GCP-FLOW-ICORE-106 [Icore Data Flow into Service]*

<BEGIN 254035c.DEFECT-15114>

#### HLD-254035.DEFECT-15114- GCP-FLOW-ICORE-202-C [Icore Data Flow into Asset (“Network Connection”) to Service <STATIC DATA CLEANSING>]

This requirement is for cleansing “network-connection-to-service” associations from “bad” service associations.

This is a one-time-only requirement that has to be used as an “addendum” to the logic fix for defect “15114”.

The execution of this process requires the maintenance of a change tracking event into GDB.CHANGE\_TRACKING, CHANGE\_SYSTEM and CHANGE\_USER. The change tracking event for the service delivery data flow must use CHANGE\_SYSTEM.name = ’ICORE\_TO\_GDB’ and CHANGE\_USER.name = ’ICORE\_TO\_GDB’. It is sufficient to have one change tracking event for all affected records during a single execution of this process.

**(1) “Layer-3” “Networks Connections”**

“layer-3” “network connections” assets are identified by the logic given under “HLD-254035-GCP-FLOW-ICORE-104-A [Icore Data Flow into Asset PVC (Layer-3 “Network Connection”)]”.

For those assets all associations to service type values of “FR” and/or “ATM” need to be removed.

**(2) “Layer-2” “Networks Connections”**

“layer-2” “network connections” assets are identified by the logic given under “HLD-254035-GCP-FLOW-ICORE-104-B [Icore Data Flow into Asset PVC (Layer-2 "Network Connection")]”.

For those assets all associations to service type values *that are different from* “FR” and/or “ATM” need to be removed.

**(3) “Internet VLAN” “Networks Connections”**

“Internet VLAN” “network connections” assets are identified by the logic given under “HLD-254035-GCP-FLOW-ICORE-104-C [Icore Data Flow into Asset PVC (“Internet VLAN” “Network Connection”)]”.

For those assets all associations to service type values of “FR” and/or “ATM” need to be removed.

**(4) “Tunnel VLAN” “Networks Connections”**

“Tunnel VLAN” “network connections” assets are identified by the logic given under “HLD-293772-CR160298-GCP-FLOW-USRP-104-D [Usrp Data Flow into Asset PVC (“Tunnel Vlan” “Network Connection”)]”.

<END 254035c.DEFECT-15114>

#### HLD-254035-GCP-FLOW-ICORE-203 [Icore Data Flow into Asset (“Access Circuit Parent”) to Asset]

This requirement is for covering the relationship between access circuit assets where one access circuit is the “parent” access circuit of another one.

Implement the following:

Process the following Icore records into the GDB asset to asset (access circuit parent) association and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**ASSET**’, ROLE with FUNCTION\_TYPE ‘**ROLLS\_UP\_TO**’ and FUNCTION\_ROLE NULL |
| CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id +  CUST\_ACCESS\_CKT.seq\_no | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| CUST\_ACCESS.ia\_ckt = CUST\_ACCESS(parent).acc\_ckt  <or>  CUST\_ACCESS\_CKT.site\_id = CUST\_ACCESS.site\_id  and  CUST\_ACCESS.ia\_ckt = CUST\_ACCESS(parent).acc\_ckt | Id\_object\_to [via Enterprise Key Translation for CUST\_ACCESS(parent).site\_id] | Use SOURCE\_KEY.id which populated the respective ~~SITE.id~~ ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id +  CUST\_ACCESS\_CKT.seq\_no | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that the following steps have been executed:*

1. *HLD-254035-GCP-FLOW-ICORE-102-A [Icore Data Flow into Asset CUST\_ACCESS (“Access Circuit”)]*
2. *HLD-254035-GCP-FLOW-ICORE-102-B [Icore Data Flow into Asset CUST\_ACCESS\_CKT (“Access Circuit”)]*

#### HLD-254035-GCP-FLOW-ICORE-204 [Icore Data Flow into Phy Port (“Access Circuit Port”) to Asset]

This requirement is for covering the relationship between access circuit assets and physical ports associated with them.

Implement the following:

Process the following Icore records into the GDB physical port to asset association and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

**(A) CE Physical Port**

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**PHY\_PORT**’, OBJECT\_TYPE\_TO ‘**ASSET**’, ROLE with FUNCTION\_TYPE ‘**USED\_BY**’ and FUNCTION\_ROLE NULL |
| SITE.site\_id | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective PHY\_PORT.id  (where PHY\_PORT.id\_port\_type references “**CE**”)  <BEGIN DEFECT-FIX-34627.2015-02-16> ~~<BEGIN DEFECT-FIX-34627.2015-02-12> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “port unifications” where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END DEFECT-FIX-34627.2015-02-12>~~ <END DEFECT-FIX-34627.2015-02-16> |
| CUST\_ACCESS.site\_id  (=SITE.site\_id)  <or>  CUST\_ACCESS\_CKT.site\_id + CUST\_ACCESS\_CKT.seq\_no  (=SITE.site\_id) | Id\_object\_to [via Enterprise Key Translation for CUST\_ACCESS.site\_id] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN DEFECT-FIX-34627.2015-02-16> ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ <END DEFECT-FIX-34627.2015-02-16> |
| SITE.site\_id | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective PHY\_PORT.id |
| - | Gdb\_internal\_flags | NULL |

**(B) PE Physical Port**

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**PHY\_PORT**’, OBJECT\_TYPE\_TO ‘**ASSET**’, ROLE with FUNCTION\_TYPE ‘**USED\_BY**’ and FUNCTION\_ROLE NULL |
| PORT\_ASGMT.cust\_port | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective PHY\_PORT.id  (where PHY\_PORT.id\_port\_type references “**PE**”)  <BEGIN DEFECT-FIX-34627.2015-02-16> ~~<BEGIN DEFECT-FIX-34627.2015-02-12> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “port unifications” where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END DEFECT-FIX-34627.2015-02-12>~~ <END DEFECT-FIX-34627.2015-02-16> |
| CUST\_ACCESS.site\_id  (=GDB.PHY\_PORT.  icore\_ce\_port\_icore\_site\_id)  <or>  CUST\_ACCESS\_CKT.site\_id + CUST\_ACCESS\_CKT.seq\_no  (=GDB.PHY\_PORT.  icore\_ce\_port\_icore\_site\_id) | Id\_object\_to [via Enterprise Key Translation for CUST\_ACCESS.site\_id] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN DEFECT-FIX-34627.2015-02-16> ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ <END DEFECT-FIX-34627.2015-02-16> |
| PORT\_ASGMT.cust\_port | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective PORT.id |
| - | Gdb\_internal\_flags | NULL |

Database Traversal (1) [for “PE” only]:

Use the record from “GDB.PHY\_PORT” where the “ID” value corresponds to “PORT\_ASGMT.cust\_port” via the EKT and where “GDB.PHY\_PORT.id\_port\_type” references “PE”.

<BEGIN: Port Load Corrections>

Database traversals (2.1) [CUST\_ACCESS]:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SITE.site\_id | CUST\_ACCESS.site\_id |
| SITE.site\_id | PORT\_ASGMT.site\_id |
| PORT\_ASGMT.equip\_id | EQUIPMENT.equip\_id |

Database traversals (2.2) [CUST\_ACCESS\_CKT]:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SITE.site\_id | CUST\_ACCESS\_CKT.site\_id |
| SITE.site\_id | PORT\_ASGMT.site\_id |
| PORT\_ASGMT.equip\_id | EQUIPMENT.equip\_id |

Database constraints (4):

*(generic)*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |

<END: Port Load Corrections>

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that the following steps have been executed:*

1. *HLD-254035-GCP-FLOW-ICORE-102-A [Icore Data Flow into Asset CUST\_ACCESS (“Access Circuit”)]*
2. *HLD-254035-GCP-FLOW-ICORE-102-B [Icore Data Flow into Asset CUST\_ACCESS\_CKT (“Access Circuit”)]*
3. *HLD-254035-GCP-FLOW-ICORE-120-A [Icore Data Flow into Phy Port SITE (Layer-3 “Phy Port”)]*
4. *HLD-254035-GCP-FLOW-ICORE-120-B [Icore Data Flow into Phy Port SITE (Layer-2 Phy Port”)]*

#### HLD-CR-156194-Defect-69175-GCP-FLOW-ICORE-204-A [Icore Data Flow into Asset (“Access Circuit”) to Asset (CSU/DSU)]

<CR-156194-Defect-69175>

This requirement is for covering the relationship between access circuit assets and related CSU/DSU Equipment associated with them. The Equipment should already have been loaded into GDB.

|  |  |  |
| --- | --- | --- |
| **Icore/NC3 table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**ASSET**’, ROLE with FUNCTION\_TYPE ‘IS\_RELATED\_CSU\_DSU’ and FUNCTION\_ROLE NULL |
| CUST\_ACCESS.site\_id  (for acc\_ckt = nc3.ckt.common\_name)  <or>  CUST\_ACCESS\_CKT.site\_id + CUST\_ACCESS\_CKT.seq\_no  (for circuit\_id = nc3.ckt.common\_name)  See [Access Circuit Matching logic](#_Access_Circuit_Matching) for comparing the access circuits above. Use similar logic to compare. | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id |
| NC3.EQP.ODBID | Id\_object\_to [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id |
| CUST\_ACCESS.site\_id  <or>  CUST\_ACCESS\_CKT.site\_id + CUST\_ACCESS\_CKT.seq\_no | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id |
| - | Gdb\_internal\_flags | NULL |

##### Logic used to create the association

<CR-156194-Defect-69175>

The below SQL shows the relationships to be traveresed to retrieve the Access circuit id from the NC3 tables for comparison in the above. The following should be run for NC3 equipment that is already loaded into GDB that are not routers. This is shown as an example to show the relationships and may not necessarily be the most efficient query:

select ckt.common\_name as circuit\_id, parm.value as icore\_site\_id, eqp.odbid,

eqp\_ref.eqp\_category, eqp.service\_name, eqp.common\_name as hostname

from nc3\_mat.eqp, nc3\_mat.eqp\_ref, nc3\_mat.eqp card, nc3\_mat.port,

nc3\_mat.cmp\_usage cu, nc3\_mat.cmp\_usage cu2, nc3\_mat.config,

nc3\_mat.cmp\_usage cu3, nc3\_mat.cmp\_usage cu4, nc3\_mat.phy\_circuit ckt, nc3\_mat.parm

where eqp.odbid = card.odbid\_main\_eqp and eqp.stop\_date > sysdate

and eqp.status = 'Useable'

and eqp\_ref.odbid = eqp.odbid\_eqp\_ref

and card.odbid = port.odbid\_eqp and card.stop\_date > sysdate

and card.status = 'Useable'

and port.odbid = cu.odbid\_cmponent and port.stop\_date > sysdate

and cu.odbid\_config = cu2.odbid\_cmponent and cu.status <> 'Stopped'

and cu2.odbid\_config = config.odbid and cu2.status <> 'Stopped'

and config.odbid = cu3.odbid\_config and cu3.component\_type = 'Configuration'

and config.status not in ('Cancelled', 'Disco')

and cu3.odbid\_cmponent = cu4.odbid\_config

and cu4.component\_type = 'PhysicalCktSgmt' and cu3.status <> 'Stopped'

and cu4.odbid\_cmponent = ckt.odbid and cu4.status <> 'Stopped'

and ckt.status = 'Useable'

and ckt.odbid = parm.odbid\_cmponent (+) and parm.internal\_label = 'ICORE\_SITE\_ID'

and parm.status <> 'Stopped'

and eqp\_ref.eqp\_category in ('Probe', 'CSU', 'DSU', 'CSU/DSU')

and eqp.service\_name = 'NB-IPVPN'

#### HLD-CR-156194-Defect-69175-GCP-FLOW-ICORE-204-B [Icore One-time Data Flow into Asset (“Access Circuit”) to Asset (CSU/DSU)]

<CR-156194-Defect-69175>

This is the one-time load requirement for building the Access Circuit to CSU/DSU relationship for existing access circuits. The logic is the same as described in [HLD-CR-156194-Defect-69175-GCP-FLOW-ICORE-204-A [Icore Data Flow into Asset (“Access Circuit”) to Asset (CSU/DSU)]](#_HLD-CR-156194-Defect-69175-GCP-FLOW)

#### HLD-254035-GCP-FLOW-ICORE-205 [Icore Data Flow into Asset (“Network Connection Access Circuits”) to Asset]

This requirement is for covering the relationship between access circuit assets that are used by a network connection.

*NOTE: For the following we have to do with data based on a PVC record and either a PVC.pvc\_lsite\_id or a PVC.pvc\_rsite\_id. For the Enterprise Key Translation (EKT) this is “abstracted away” by making use of the “NETWORK\_CONNECTION” and “NETWORK\_CONNECTION\_INET\_VLAN” view.*

Implement the following:

**(1) “Layer-2” and “Layer-3” “Network Connections”**

For “Layer-2” and “Layer-3” “Network Connections” process the following Icore records into the GDB asset to asset (access circuit(s) for network connection) association and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**ASSET**’, ROLE with FUNCTION\_TYPE ‘**USED\_BY**’ and FUNCTION\_ROLE NULL |
| CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id + CUST\_ACCESS\_CKT.seq\_no | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id;  (=CUST\_ACCESS.site\_id  <or>  CUST\_ACCESS\_CKT.site\_id) | Id\_object\_to [via Enterprise Key Translation for NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id + CUST\_ACCESS\_CKT.seq\_no | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

**(2) “Internet VLAN” “Network Connections”**

For “Internet VLAN” “Network Connections” process the following Icore records into the GDB asset to asset (access circuit(s) for network connection) association and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**ASSET**’, ROLE with FUNCTION\_TYPE ‘**USED\_BY**’ and FUNCTION\_ROLE NULL |
| CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id + CUST\_ACCESS\_CKT.seq\_no | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id;  (=CUST\_ACCESS.site\_id  <or>  CUST\_ACCESS\_CKT.site\_id) | Id\_object\_to [via Enterprise Key Translation for NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id + CUST\_ACCESS\_CKT.seq\_no | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

<293772-CR160298>

**(3) “Tunnel Vlan” “Network Connections”**

For “Tunnel Vlan” “Network Connections” process the following Icore records into the GDB asset to asset (access circuit(s) for network connection) association and into the Data Index (ref. Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **USRP table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**ASSET**’, ROLE with FUNCTION\_TYPE ‘**USED\_BY**’ and FUNCTION\_ROLE NULL |
| CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id + CUST\_ACCESS\_CKT.seq\_no  (For usrp.tunnel\_vlan\_vw.icore\_site\_id) | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id |
| usrp.tunnel\_vlan\_vw.icore\_pvc\_id | Id\_object\_to [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id |
| CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id + CUST\_ACCESS\_CKT.seq\_no | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id |
| - | Gdb\_internal\_flags | NULL |

</293772-CR160298>

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that the following steps have been executed:*

1. *HLD-254035-GCP-FLOW-ICORE-102-A [Icore Data Flow into Asset CUST\_ACCESS (“Access Circuit”)]*
2. *HLD-254035-GCP-FLOW-ICORE-102-B [Icore Data Flow into Asset CUST\_ACCESS\_CKT (“Access Circuit”)]*
3. *HLD-254035-GCP-FLOW-ICORE-104-A [Icore Data Flow into Asset PVC (Layer-3 “Network Connection”)]*
4. *HLD-254035-GCP-FLOW-ICORE-104-B [Icore Data Flow into Asset PVC (Layer-2 "Network Connection")]*
5. *HLD-254035-GCP-FLOW-ICORE-104-C [Icore Data Flow into Asset PVC (“Internet VLAN” “Network Connection”)]*
6. HLD-293772-CR160298-GCP-FLOW-USRP-104-D [Usrp Data Flow into Asset PVC (“Tunnel Vlan” “Network Connection”)]

#### HLD-254035-GCP-FLOW-ICORE-206-A [Icore Data Flow into Asset (“Customer Network for Network Connections”(Layer-3)) to Asset]

This requirement is for covering the relationship between a layer-3 customer network asset and its associated layer-3 network connections.

*NOTE: For the following we have to do with data based on a PVC record and either a PVC.pvc\_lsite\_id or a PVC.pvc\_rsite\_id. For the Enterprise Key Translation (EKT) this is “abstracted away” by making use of the “NETWORK\_CONNECTION” view.*

Implement the following:

Process the following Icore records into the GDB asset to asset (customer network for network connections) association and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**ASSET**’, ROLE with FUNCTION\_TYPE ‘**PART\_OF**’ and FUNCTION\_ROLE NULL |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| NETWORK\_CONNECTION.pvc\_id =  PVC.pvc\_id = IPFR.pvc\_id  <and>  IPFR.vpn\_id = VPN.vpn\_id  (= VPN.vpn\_id) | Id\_object\_to [via Enterprise Key Translation for VPN.vpn\_id] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

<279006><REL1602>

Update asset\_ext\_network\_connection.id\_network\_type (id\_object\_what) to value from asset\_ext\_customer\_network.id\_network\_type (id\_object\_to)

</279006>

<BEGIN 258491>

The following “Database traversals” and “Database constraints” are being introduced to make sure only “SVL” (i.e. non-MVL) PVCs are associated here. For MVL PVCs see the corresponding (new) requirement below.

Database traversals (1: ICORE only):

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| **ICORE**.PVC\_EXTENSION.pvc\_id (use “outer join”, as not all PVC/network connection records may have a PVC\_EXTENSION record !) | **ICORE**.NETWORK\_CONNECTION.pvc\_id |

Database constraints (1):

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| **ICORE**.PVC\_EXTENSION | MVL | (Is null) or (!= ‘Y’) or [no PVC\_EXTENSION record found] |

<END 258491>

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that the following steps have been executed:*

1. HLD-254035-GCP-FLOW-ICORE-103-A [Icore Data Flow into Asset VPN (Layer-3 “Customer Network”)]
2. HLD-254035-GCP-FLOW-ICORE-104-A [Icore Data Flow into Asset PVC (Layer-3 “Network Connection”)]

<BEGIN 258491>

#### HLD-258491-GCP-FLOW-ICORE-206-A [Icore Data Flow into Asset (“Customer Network for Network Connections”(Layer-3, MVL)) to Asset]

This requirement is for covering the relationship between a layer-3 customer network asset and its associated layer-3 MVL network connections.

*NOTE: For the following we have to do with data based on a PVC record and either a PVC.pvc\_lsite\_id or a PVC.pvc\_rsite\_id. For the Enterprise Key Translation (EKT) this is “abstracted away” by making use of the “NETWORK\_CONNECTION” view.*

***NOTE*** *that here for 1 PVC ID usually more than 1 VPN IDs are returned, so a PVC ID (“network connection”) gets associated with many VPNs (“customer networks”).*

Implement the following:

Process the following Icore records into the GDB asset to asset (customer network for network connections) association and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**ASSET**’, ROLE with FUNCTION\_TYPE ‘**PART\_OF**’ and FUNCTION\_ROLE NULL |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| NETWORK\_CONNECTION.pvc\_id =  PVC.pvc\_id = IPFR.pvc\_id  <and>  IPFR.vpn\_id = VPN.vpn\_id  (= VPN.vpn\_id) | Id\_object\_to [via Enterprise Key Translation for VPN.vpn\_id] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

<279006><REL1602>

Update asset\_ext\_network\_connection.id\_network\_type (id\_object\_what) to value from asset\_ext\_customer\_network.id\_network\_type (id\_object\_to)

</279006>

The following “Database traversals” and “Database constraints” are being introduced to make sure only “MVL” PVCs are associated here. For non-MVL (i.e. “SVL”) PVCs see the corresponding (old, but updated) requirement above.

Database traversals (1: ICORE only):

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| **ICORE**.PVC\_EXTENSION.pvc\_id | **ICORE**.NETWORK\_CONNECTION.pvc\_id |

Database constraints (1):

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| **ICORE**.PVC\_EXTENSION | MVL | ‘Y’ |

Database traversals (2: ICORE + INSTAR involved):

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| **INSTAR**.IP\_SERV\_ACC\_PT.icore\_pvc\_id | **ICORE**.NETWORK\_CONNECTION.pvc\_id |
| **INSTAR**.IP\_SERV\_ACC\_PT.serv\_acc\_pt\_id | **INSTAR**.IP\_PORT\_ASGMT.serv\_acc\_pt\_id |
| **INSTAR**.ENDPT2VPN.endpoint\_id | **INSTAR**.IP\_PORT\_ASGMT.ip\_endpt |
| **INSTAR.**ENDPT2VPN.vpn\_id | **ICORE**.VPN.vpn\_id |

<BEGIN MVL-VPN-FIX.2015-06-26 Defect 52461>

Database constraints (2: ICORE + INSTAR involved):

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| **INSTAR.**ENDPT2VPN.vpn\_id | Status | ‘ACTIVE’ |

<END MVL-VPN-FIX.2015-06-26 Defect 52461>

**NOTE** that here for 1 PVC ID usually more than 1 VPN IDs are returned, so a PVC ID (“network connection”) gets associated with many VPNs (“customer networks”).

The following embedded SQL query file contains an example of the logic needed for identifying the data to be used from INSTAR:



**NOTE** that this embedded SQL file may not be used “as is” for deployment. However, it gives the details of the logic that is needed.

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that the following steps have been executed:*

1. HLD-254035-GCP-FLOW-ICORE-103-A [Icore Data Flow into Asset VPN (Layer-3 “Customer Network”)]
2. HLD-254035-GCP-FLOW-ICORE-104-A [Icore Data Flow into Asset PVC (Layer-3 “Network Connection”)]

<END 258491>

#### HLD-254035-GCP-FLOW-ICORE-206-B [Icore Data Flow into Asset (“Customer Network for Network Connections”(Layer-2)) to Asset]

This requirement is for covering the relationship between a layer-2 customer network asset and its associated layer-2 network connections.

*NOTE: For the following we have to do with data based on a PVC record and either a PVC.pvc\_lsite\_id or a PVC.pvc\_rsite\_id. For the Enterprise Key Translation (EKT) this is “abstracted away” by making use of the “NETWORK\_CONNECTION” view.*

Implement the following:

Process the following Icore records into the GDB asset to asset (customer network for network connections) association and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**ASSET**’, ROLE with FUNCTION\_TYPE ‘**PART\_OF**’ and FUNCTION\_ROLE NULL |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| NETWORK\_CONNECTION.pvc\_id =  PVC.pvc\_id [network connection]  (=PVC.pvc\_id [customer network]) | Id\_object\_to [via Enterprise Key Translation for PVC.pvc\_id [customer network]] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

<279006><REL1602>

Update asset\_ext\_network\_connection.id\_network\_type (id\_object\_what) to value from asset\_ext\_customer\_network.id\_network\_type (id\_object\_to)

</279006>

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that the following steps have been executed:*

1. *HLD-254035-GCP-FLOW-ICORE-103-B [Icore Data Flow into Asset PVC (as VPN) (Layer-2 "Customer Network")]*
2. *HLD-254035-GCP-FLOW-ICORE-104-B [Icore Data Flow into Asset PVC (Layer-2 "Network Connection")]*

#### HLD-254035-GCP-FLOW-ICORE-206-C [Icore Data Flow into Asset (“Customer Network for Network Connections”(Internet VLAN)) to Asset]

This requirement is for covering the relationship between a Internet VLAN customer network asset and its associated Internet VLAN network connections.

*NOTE: For the following we have to do with data based on a PVC record and either a PVC.pvc\_lsite\_id or a PVC.pvc\_rsite\_id. For the Enterprise Key Translation (EKT) this is “abstracted away” by making use of the “NETWORK\_CONNECTION\_INET\_VLAN” view.*

Implement the following:

Process the following Icore records into the GDB asset to asset (customer network for network connections) association and into the Data Index (ref.

Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**ASSET**’, ROLE with FUNCTION\_TYPE ‘**PART\_OF**’ and FUNCTION\_ROLE NULL |
| NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| CUSTOMER\_NETWORK\_INET\_VLAN.vpn\_id + CUSTOMER\_NETWORK\_INET\_VLAN.pvc\_id + CUSTOMER\_NETWORK\_INET\_VLAN.site\_id   (=  *NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id* ) | Id\_object\_to [via Enterprise Key Translation for CUSTOMER\_NETWORK\_INET\_VLAN.vpn\_id + CUSTOMER\_NETWORK\_INET\_VLAN.pvc\_id + CUSTOMER\_NETWORK\_INET\_VLAN.site\_id] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

<279006><REL1602>

Update asset\_ext\_network\_connection.id\_network\_type (id\_object\_what) to value from asset\_ext\_customer\_network.id\_network\_type (id\_object\_to)

</279006>

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that the following steps have been executed:*

1. *HLD-254035-GCP-FLOW-ICORE-103-C [Icore Data Flow into Asset VPN (“Internet VLAN” “Customer Network”)]*
2. *HLD-254035-GCP-FLOW-ICORE-104-C [Icore Data Flow into Asset PVC (“Internet VLAN” “Network Connection”)]*

#### HLD-254035-GCP-FLOW-ICORE-207 [Icore Data Flow into Log Port (“Network Connection Port”) to Asset]

This requirement is for covering the relationship between network connection assets and logical ports associated with them.

Implement the following:

Process the following Icore records into the GDB logical port to asset association and into the Data Index (ref. Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

**(A) CE Logical Port**

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**LOG\_PORT**’, OBJECT\_TYPE\_TO ‘**ASSET**’, ROLE with FUNCTION\_TYPE ‘**USED\_BY**’ and FUNCTION\_ROLE NULL |
| <BEGIN: Port Load Corrections> |  |  |
| **(1) “Layer-3” and “Layer-2” Logical Ports:**  <LOG-PORT-EKT-FIX> LOG\_PORT\_CE\_VW.pvc\_id + LOG\_PORT\_CE\_VW.site\_id;  **(2) “Internet VLAN” Logical Ports:**  <LOG-PORT-EKT-FIX> LOG\_PORT\_CE\_INET\_VLAN\_VW.vpn\_id + LOG\_PORT\_CE\_INET\_VLAN\_VW.pvc\_id + LOG\_PORT\_CE\_INET\_VLAN\_VW.site\_id;  **(3) “Tunnel VLAN” Logical Ports:**  usrp.logical\_port.log\_port\_id  <293772-CR160298> | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective LOG\_PORT.id  (where LOG\_PORT.id\_port\_type references “**CE**”)  <BEGIN DEFECT-FIX-34627.2015-02-16> ~~<BEGIN DEFECT-FIX-34627.2015-02-12> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “port unifications” where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END DEFECT-FIX-34627.2015-02-12>~~ <END DEFECT-FIX-34627.2015-02-16> |
| <END: Port Load Corrections> |  |  |
| **(1) “Layer-3” and “Layer-2” Network Connections:**  NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id;  (={same as log port; see above})  **(2) “Internet VLAN” Network Connections:**  NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id;  (={same as log port; see above})  **(3) “Tunnel VLAN” Network Connections:**  usrp.tunnel\_vlan\_vw.icore\_pvc\_id  <293772-CR160298> | Id\_object\_to [via Enterprise Key Translation for  NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id  ***<resp.>***  NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id]  usrp.pvc.pvc\_id | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN DEFECT-FIX-34627.2015-02-16> ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ <END DEFECT-FIX-34627.2015-02-16> |
| <BEGIN: Port Load Corrections> |  |  |
| **(1) “Layer-3” and “Layer-2” Logical Ports:**  <LOG-PORT-EKT-FIX> LOG\_PORT\_CE\_VW.pvc\_id + LOG\_PORT\_CE\_VW.site\_id;  **(2) “Internet VLAN” Logical Ports:**  <LOG-PORT-EKT-FIX> LOG\_PORT\_CE\_INET\_VLAN\_VW.vpn\_id + LOG\_PORT\_CE\_INET\_VLAN\_VW.pvc\_id + LOG\_PORT\_CE\_INET\_VLAN\_VW.site\_id;  **(3) “Tunnel VLAN” Logical Ports:**  usrp.logical\_port.log\_port\_id  <293772-CR160298> | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective LOG\_PORT.id |
| <END: Port Load Corrections> |  |  |
| - | Gdb\_internal\_flags | NULL |

**(B) PE Logical Port**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | | | **Processing comment** |
| - | id\_change\_tracking | | | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | | | ‘Y’ |
| - | Id\_role\_notation | | | NULL |
| - | Id\_association\_type | | | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**LOG\_PORT**’, OBJECT\_TYPE\_TO ‘**ASSET**’, ROLE with FUNCTION\_TYPE ‘**USED\_BY**’ and FUNCTION\_ROLE NULL |
| <BEGIN: Port Load Corrections> | |  |  | |
| LOG\_PORT\_PE\_VW.pvc\_id + LOG\_PORT\_PE\_VW.site\_id + LOG\_PORT\_PE\_VW.cust\_port;  Tunnel Vlan:  usrp.TUNNEL\_VLAN\_PE\_PORT\_VW.log\_port\_id  <293772-CR160298> | Id\_object\_what [via Enterprise Key Translation] | | | Use SOURCE\_KEY.id which populated the respective LOG\_PORT.id  (where LOG\_PORT.id\_port\_type references “**PE**”)  <BEGIN DEFECT-FIX-34627.2015-02-16> ~~<BEGIN DEFECT-FIX-34627.2015-02-12> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “port unifications” where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END DEFECT-FIX-34627.2015-02-12>~~ <END DEFECT-FIX-34627.2015-02-16> |
| **(1) “Layer-3” and “Layer-2” Network Connections:**  NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id;  **(2) “Internet VLAN” Network Connections:**  NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id;  **(3) “Tunnel VLAN” Network Connections:**  usrp.tunnel\_vlan\_vw.icore\_pvc\_id  <293772-CR160298> | Id\_object\_to [via Enterprise Key Translation for  NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id  ***<resp.>***  NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id] | | | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN DEFECT-FIX-34627.2015-02-16> ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ <END DEFECT-FIX-34627.2015-02-16> |
| LOG\_PORT\_PE\_VW.pvc\_id + LOG\_PORT\_PE\_VW.site\_id + LOG\_PORT\_PE\_VW.cust\_port;  Tunnel Vlan:  usrp.TUNNEL\_VLAN\_PE\_PORT\_VW.log\_port\_id  <293772-CR160298> | Id\_source\_key [via Enterprise Key Translation] | | | Use SOURCE\_KEY.id which populated the respective LOG\_PORT.id |
| <END: Port Load Corrections> | |  |  | |
| - | Gdb\_internal\_flags | | | NULL |

Database Traversal (1) [“PE” ports only]:

Use the record from “GDB.LOG\_PORT” where the “ID” value corresponds to “(LOG\_PORT\_PE\_VW.pvc\_id + LOG\_PORT\_PE\_VW.site\_id + LOG\_PORT\_PE\_VW.cust\_port)” via the EKT and where “GDB.LOG\_PORT.id\_port\_type” references “PE”.

**(A) “Layer-3” logical PE ports**

The next two “Database Traversal/Constraint” blocks are for identifying the “PE port” side of the PVC record for **“Layer-3” logical PE ports**.

<BEGIN: Port Load Corrections>

Database traversals (Generic):

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| LOG\_PORT\_PE\_VW.pvc\_id | NETWORK\_CONNECTION.pvc\_id |
| LOG\_PORT\_PE\_VW.site\_id | NETWORK\_CONNECTION.site\_id |
| LOG\_PORT\_PE\_VW.cust\_port | PORT\_ASGMT.cust\_port |

Database traversals (2):

*(PVC PE port “r” side [“PE port” on the “r” side])*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| PVC.pvc\_id | NETWORK\_CONNECTION.pvc\_id |
| PVC.pvc\_id | IPFR.pvc\_id |
| PVC.pvc\_rcust\_port | PORT\_ASGMT.cust\_port |

Database constraints (2):

*(PVC PE port “r” side [“PE port” on the “r” side])*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| NETWORK\_CONNECTION | side | ‘remote’ |

Database traversals (3):

*(PVC PE port “l” side [“PE port” on the “l” side])*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| PVC.pvc\_id | NETWORK\_CONNECTION.pvc\_id |
| PVC.pvc\_id | IPFR.pvc\_id |
| PVC.pvc\_lcust\_port | PORT\_ASGMT.cust\_port |

Database constraints (3):

*(PVC PE port “l” side [“PE port” on the “l” side])*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| NETWORK\_CONNECTION | side | ‘local’ |

<END: Port Load Corrections>

<see also logic under:   
Logic for identifying a “customer” side in the ICORE.PVC table>

**(B) “Layer-2” logical PE ports**

The next two “Database Traversal/Constraint” blocks are for identifying the “PE port” side of the PVC record for **“Layer-2” logical PE ports**.

Database traversals:

*(PVC “l” side [“site” record located on the “l” PVC side])*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SITE.site\_id | PVC.pvc\_lsite\_id |

*(PVC “r” side [“site” record located on the “r” PVC side])*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SITE.site\_id | PVC.pvc\_rsite\_id |

<BEGIN: Port Load Corrections>

*(generic, i.e. independently of where “site” record located in the PVC record)*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| LOG\_PORT\_PE\_VW.pvc\_id | NETWORK\_CONNECTION.pvc\_id |
| LOG\_PORT\_PE\_VW.site\_id | NETWORK\_CONNECTION.site\_id |
| LOG\_PORT\_PE\_VW.cust\_port | PORT\_ASGMT.cust\_port |
| NETWORK\_CONNECTION.pvc\_id | PVC.pvc\_id |
| SITE.site\_id | PORT\_ASGMT.site\_id |
| PORT\_ASGMT.equip\_id | EQUIPMENT.equip\_id |
| PVC.pvc\_id | Not exists(IPFR.pvc\_id) |

<END: Port Load Corrections>

Database constraints (both sets must hold):

*(PVC “l” side)*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| PVC | PVC\_**L**CUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) |
| PVC | PVC\_**R**CUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) |
| PORT\_ASGMT *(via “PVC.pvc\_lsite\_id”)* | PORT\_STAT | IN-SERVICE |

<see also logic under:   
Logic for identifying a “customer” side in the ICORE.PVC table>

**(C) “Internet VLAN” logical PE ports**

The next two “Database Traversal/Constraint” blocks are for identifying the “PE port” side of the PVC record for **“Internet VLAN” logical PE ports**.

Database traversals:

<BEGIN: Port Load Corrections>

*(generic, i.e for both sides)*

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| LOG\_PORT\_PE\_VW.pvc\_id | NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id |
| LOG\_PORT\_PE\_VW.site\_id | NETWORK\_CONNECTION\_INET\_VLAN.site\_id |
| LOG\_PORT\_PE\_VW.cust\_port | PORT\_ASGMT.cust\_port |
| PVC.pvc\_id | NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id |
| SITE.site\_id | NETWORK\_CONNECTION\_INET\_VLAN.site\_id |
| SITE.site\_id | PORT\_ASGMT.site\_id |
| PORT\_ASGMT.equip\_id | EQUIPMENT.equip\_id |

<END: Port Load Corrections>

Database constraints:

*(PVC “l” side [“customer/site” on the “l” side])*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| NETWORK\_CONNECTION\_INET\_VLAN | SIDE | ‘local’ |
| NETWORK\_CONNECTION\_INET\_VLAN | CUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) |
| PVC | PVC\_**R**CUST\_ID | = 395 (i.e. is “AT&T” customer id) |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |

Database constraints:

*(PVC “r” side [“customer/site” on the “r” side])*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| NETWORK\_CONNECTION\_INET\_VLAN | SIDE | ‘remote’ |
| NETWORK\_CONNECTION\_INET\_VLAN | CUST\_ID | != 395 (i.e. is “non-AT&T/real customer” customer id) |
| PVC | PVC\_**L**CUST\_ID | = 395 (i.e. is “AT&T” customer id) |
| PORT\_ASGMT | PORT\_STAT | IN-SERVICE |

<see also logic under:   
Logic for identifying a “customer” side in the ICORE.PVC table>

**(D) “Tunnel Vlan” logical PE ports**

see view usrp.TUNNEL\_VLAN\_PE\_PORT\_VW and HLD-293772-CR160298-GCP-FLOW-USRP-104-D [Usrp Data Flow into Asset PVC (“Tunnel Vlan” “Network Connection”)].

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that the following steps have been executed:*

1. *HLD-254035-GCP-FLOW-ICORE-104-A [Icore Data Flow into Asset PVC (Layer-3 “Network Connection”)]*
2. *HLD-254035-GCP-FLOW-ICORE-104-B [Icore Data Flow into Asset PVC (Layer-2 "Network Connection")]*
3. *HLD-254035-GCP-FLOW-ICORE-104-C [Icore Data Flow into Asset PVC (“Internet VLAN” “Network Connection”)]*
4. *HLD-254035-GCP-FLOW-ICORE-110-A [Icore Data Flow into Log Port SITE (Layer-3 “Log Port”)]*
5. *HLD-254035-GCP-FLOW-ICORE-110-B [Icore Data Flow into Log Port SITE (Layer-2 “Log Port”)]*
6. *HLD-254035-GCP-FLOW-ICORE-110-C [Icore Data Flow into Log Port SITE (Internet VLAN “Log Port”)]*
7. *HLD-293772-CR160298-GCP-FLOW-USRP-104-D [Usrp Data Flow into Asset PVC (“Tunnel Vlan” “Network Connection”)]*

#### HLD-254035-GCP-FLOW-ICORE-209-A [Icore Data Flow into Asset (“Access Circuit”) to Organization CUSTOMER]

Implement the following:

Process the following Icore records into the GDB asset to organization association and into the Data Index (ref. Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**ORGANIZATION**’, ROLE with FUNCTION\_TYPE ‘**CONTRACTED\_BY**’ and FUNCTION\_ROLE NULL |
| CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id +  CUST\_ACCESS\_CKT.seq\_no | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| CUST\_ACCESS.cust\_id (=CUSTOMER.cust\_id)  <or>  CUST\_ACCESS.cust\_id  <via  CUST\_ACCESS.site\_id =  CUST\_ACCESS\_CKT.site\_id  >  (=CUSTOMER.cust\_id)  Or  customer\_ubsubaccount\_vw.site\_id <302503>  <via cust\_access.site\_id = site.site\_id> | Id\_object\_to [via Enterprise Key Translation for CUSTOMER.cust\_id] | Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id  Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules] where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id |
| CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id +  CUST\_ACCESS\_CKT.seq\_no | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that the following steps have been executed:*

1. *HLD-254035-GCP-FLOW-ICORE-100 [Icore Data Flow into Organization CUSTOMER]*
2. *HLD-254035-GCP-FLOW-ICORE-102-A [Icore Data Flow into Asset CUST\_ACCESS (“Access Circuit”)]*
3. *HLD-254035-GCP-FLOW-ICORE-102-B [Icore Data Flow into Asset CUST\_ACCESS\_CKT (“Access Circuit”)]*

#### HLD-254035-GCP-FLOW-ICORE-209-B [Icore Data Flow into Asset (“Access Circuit”) to Organization MCN\_GRC\_SOC\_SOURCE]

Implement the following:

Process the following Icore records into the GDB asset to organization association and into the Data Index (ref. Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

NOTE that here more than one association record (having different “MCN\_GRC\_SOC\_SOURCE.id” values) may be possible !

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**ORGANIZATION**’, ROLE with FUNCTION\_TYPE ‘**CONTRACTED\_BY**’ and FUNCTION\_ROLE NULL |
| CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id +  CUST\_ACCESS\_CKT.seq\_no | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| MCN\_GRC\_SOC\_SOURCE.id (via database traversal given below) | Id\_object\_to [via Enterprise Key Translation for MCN\_GRC\_SOC\_SOURCE.id] | Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id  Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules] where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id |
| CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id +  CUST\_ACCESS\_CKT.seq\_no | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

Database traversals (1: CUST\_ACCESS.site\_id):

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| CUST\_ACCESS.site\_id | MCN\_GRC\_SOC\_CIRCUIT\_IDS.circuit\_id\_site\_id |
| <literal numerical 0> | MCN\_GRC\_SOC\_CIRCUIT\_IDS.circuit\_id\_seq\_no |
| MCN\_GRC\_SOC\_CIRCUIT\_IDS.id\_m\_g\_s\_source | MCN\_GRC\_SOC\_SOURCE.id |

Database traversals (2: CUST\_ACCESS\_CKT.site\_id):

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| CUST\_ACCESS\_CKT.site\_id | MCN\_GRC\_SOC\_CIRCUIT\_IDS.circuit\_id\_site\_id |
| CUST\_ACCESS\_CKT.seq\_no | MCN\_GRC\_SOC\_CIRCUIT\_IDS.circuit\_id\_seq\_no |
| MCN\_GRC\_SOC\_CIRCUIT\_IDS.id\_m\_g\_s\_source | MCN\_GRC\_SOC\_SOURCE.id |

**Constraints**:

Only take into account “MCN\_GRC\_SOC\_SOURCE.id” values for which all three components “MCN”, GRC” and “SOC” in the corresponding record are defined, i.e. all three values are not equal to “<<NONE>>”.

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that the following steps have been executed:*

1. *HLD-254035-GCP-FLOW-ICORE-100-B [Icore Data Flow into Organization MCN\_GRC\_SOC\_SOURCE]*
2. *HLD-254035-GCP-FLOW-ICORE-102-A [Icore Data Flow into Asset CUST\_ACCESS (“Access Circuit”)]*
3. *HLD-254035-GCP-FLOW-ICORE-102-B [Icore Data Flow into Asset CUST\_ACCESS\_CKT (“Access Circuit”)]*

#### HLD-286282-US704600-GCP-FLOW-ICORE-209-B2 [Icore Data Flow into Asset (“Access Circuit”) to Organization Billing Accounts]

Implement the following:

Process the following Icore records into the GDB asset to organization association and into the Data Index (ref. Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

NOTE that here more than one association record (having different ACCT\_TYPE\_CD values) may be possible !

Note: only load association for acct\_type\_cd = ‘AS’ or ‘XX’ (<Ticket-000000224475375>)

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**ORGANIZATION**’, ROLE with FUNCTION\_TYPE ‘**BILLED\_BY**’ (ACCT\_TYPE\_CD is ‘XX’), or ‘**CONTRACTED\_BY**’ (ACCT\_TYPE\_CD is ‘AS’) and FUNCTION\_ROLE NULL (<Ticket-000000224475375>) |
| CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id +  CUST\_ACCESS\_CKT.seq\_no | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id |
| GDB\_BILLING\_ACCOUNT.account\_key (via database traversal given below) | Id\_object\_to [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id  Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules] where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id |
| CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id +  CUST\_ACCESS\_CKT.seq\_no | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id |
| - | Gdb\_internal\_flags | NULL |

Database traversals (1: CUST\_ACCESS.site\_id):

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| CUST\_ACCESS.site\_id | SITE2BILLING\_ACCOUNT.site\_id |
| SITE2BILLING\_ACCOUNT.account\_key | GDB\_BILLING\_ACCOUNT.account\_key |

<Ticket-000000224475375>

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| CUST\_ACCESS.site\_id | CKT2BILLING\_ACCOUNT.site\_id |
| CKT2BILLING\_ACCOUNT.account\_key | GDB\_BILLING\_ACCOUNT.account\_key |
| <literal numerical 0> | CKT2BILLING\_ACCOUNT.seq\_no |

</Ticket-000000224475375>

Database traversals (2: CUST\_ACCESS\_CKT.site\_id):

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| CUST\_ACCESS\_CKT.site\_id | SITE2BILLING\_ACCOUNT.site\_id |
| SITE2BILLING\_ACCOUNT.account\_key | GDB\_BILLING\_ACCOUNT.account\_key |

<Ticket-000000224475375>

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| CUST\_ACCESS\_CKT.site\_id | CKT2BILLING\_ACCOUNT.site\_id |
| CKT2BILLING\_ACCOUNT.account\_key | GDB\_BILLING\_ACCOUNT.account\_key |
| CUST\_ACCESS\_CKT.seq\_no | CKT2BILLING\_ACCOUNT.seq\_no |

</Ticket-000000224475375>

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

#### HLD-254035-GCP-FLOW-ICORE-209-C [Icore Data Flow into Asset ( “Network Connection”) to Organization CUSTOMER]

Implement the following:

Process the following Icore records into the GDB asset to organization association and into the Data Index (ref. Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**ORGANIZATION**’, ROLE with FUNCTION\_TYPE ‘**CONTRACTED\_BY**’ and FUNCTION\_ROLE NULL |
| **(1) “Layer-3” and “Layer-2” Network Connections:**  NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id;  **(2) “Internet VLAN” Network Connections:**  NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id;  **(3) “Tunnel VLAN” Network Connections:**  usrp.tunnel\_vlan\_vw.icore\_pvc\_id  <293772-CR160298> | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| **(1) “Layer-3” and “Layer-2” Network Connections:**  NETWORK\_CONNECTION.cust\_id (=CUSTOMER.cust\_id)  Or  customer\_ubsubaccount\_vw.site\_id <302503>  <via network\_connection.cust\_id = site.cust\_id>  **(2) “Internet VLAN” Network Connections:**  NETWORK\_CONNECTION\_INET\_VLAN.cust\_id (=CUSTOMER.cust\_id)  **(3) “Tunnel VLAN” Network Connections:**  Icore.site.cust\_id for usrp.tunnel\_vlan\_vw.icore\_site\_id (=CUSTOMER.cust\_id)  <293772-CR160298> | Id\_object\_to [via Enterprise Key Translation for CUSTOMER.cust\_id] | Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id  Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules] where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id |
| **(1) “Layer-3” and “Layer-2” Network Connections:**  NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id;  **(2) “Internet VLAN” Network Connections:**  NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id;  **(3) “Tunnel VLAN” Network Connections:**  usrp.tunnel\_vlan\_vw.icore\_pvc\_id  <293772-CR160298> | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that the following steps have been executed:*

1. *HLD-254035-GCP-FLOW-ICORE-100 [Icore Data Flow into Organization CUSTOMER]*
2. *HLD-254035-GCP-FLOW-ICORE-104-A [Icore Data Flow into Asset PVC (Layer-3 “Network Connection”)]*
3. *HLD-254035-GCP-FLOW-ICORE-104-B [Icore Data Flow into Asset PVC (Layer-2 "Network Connection")]*
4. *HLD-254035-GCP-FLOW-ICORE-104-C [Icore Data Flow into Asset PVC (“Internet VLAN” “Network Connection”)]*
5. HLD-293772-CR160298-GCP-FLOW-USRP-104-D [Usrp Data Flow into Asset PVC (“Tunnel Vlan” “Network Connection”)]

#### HLD-254035-GCP-FLOW-ICORE-209-D [Icore Data Flow into Asset ( “Network Connection”) to Organization MCN\_GRC\_SOC\_SOURCE]

Implement the following:

Process the following Icore records into the GDB asset to organization association and into the Data Index (ref. Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

NOTE that here more than one association record (having different “MCN\_GRC\_SOC\_SOURCE.id” values) may be possible !

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**ORGANIZATION’**, ROLE with FUNCTION\_TYPE ‘**CONTRACTED\_BY**’ and FUNCTION\_ROLE NULL |
| **(1) “Layer-3” and “Layer-2” Network Connections:**  NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id;  **(2) “Internet VLAN” Network Connections:**  NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id;  **(3) “Tunnel VLAN” Network Connections:**  usrp.tunnel\_vlan\_vw.icore\_pvc\_id  <293772-CR160298> | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| MCN\_GRC\_SOC\_SOURCE.id (via database traversal given below) | Id\_object\_to [via Enterprise Key Translation for MCN\_GRC\_SOC\_SOURCE.id] | Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id  Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules] where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id |
| **(1) “Layer-3” and “Layer-2” Network Connections:**  NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id;  **(2) “Internet VLAN” Network Connections:**  NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id;  **(3) “Tunnel VLAN” Network Connections:**  usrp.tunnel\_vlan\_vw.icore\_pvc\_id  <293772-CR160298> | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

Database traversals (from “NETWORK\_CONNECTION”):

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| NETWORK\_CONNECTION.site\_id | MCN\_GRC\_SOC\_SITE\_IDS.site\_id |
| MCN\_GRC\_SOC\_SITE\_IDS.id\_m\_g\_s\_source | MCN\_GRC\_SOC\_SOURCE.id |

Database traversals (from “NETWORK\_CONNECTION\_INET\_VLAN”):

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| NETWORK\_CONNECTION\_INET\_VLAN.site\_id | MCN\_GRC\_SOC\_SITE\_IDS.site\_id |
| MCN\_GRC\_SOC\_SITE\_IDS.id\_m\_g\_s\_source | MCN\_GRC\_SOC\_SOURCE.id |

Database traversals (from “TUNNEL\_VLAN\_VW”):

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| TUNNEL\_VLAN\_VW.site\_id | MCN\_GRC\_SOC\_SITE\_IDS.site\_id |
| MCN\_GRC\_SOC\_SITE\_IDS.id\_m\_g\_s\_source | MCN\_GRC\_SOC\_SOURCE.id |

**Constraints**:

Only take into account “MCN\_GRC\_SOC\_SOURCE.id” values for which all three components “MCN”, GRC” and “SOC” in the corresponding record are defined, i.e. all three values are not equal to “<<NONE>>”.

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that the following steps have been executed:*

1. *HLD-254035-GCP-FLOW-ICORE-100-B [Icore Data Flow into Organization MCN\_GRC\_SOC\_SOURCE]*
2. *HLD-254035-GCP-FLOW-ICORE-104-A [Icore Data Flow into Asset PVC (Layer-3 “Network Connection”)]*
3. *HLD-254035-GCP-FLOW-ICORE-104-B [Icore Data Flow into Asset PVC (Layer-2 "Network Connection")]*
4. *HLD-254035-GCP-FLOW-ICORE-104-C [Icore Data Flow into Asset PVC (“Internet VLAN” “Network Connection”)]*
5. HLD-293772-CR160298-GCP-FLOW-USRP-104-D [Usrp Data Flow into Asset PVC (“Tunnel Vlan” “Network Connection”)]

#### HLD-286282-US704600-GCP-FLOW-ICORE-209-D2 [Icore Data Flow into Asset ( “Network Connection”) to Organization Billing Accounts ]

Implement the following:

Process the following Icore records into the GDB asset to organization association and into the Data Index (ref. Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

NOTE that here more than one association record (having different “ACCT\_TYPE\_CD” values) may be possible !

Note: only load association for acct\_type\_cd = ‘AS’, or ‘AM’ --- ticket 223595795

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**ORGANIZATION’**, ROLE with FUNCTION\_TYPE ~~‘~~**~~BILLED\_BY~~**~~’ (ACCT\_TYPE\_CD is not ‘AS’), or~~ ‘**CONTRACTED\_BY**’ ~~(ACCT\_TYPE\_CD is ‘AS’)~~ and FUNCTION\_ROLE NULL |
| **(1) “Layer-3” and “Layer-2” Network Connections:**  NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id;  **(2) “Internet VLAN” Network Connections:**  NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id;  **(3) “Tunnel VLAN” Network Connections:**  usrp.tunnel\_vlan\_vw.icore\_pvc\_id  <293772-CR160298> | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| GDB\_BILLING\_ACCOUNT.account\_key (via database traversal given below) | Id\_object\_to [via Enterprise Key Translation ] | Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id  Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules] where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id |
| **(1) “Layer-3” and “Layer-2” Network Connections:**  NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id;  **(2) “Internet VLAN” Network Connections:**  NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id;  **(3) “Tunnel VLAN” Network Connections:**  usrp.tunnel\_vlan\_vw.icore\_pvc\_id  <293772-CR160298> | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id |
| - | Gdb\_internal\_flags | NULL |

Database traversals (from “NETWORK\_CONNECTION”):

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| NETWORK\_CONNECTION.site\_id | SITE2BILLING\_ACCOUNT.icore\_site\_id |
| SITE2BILLING\_ACCOUNT.account\_key | GDB\_BILLING\_ACCOUNT.account\_key |

Database traversals (from “NETWORK\_CONNECTION\_INET\_VLAN”):

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| NETWORK\_CONNECTION\_INET\_VLAN.site\_id | SITE2BILLING\_ACCOUNT.icore\_site\_id |
| SITE2BILLING\_ACCOUNT.account\_key | GDB\_BILLING\_ACCOUNT.account\_key |

Database traversals (from “TUNNEL\_VLAN\_VW”):

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| TUNNEL\_VLAN\_VW.site\_id | SITE2BILLING\_ACCOUNT.icore\_site\_id |
| SITE2BILLING\_ACCOUNT.account\_key | GDB\_BILLING\_ACCOUNT.account\_key |

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

#### HLD-254035-GCP-FLOW-ICORE-209-E [Icore Data Flow into Asset (Layer-3 “Customer Network”) to Organization CUSTOMER]

Implement the following:

Process the following Icore records into the GDB asset to organization association and into the Data Index (ref. Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**ORGANIZATION**’, ROLE with FUNCTION\_TYPE ‘**CONTRACTED\_BY**’ and FUNCTION\_ROLE NULL |
| VPN.vpn\_id | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| VPN.cust\_id (=CUSTOMER.cust\_id)  Or  customer\_ubsubaccount\_vw.site\_id <302503> | Id\_object\_to [via Enterprise Key Translation for CUSTOMER.cust\_id or site.site\_id] | Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id  Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules] where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id |
| VPN.vpn\_id | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that the following steps have been executed:*

1. *HLD-254035-GCP-FLOW-ICORE-100 [Icore Data Flow into Organization CUSTOMER]*
2. *HLD-254035-GCP-FLOW-ICORE-103-A [Icore Data Flow into Asset VPN (Layer-3 “Customer Network”)]*

#### HLD-254035-GCP-FLOW-ICORE-209-F [Icore Data Flow into Asset (Layer-3 “Customer Network”) to Organization MCN\_GRC\_SOC\_SOURCE]

Implement the following:

Process the following Icore records into the GDB asset to organization association and into the Data Index (ref. Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

NOTE that here more than one association record (having different “MCN\_GRC\_SOC\_SOURCE.id” values) may be possible !

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘ASSET’, OBJECT\_TYPE\_TO ‘ORGANIZATION’, ROLE with FUNCTION\_TYPE ‘CONTRACTED\_BY’ and FUNCTION\_ROLE NULL |
| VPN.vpn\_id | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| MCN\_GRC\_SOC\_SOURCE.id (via database traversal given below) | Id\_object\_to [via Enterprise Key Translation for MCN\_GRC\_SOC\_SOURCE.id] | Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id  Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules] where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id |
| VPN.vpn\_id | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| VPN.cust\_id | MCN\_GRC\_SOC\_SOURCE.icore\_cust\_id |
| MCN\_GRC\_SOC\_SITE\_IDS.id\_m\_g\_s\_source | MCN\_GRC\_SOC\_SOURCE.id |
| MCN\_GRC\_SOC\_SITE\_IDS.site\_id | NETWORK\_CONNECTION.site\_id |

**Constraints**:

(1) Only take into account “MCN\_GRC\_SOC\_SOURCE.id” values for which all three components “MCN”, GRC” and “SOC” in the corresponding record are defined, i.e. all three values are not equal to “<<NONE>>”.

(2) Using the Enterprise Key Translation, an association (*Referencing ASSOCIATION\_TYPE.id which represents:OBJECT\_TYPE\_WHAT ‘****ASSET****’, OBJECT\_TYPE\_TO ‘****ASSET****’, ROLE with FUNCTION\_TYPE ‘****PART\_OF****’ and FUNCTION\_ROLE NULL*) must exist between “VPN.vpn\_id” and “network\_connection.site\_id” (+ “network\_connection.pvc\_id”).

(This constraint is added here to make sure we only add MCN/GRC/SOC triplets and their organizations where a corresponding network connection (belonging to the current customer network (i.e. “VPN”) exists; anything else must not be included !)

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that the following steps have been executed:*

1. *HLD-254035-GCP-FLOW-ICORE-206-A [Icore Data Flow into Asset (“Customer Network for Network Connections”(Layer-3)) to Asset]*
2. *HLD-254035-GCP-FLOW-ICORE-100 [Icore Data Flow into Organization CUSTOMER]*
3. *HLD-254035-GCP-FLOW-ICORE-103-A [Icore Data Flow into Asset VPN (Layer-3 “Customer Network”)]*

#### HLD-254035-GCP-FLOW-ICORE-209-G [Icore Data Flow into Asset (Layer-2 “Customer Network”) to Organization CUSTOMER]

Implement the following:

Process the following Icore records into the GDB asset to organization association and into the Data Index (ref. Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

*NOTE: For the following we may have to deal with 2 different customers: The one for the “l” side (Record 1) and the one for the “r” side (Record2). So if for the corresponding PVC record (Layer-2 “Network Connection”) “pvc\_lcust\_id != pvc\_rcust\_id”, then create 2 association records, one for the 1. and one for the 2. Customer. If “pvc\_lcust\_id = pvc\_rcust\_id”, then create only one record.*

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**ORGANIZATION**’, ROLE with FUNCTION\_TYPE ‘**CONTRACTED\_BY**’ and FUNCTION\_ROLE NULL |
| PVC.pvc\_id | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| PVC.pvc\_lcust\_id (=CUSTOMER.cust\_id)  (=”local leg” customer;   **1. record**)  <and>  PVC.pvc\_rcust\_id (=CUSTOMER.cust\_id)  (=”remote leg” customer;  **2. record**)  Or  customer\_ubsubaccount\_vw.site\_id <302503>  *(only create 2 records, if “pvc\_lcust\_id != pvc\_rcust\_id”)* | Id\_object\_to [via Enterprise Key Translation for CUSTOMER.cust\_id] | Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id  Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules] where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id |
| PVC.pvc\_id | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that the following steps have been executed:*

1. *HLD-254035-GCP-FLOW-ICORE-100 [Icore Data Flow into Organization CUSTOMER]*
2. *HLD-254035-GCP-FLOW-ICORE-103-B [Icore Data Flow into Asset PVC (as VPN) (Layer-2 "Customer Network")]*

#### HLD-254035-GCP-FLOW-ICORE-209-H [Icore Data Flow into Asset (Layer-2 “Customer Network”) to Organization MCN\_GRC\_SOC\_SOURCE]

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**ORGANIZATION**’, ROLE with FUNCTION\_TYPE ‘**CONTRACTED\_BY**’ and FUNCTION\_ROLE NULL |
| PVC.pvc\_id | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| MCN\_GRC\_SOC\_SOURCE.id (via database traversal given below)  (**1. processing for** (PVC.pvc\_lcust\_id)  <and>  **2.processing for** (PVC.pvc\_rcust\_id))  *(only apply 2 processings, if “pvc\_lcust\_id != pvc\_rcust\_id”)* | Id\_object\_to [via Enterprise Key Translation for MCN\_GRC\_SOC\_SOURCE.id] | Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id  Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules] where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id |
| PVC.pvc\_id | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

Implement the following:

Process the following Icore records into the GDB asset to organization association and into the Data Index (ref. Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

*NOTE: For the following we may have to deal with 2 different customers: The one for the “l” side (Record 1) and the one for the “r” side (Record2). So if for the corresponding PVC record (Layer-2 “Network Connection”) “pvc\_lcust\_id != pvc\_rcust\_id”, then create 2 association records, one for the 1. and one for the 2. Customer. If “pvc\_lcust\_id = pvc\_rcust\_id”, then create only one record.*

Database traversals (1. PVC.pvc\_lcust\_id):

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| PVC.pvc\_id | NETWORK\_CONNECTION.pvc\_id |
| PVC.pvc\_lsite\_id | NETWORK\_CONNECTION.site\_id |
| MCN\_GRC\_SOC\_SITE\_IDS.site\_id | NETWORK\_CONNECTION.site\_id |
| MCN\_GRC\_SOC\_SITE\_IDS.id\_m\_g\_s\_source | MCN\_GRC\_SOC\_SOURCE.id |

Database constraints:

*(PVC “l” side)*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| NETWORK\_CONNECTION | SIDE | ‘local’ |

Database traversals (2. PVC.pvc\_rcust\_id (if “PVC.pvc\_lcust\_id != PVC.pvc\_rcust\_id”)):

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| PVC.pvc\_id | NETWORK\_CONNECTION.pvc\_id |
| PVC.pvc\_rsite\_id | NETWORK\_CONNECTION.site\_id |
| MCN\_GRC\_SOC\_SITE\_IDS.site\_id | NETWORK\_CONNECTION.site\_id |
| MCN\_GRC\_SOC\_SITE\_IDS.id\_m\_g\_s\_source | MCN\_GRC\_SOC\_SOURCE.id |

Database constraints:

*(PVC “r” side)*

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| NETWORK\_CONNECTION | SIDE | ‘remote’ |

**Constraints**:

(1) Only take into account “MCN\_GRC\_SOC\_SOURCE.id” values for which all three components “MCN”, GRC” and “SOC” in the corresponding record are defined, i.e. all three values are not equal to “<<NONE>>”.

(2) Using the Enterprise Key Translation, an association (*Referencing ASSOCIATION\_TYPE.id which represents:OBJECT\_TYPE\_WHAT ‘****ASSET****’, OBJECT\_TYPE\_TO ‘****ASSET****’, ROLE with FUNCTION\_TYPE ‘****PART\_OF****’ and FUNCTION\_ROLE NULL*) must exist between “PVC.pvc\_id” and “network\_connection.site\_id” (+ “network\_connection.pvc\_id”).

(This constraint is added here to make sure we only add MCN/GRC/SOC triplets and their organizations where a corresponding network connection (belonging to the current customer network (i.e. layer-2 “PVC”) exists; anything else must not be included !)

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that the following steps have been executed:*

1. *HLD-254035-GCP-FLOW-ICORE-206-B [Icore Data Flow into Asset (“Customer Network for Network Connections”(Layer-2)) to Asset]*
2. *HLD-254035-GCP-FLOW-ICORE-100 [Icore Data Flow into Organization CUSTOMER]*
3. *HLD-254035-GCP-FLOW-ICORE-103-B [Icore Data Flow into Asset PVC (as VPN) (Layer-2 "Customer Network")]*

#### HLD-254035-GCP-FLOW-ICORE-209-I [Icore Data Flow into Asset (Internet VLAN “Customer Network”) to Organization CUSTOMER]

Implement the following:

Process the following Icore records into the GDB asset to organization association and into the Data Index (ref. Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**ORGANIZATION**’, ROLE with FUNCTION\_TYPE ‘**CONTRACTED\_BY**’ and FUNCTION\_ROLE NULL |
| CUSTOMER\_NETWORK\_INET\_VLAN.vpn\_id + CUSTOMER\_NETWORK\_INET\_VLAN.pvc\_id + CUSTOMER\_NETWORK\_INET\_VLAN.site\_id | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| CUSTOMER\_NETWORK\_INET\_VLAN.cust\_id (=CUSTOMER.cust\_id) | Id\_object\_to [via Enterprise Key Translation for CUSTOMER.cust\_id] | Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id  Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules] where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id |
| CUSTOMER\_NETWORK\_INET\_VLAN.vpn\_id + CUSTOMER\_NETWORK\_INET\_VLAN.pvc\_id + CUSTOMER\_NETWORK\_INET\_VLAN.site\_id | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that the following steps have been executed:*

1. *HLD-254035-GCP-FLOW-ICORE-100 [Icore Data Flow into Organization CUSTOMER]*
2. *HLD-254035-GCP-FLOW-ICORE-103-C [Icore Data Flow into Asset VPN (“Internet VLAN” “Customer Network”)]*

#### HLD-254035-GCP-FLOW-ICORE-209-J [Icore Data Flow into Asset (Internet VLAN “Customer Network”) to Organization MCN\_GRC\_SOC\_SOURCE]

Implement the following:

Process the following Icore records into the GDB asset to organization association and into the Data Index (ref. Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**ORGANIZATION**’, ROLE with FUNCTION\_TYPE ‘**CONTRACTED\_BY**’ and FUNCTION\_ROLE NULL |
| CUSTOMER\_NETWORK\_INET\_VLAN.vpn\_id + CUSTOMER\_NETWORK\_INET\_VLAN.pvc\_id + CUSTOMER\_NETWORK\_INET\_VLAN.site\_id | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| MCN\_GRC\_SOC\_SOURCE.id (via database traversal given below) | Id\_object\_to [via Enterprise Key Translation for MCN\_GRC\_SOC\_SOURCE.id] | Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id  Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules] where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id |
| CUSTOMER\_NETWORK\_INET\_VLAN.vpn\_id + CUSTOMER\_NETWORK\_INET\_VLAN.pvc\_id + CUSTOMER\_NETWORK\_INET\_VLAN.site\_id | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| CUSTOMER\_NETWORK\_INET\_VLAN.site\_id | MCN\_GRC\_SOC\_SITE\_IDS.site\_id |
| MCN\_GRC\_SOC\_SOURCE.id | MCN\_GRC\_SOC\_SITE\_IDS.id\_m\_g\_s\_source |

**Constraints**:

(1) Only take into account “MCN\_GRC\_SOC\_SOURCE.id” values for which all three components “MCN”, GRC” and “SOC” in the corresponding record are defined, i.e. all three values are not equal to “<<NONE>>”.

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that the following steps have been executed:*

1. *HLD-254035-GCP-FLOW-ICORE-100 [Icore Data Flow into Organization CUSTOMER]*
2. *HLD-254035-GCP-FLOW-ICORE-103-C [Icore Data Flow into Asset VPN (“Internet VLAN” “Customer Network”)]*

#### HLD- CR-155856\_Defect-50973-GCP-FLOW-ICORE-209-K [Icore Data Flow into Asset (Internet VLAN “Network Connection”) to (Internet VLAN “Network Connection”)]

<CR-155856\_Defect-50973>

Implement the following:

Process the following Icore records into the GDB asset to organization association and into the Data Index (ref. Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]):

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**ASSET**’, ROLE with FUNCTION\_TYPE ‘**IS\_BACKUP\_FOR**’ and FUNCTION\_ROLE NULL |
| NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  Use site\_id where SERVICE\_OPTION\_VALUES.value = ‘Backup’ (see [Logic for retrieving Primary and Backup Internet VLAN Network Connection](#_Logic_for_retrieving))  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id | Id\_object\_to [via Enterprise Key Translation for MCN\_GRC\_SOC\_SOURCE.id] | Use SOURCE\_KEY.id which populated the respective ORGANIZATION.id  Use site\_id where SERVICE\_OPTION\_VALUES.value = ‘Primary’ (see [Logic for retrieving Primary and Backup Internet VLAN Network Connection](#_Logic_for_retrieving))  Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-001 [Organization Unification Rules] where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id |
| NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id |
| - | Gdb\_internal\_flags | NULL |

For the ASSOCIATION record created, also update the corresponding GDB.ASSET records as follows:

For the Primary ASSET above (matching id\_object\_to): ASSET.is\_role\_primary = ‘Y’; ASSET.is\_role\_backup = ‘N’

For the Backup ASSET above (matching id\_object\_what): ASSET.is\_role\_primary = ‘N’; ASSET.is\_role\_backup = ‘Y’

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that the following steps have been executed:*

1. *HLD-254035-GCP-FLOW-ICORE-104-C [Icore Data Flow into Asset PVC (“Internet VLAN” “Network Connection”)]*

##### Logic for retrieving Primary and Backup Internet VLAN Network Connection

Use the below query in INSTAR to retrieve the Primary and Backup Internet VLAN Network Connections. The Primary and Backup Network Connections can be determined based on matching SERVICE\_OPTION\_VALUES.value where SERVICE\_OPTION\_VALUES.name = ‘Redundancy Key’. The SERVICE\_OPTION\_VALUES.name should be ‘Dual’ for SERVICE\_OPTION\_VALUES.name = ‘Redundancy Type’ for the same AVPN\_ICORE\_SITE\_ID. The SERVICE\_OPTION\_VALUES.name = ‘Primary’ or ‘Backup’ for SERVICE\_OPTION\_VALUES.name = ‘Redundancy Option’ will indicate which AVPN\_ICORE\_SITE\_ID should be Primary and which should be Backup (see data example below):

select SITE.AVPN\_ICORE\_SITE\_ID, pa\_type, ip\_endpt,pa.IP\_PORT\_ASGMT\_ID,pa.SERV\_ACC\_PT\_ID, so.SERVICE\_OPTION\_ID, sr.SELECTED\_OPT\_ID, so.OPTION\_NAME,

s.SERV\_ID, s.SERV\_NAME, iso.STATUS, sov.NAME, sov.VALUE, so.import\_rt\_group

from instar.ip\_port\_asgmt pa, instar.selected\_opts\_rel sr,

instar.ip\_selected\_opts iso, instar.ip\_service\_options so, instar.service s,

instar.service\_option\_values sov

,instar.site ,

instar.ip\_serv\_acc\_pt sap

where so.option\_name='BGP Redundancy'

and PA.SERV\_ACC\_PT\_ID=SAP.SERV\_ACC\_PT\_ID

and sap.site\_id=site.site\_id

and pa.SERV\_ACC\_PT\_ID = sr.SERV\_ACC\_PT\_ID

and sr.SELECTED\_OPT\_ID = iso.SELECTED\_OPT\_ID

and iso.SERVICE\_OPTION\_ID = so.SERVICE\_OPTION\_ID

and so.SERV\_ID = s.SERV\_ID

and iso.SELECTED\_OPT\_ID = sov.SELECTED\_OPT\_ID(+)

and SITE.AVPN\_ICORE\_SITE\_ID = <site\_id from icore.network\_connection\_inet\_vlan>

order by pa.create\_date desc

|  |  |  |
| --- | --- | --- |
| **AVPN\_ICORE\_SITE\_ID** | **SERVICE\_OPTION\_VALUES.NAME** | **SERVICE\_OPTION\_VALUES.VALUE** |
| 1414799 | Redundancy Key | **569902** |
| 1414799 | Redundancy Option | **Backup** |
| 1414799 | Redundancy Type | Dual |
| 1414796 | Redundancy Key | **569902** |
| 1414796 | Redundancy Option | **Primary** |
| 1414796 | Redundancy Type | Dual |

In the above example, SITE ID 1414796 represents the “Primary” Site and SITE ID 1414799 represents the related “Backup” site

</CR-155856\_Defect-50973>

#### HLD-254035-GCP-FLOW-ICORE-CLEANUP [Icore Data Flow for Deletions]

Implement the following:

Implement asynchronous delete processes based on the Enterprise Key Translation SOURCE\_KEY\_DELETION for the following GDB tables. The asynchronous delete process must ensure that the GDB table content is only deleted after the specified safety duration (i.e. if database instance timestamp of processing time is beyond the SOURCE\_KEY\_DELETION.deletion\_timestamp plus the specified safety duration):

|  |  |  |
| --- | --- | --- |
| **GDB table** | **Safety duration** | **Comment** |
| ORGANIZATION | 5 days | SOURCE\_KEY\_DELETION.id\_source\_key=ORGANIZATION.id |
| SITE | 5 days | SOURCE\_KEY\_DELETION.id\_source\_key=SITE.id |
| ASSET | 5 days | SOURCE\_KEY\_DELETION.id\_source\_key=ASSET.id |
| PHY\_PORT | 5 days | SOURCE\_KEY\_DELETION.id\_source\_key=PHY\_PORT.id |
| LOG\_PORT | 5 days | SOURCE\_KEY\_DELETION.id\_source\_key=LOG\_PORT.id |
| ASSOCIATION | 1 day | SOURCE\_KEY\_DELETION.id\_source\_key=ASSOCIATION.id\_source\_key |

When deleting records from base tables for an entity, the referencing property/child table records must also be deleted.

For “ASSET” all current asset “satellite/extension” tables (table name starting with “ASSET\_EXT\_”) are considered property/child tables.

**<289037c-ICORE-050>**

For “ASSET”, the GDB ASSET\_ACCESS\_CIRCUIT\_DETAILS, CUSTOMER\_PORT\_DETAILS and PVC\_DETAILS tables are considered property/child tables.

**<End of 289037c-ICORE-050>**

ASSOCIATION records referencing (either via id\_object\_what or id\_object\_to) the base tables for an entity must also be deleted when a base table record gets deleted.

Implement ORGANIZATION deletions according to HLD-232213e-GCP-ORGANIZATION\_UNIFICATION-003 [Organization Deletion Rule].

Process the deleted records into the Data Index (ref. Process HLD-232213e-GCP-DATA-INDEX-001 [Data Index Processing Rules]).

The processed SOURCE\_KEY\_DELETION and referenced SOURCE\_KEY record must be deleted as well.

For deletions first update the GDB tables ID\_CHANGE\_TRACKING to a new record, then process the delete step (this ensures the latest data gets copied to the change history and the delete gets a dedicated change tracking event into the change history as well).

*Note:*

*This step must be executed after the following steps have been completed:*

1. HLD-254035-GCP-FLOW-ICORE-010 [Icore Data Flow: Maintain MCN/GRC/SOC Mapping]
2. HLD-286282-US704600-GCP-FLOW-ICORE-020 [Icore Data Flow: Maintain Billing Account Mapping]
3. HLD-254035-GCP-FLOW-ICORE-100-A [Icore Data Flow into Organization CUSTOMER]
4. HLD-254035-GCP-FLOW-ICORE-100-B [Icore Data Flow into Organization MCN\_GRC\_SOC\_SOURCE]
5. HLD-286282-US704600-GCP-FLOW-ICORE-100-C [Icore Data Flow into Organization BILLING ACCOUNT]
6. HLD-286282-US704600-GCP-FLOW-ICORE-100-D [Icore Data Flow into Organization to Organization]
7. HLD-254035-GCP-FLOW-ICORE-101 [Icore Data Flow into Site]
8. HLD-254035-GCP-FLOW-ICORE-102-A [Icore Data Flow into Asset CUST\_ACCESS (“Access Circuit”)]
9. HLD-254035-GCP-FLOW-ICORE-102-B [Icore Data Flow into Asset CUST\_ACCESS\_CKT (“Access Circuit”)]
10. HLD-254035-GCP-FLOW-ICORE-103-A [Icore Data Flow into Asset VPN (Layer-3 “Customer Network”)]
11. HLD-254035-GCP-FLOW-ICORE-103-B [Icore Data Flow into Asset PVC (as VPN) (Layer-2 "Customer Network")]
12. HLD-254035-GCP-FLOW-ICORE-103-C [Icore Data Flow into Asset VPN (“Internet VLAN” “Customer Network”)]
13. HLD-254035-GCP-FLOW-ICORE-104-A [Icore Data Flow into Asset PVC (Layer-3 “Network Connection”)]
14. HLD-254035-GCP-FLOW-ICORE-104-B [Icore Data Flow into Asset PVC (Layer-2 "Network Connection")]
15. HLD-254035-GCP-FLOW-ICORE-104-C [Icore Data Flow into Asset PVC (“Internet VLAN” “Network Connection”)]
16. HLD-293772-CR160298-GCP-FLOW-USRP-104-D [Usrp Data Flow into Asset PVC (“Tunnel Vlan” “Network Connection”)]
17. HLD-254035-GCP-FLOW-ICORE-110-A [Icore Data Flow into Phy Port SITE (Layer-3 “Phy Port”)]
18. HLD-254035-GCP-FLOW-ICORE-110-B [Icore Data Flow into Phy Port SITE (Layer-2 “Phy Port”)]
19. HLD-254035-GCP-FLOW-ICORE-110-D [Icore Data Flow into Phy Port SITE (“Interface Name” data)]
20. HLD-254035-GCP-FLOW-ICORE-110-E [Icore Data Flow into Phy Port SITE (“id\_asset Equipment” data)]
21. HLD-254035-GCP-FLOW-ICORE-120-A [Icore Data Flow into Log Port SITE (Layer-3 “Log Port”)]
22. HLD-254035-GCP-FLOW-ICORE-120-B [Icore Data Flow into Log Port SITE (Layer-2 “Log Port”)]
23. HLD-254035-GCP-FLOW-ICORE-120-C [Icore Data Flow into Log Port SITE (Internet VLAN “Log Port”)]
24. HLD-254035-GCP-FLOW-ICORE-120-D [Icore Data Flow into Log Port SITE (“Interface Name” data)]
25. HLD-254035-GCP-FLOW-ICORE-120-E [Icore Data Flow into Log Port SITE (“id\_asset Equipment” data)]
26. HLD-293772-CR160298-GCP-FLOW-USRP-120-G [Icore Data Flow into Log Port SITE (Tunnel Vlan "Log Port")]
27. HLD-254035-GCP-FLOW-ICORE-130 [Icore Data Flow into Service]
28. HLD-254035-GCP-FLOW-ICORE-200-A [Icore Data Flow into Site to Organization (via “CUSTOMER”)]
29. HLD-254035-GCP-FLOW-ICORE-200-B [Icore Data Flow into Site to Organization (via “MRC\_GRC\_SOC\_SOURCE”)]
30. HLD-254035-GCP-FLOW-ICORE-201-A [Icore Data Flow into Asset (“Access Circuit”) to Site]
31. HLD-254035-GCP-FLOW-ICORE-201-B [Icore Data Flow into Asset (“Customer Network”) to Site]
32. HLD-254035-GCP-FLOW-ICORE-201-C [Icore Data Flow into Asset (“Network Connection”) to Site]
33. HLD-254035-GCP-FLOW-ICORE-202-A [Icore Data Flow into Asset (“Access Circuit”) to Service]
34. HLD-254035-GCP-FLOW-ICORE-202-B [Icore Data Flow into Asset (“Customer Network”) to Service]
35. HLD-254035-GCP-FLOW-ICORE-202-C [Icore Data Flow into Asset (“Network Connection”) to Service]
36. HLD-254035-GCP-FLOW-ICORE-203 [Icore Data Flow into Asset (“Access Circuit Parent”) to Asset]
37. HLD-254035-GCP-FLOW-ICORE-204 [Icore Data Flow into Port (“Access Circuit Port”) to Asset]
38. HLD-254035-GCP-FLOW-ICORE-205 [Icore Data Flow into Asset (“Network Connection Access Circuits”) to Asset]
39. HLD-254035-GCP-FLOW-ICORE-206-A [Icore Data Flow into Asset (“Customer Network for Network Connections”(Layer-3)) to Asset]
40. HLD-254035-GCP-FLOW-ICORE-206-B [Icore Data Flow into Asset (“Customer Network for Network Connections”(Layer-2)) to Asset]
41. HLD-254035-GCP-FLOW-ICORE-206-C [Icore Data Flow into Asset (“Customer Network for Network Connections”(Internet VLAN)) to Asset]
42. HLD-254035-GCP-FLOW-ICORE-207 [Icore Data Flow into Port (“Network Connection Port”) to Asset]
43. HLD-254035-GCP-FLOW-ICORE-209-A [Icore Data Flow into Asset (“Access Circuit”) to Organization CUSTOMER]
44. HLD-254035-GCP-FLOW-ICORE-209-B [Icore Data Flow into Asset (“Access Circuit”) to Organization MCN\_GRC\_SOC\_SOURCE]
45. HLD-286282-US704600-GCP-FLOW-ICORE-209-B2 [Icore Data Flow into Asset (“Access Circuit”) to Organization Billing Accounts]
46. HLD-254035-GCP-FLOW-ICORE-209-C [Icore Data Flow into Asset ( “Network Connection”) to Organization CUSTOMER]
47. HLD-254035-GCP-FLOW-ICORE-209-D [Icore Data Flow into Asset ( “Network Connection”) to Organization MCN\_GRC\_SOC\_SOURCE]
48. HLD-286282-US704600-GCP-FLOW-ICORE-209-D2 [Icore Data Flow into Asset ( “Network Connection”) to Organization Billing Accounts ]
49. HLD-254035-GCP-FLOW-ICORE-209-E [Icore Data Flow into Asset (Layer-3 “Customer Network”) to Organization CUSTOMER]
50. HLD-254035-GCP-FLOW-ICORE-209-F [Icore Data Flow into Asset (Layer-3 “Customer Network”) to Organization MCN\_GRC\_SOC\_SOURCE]
51. HLD-254035-GCP-FLOW-ICORE-209-G [Icore Data Flow into Asset (Layer-2 “Customer Network”) to Organization CUSTOMER]
52. HLD-254035-GCP-FLOW-ICORE-209-H [Icore Data Flow into Asset (Layer-2 “Customer Network”) to Organization MCN\_GRC\_SOC\_SOURCE]
53. HLD-254035-GCP-FLOW-ICORE-209-I [Icore Data Flow into Asset (Internet VLAN “Customer Network”) to Organization CUSTOMER]
54. HLD-254035-GCP-FLOW-ICORE-209-J [Icore Data Flow into Asset (Internet VLAN “Customer Network”) to Organization MCN\_GRC\_SOC\_SOURCE]

*Note:*

*Ensure that the deliverable of this project does not result in duplicate processing caused by deliverables of previous projects.*

*Note:*

*This process will delete associations sooner than entities. This is to prevent data loss caused by accidential up-stream deletes and provides a window of days to restore accidentially deleted organization, site, asset and port records in source systems.*

#### HLD-254035-GCP-FLOW-INITIALIZATION-ICORE [Icore Data Flow Initialization]

Execute the items of the process implemented by the following design item during deployment:

1. HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process]

#### HLD-254035-GCP-FLOW-SCHEDULE-ICORE [Icore Data Flow Scheduling]

Ensure that the process implemented by the following design item is scheduled to execute twice per day:

1. HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process]

*Note:*

*Ensure that process changes described in this project don’t result in duplicate processing due to the deliverables of earlier projects.*

#### HLD-295359-CR169138-GCP-FLOW-ICORE-210 [Icore Data Flow into GDB]

<295359-US325561>

A one time full load of the GDB NETWORK\_CONNECTION\_DATA table for all UCPE POKs will be needed to initialize this table.

On subsequent loads, a new record should be inserted if a new Access Circuit POK was added to GDB NETWORK\_CONNECTION\_DATA.

|  |  |  |
| --- | --- | --- |
| **ICORE table.column** | **GDB NETWORK\_CONNECTION\_DATA column** | **Processing comment** |
| GDB.ASSET.ID | ACCESS\_CIRCUIT\_POK | Where asset\_type.type=’ACCESS\_CIRCUIT’, Associated to service ‘AVPN’ |
| SELECT 'Y' from icore.service\_asgmt sa,icore.service\_option so, icore.service s where so.serv\_opt\_id=sa.serv\_opt\_id and so.serv\_id=s.serv\_id and trim(serv\_name)='VoIP' and site\_id=<icore\_Site\_Id\_Value>  GRID.PORT.BVOIP\_INDICATOR | ACCESS\_CIRCUIT\_BVOIP\_FLAG | SELECT SKV.SOURCE\_VALUE SITEID  FROM METADATA.SOURCE\_KEY\_VALUE SKV, METADATA.SOURCE\_KEY SK, METADATA.META\_COLUMN MC, METADATA.META\_TABLE MT, METADATA.META\_SYSTEM MS  WHERE SKV.ID\_SOURCE\_KEY = SK.ID  AND SKV.ID\_META\_COLUMN = MC.ID  AND MC.ID\_META\_TABLE = MT.ID  AND SK.ID\_META\_TABLE = MT.ID  AND MT.ID\_META\_SYSTEM = MS.ID  AND MS.NAME = 'ICORE'  AND MC.NAME = 'SITE\_ID'  AND SK.ID = :I\_ASSETID  GRID Network Connection Joins:  GDB.ASSET\_EXT\_ACCESS\_CIRCUIT.ID\_ASSET = GRID.CIRCUIT.ETE\_ACCESS\_KEY  AND GRID.CIRCUIT.ETE\_ACCESS\_KEY = GRID.ACCESS\_CIRCUIT.ETE\_ACCESS\_KEY  AND GRID.ACCESS\_CIRCUIT.ETE\_PORT\_KEY = GRID.PORT.ETE\_PORT\_KEY  Note: If Y -> set as true,  N, Null -> set as false |
| GDB.ASSET.ID | NETWORK\_CONNECTION\_POK | Where asset\_type=’NETWORK\_CONNECTION’  Note: Need to restrict those assets (Network\_Connection) whose Network\_Type (Asset\_Ext\_Network\_Connection.ID\_Network\_Sub\_Type) is NST\_INTERNET, NST\_TUNNEL. <PVT Defect 555310> |
| ~~GRID.PORT.BVOIP\_INDICATOR~~  ICORE.IPFR.BVOIP | NETWORK\_CONNECTION\_BVOIP\_FLAG | ~~GRID Network Connection Joins:~~  ~~GDB.ASSET\_EXT\_NETWORK\_CONNECTION.ID\_ASSET = <Asset ID>~~  ~~GRID.SERVICE\_CONNECTION.ETE\_SERVICE\_CONNECTION\_KEY = GDB.ASSET\_EXT\_NETWORK\_CONNECTION.ETE\_SERVICE\_CONNECTION\_KEY~~  Note: If Y -> set as true,  N, Null -> set as false |
| ICORE.PVC.PVC\_ID | NETWORK\_CONNECTION\_PVC\_ID |  |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| ICORE.VPN.VPN\_ID | Network\_Connection\_vpn\_Id |  |
| GDB.Asset\_Ext\_Network\_Connection.Speed | NETWORK\_CONNECTION\_VLAN\_SPEED |  |
| GDB.Asset\_Ext\_Network\_Connection.Dhv\_Flag | NETWORK\_CONNECTION\_DHV\_FLAG | Note: If Y -> set as true,  N, Null -> set as false |
| ICORE.PVC.PVC\_ID | NETWORK\_CONNECTION\_PVC\_ID |  |

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| ASSET.ID | ASSET\_EXT\_NETWORK\_CONNECTION.ID\_ASSET |
| ICORE.PVC.PVC\_ID | ICORE.NETWORK\_CONNECTION.PVC\_ID |
| ICORE.NETWORK\_CONNECTION.PVC\_ID | GDB.ASSET\_EXT\_NETWORK\_CONNECTION.NETWORK\_CONNECTION\_ID |
| GDB.ASSET\_EXT\_NETWORK\_CONNECTION.ETE\_SERVICE\_CONNECTION\_KEY | GRID.SERVICE\_CONNECTION.ETE\_SERVICE\_CONNECTION\_KEY |
| GDB.ASSET\_EXT\_ACCESS\_CIRCUIT.ETE\_ACCESS\_KEY | GRID.CIRCUIT.ETE\_ACCESS\_KEY |
| GRID.CIRCUIT.ETE\_ACCESS\_KEY | GRID.ACCESS\_CIRCUIT.ETE\_ACCESS\_KEY |
| GRID.ACCESS\_CIRCUIT.ETE\_PORT\_KEY | GRID.PORT.ETE\_PORT\_KEY |
| ASSET.ID (Access\_Circuit) | Association.ID\_Object\_What |
| Association.Id\_Object\_To | Asset.ID(Network\_Connection) |
| ICORE.PVC.PVC\_ID | ICORE.IPFR.PVC\_ID |
| ICORE.IPFR.VPN\_ID | ICORE.VPN.VPN\_ID |

A one time full load of the GDB AVPN\_NETWORK\_CONNECTION\_DATA table for all Access Circuit POKs will be needed to initialize this table.

On subsequent loads, a new record should be inserted if a new Access Circuit POK was added to GDB AVPN\_NETWORK\_CONNECTION\_DATA.

|  |  |  |
| --- | --- | --- |
| **ICORE table.column** | **GDB AVPN\_NETWORK\_CONNECTION\_DATA column** | **Processing comment** |
| GDB.ASSET.ID | ACCESS\_CIRCUIT\_POK | Where asset\_type.type=’ACCESS\_CIRCUIT’, Service=AVPN, and id\_status is other than ‘DELETED’ |
| GDB.ASSET.ID | NETWORK\_CONNECTION\_POK | Where asset\_type=’NETWORK\_CONNECTION’, and id\_status is other than ‘DELETED’ |
| GDB.Asset\_Alias.Value | Cust\_Asset\_Alias\_Name |  |
| NETWORK\_CONNECTION.pvc\_id | Connection\_Id |  |
| NETWORK\_CONNECTION.contrcir | Speed |  |
| ~~Trim(GDB.identifier\_type.type)~~ | ~~Prov\_Edge\_IP\_Data1\_Type~~ | ~~('DLCI\_PORT\_IDENTIFIER', 'PT\_FR\_DLCI',~~  ~~'VPI\_PORT\_IDENTIFIER', 'PT\_ATM\_VPI',~~  ~~'VCI\_PORT\_IDENTIFIER', 'PT\_ATM\_VCI',~~  ~~'UNDEFINED\_PORT\_IDENTIFIER', 'PT\_UNDEFINED',~~  ~~'ETH\_TOP\_PORT\_IDENTIFIER', 'PT\_ETH\_TOP',~~  ~~'ETH\_BOTTOM\_PORT\_IDENTIFIER', 'PT\_ETH\_BOTTOM',~~  ~~'VLAN\_PORT\_IDENTIFIER', 'PT\_VLAN',~~  ~~'INSTAR\_SDID\_PORT\_IDENTIFIER', 'PT\_SDID',~~    ~~'IPV4', 'PT\_IP\_V4',~~  ~~'IPV6', 'PT\_IP\_V6',~~  ~~'PORT\_SAP\_ID\_PORT\_IDENTIFIER', 'PT\_SAP\_ID', 'PT\_UNDEFINED')~~ |
| ~~GDB.Log\_Port\_identifier\_value.value~~ | ~~Prov\_Edge\_IP\_Data1\_Simple\_Network\_Address~~  ~~PROV\_EDGE\_IP\_DATA1\_NET\_ADDRESS~~ |  |
| ~~Trim(GDB.identifier\_type.type)~~ | ~~Prov\_Edge\_IP\_Data2\_Type~~ | ~~('DLCI\_PORT\_IDENTIFIER', 'PT\_FR\_DLCI',~~  ~~'VPI\_PORT\_IDENTIFIER', 'PT\_ATM\_VPI',~~  ~~'VCI\_PORT\_IDENTIFIER', 'PT\_ATM\_VCI',~~  ~~'UNDEFINED\_PORT\_IDENTIFIER', 'PT\_UNDEFINED',~~  ~~'ETH\_TOP\_PORT\_IDENTIFIER', 'PT\_ETH\_TOP',~~  ~~'ETH\_BOTTOM\_PORT\_IDENTIFIER', 'PT\_ETH\_BOTTOM',~~  ~~'VLAN\_PORT\_IDENTIFIER', 'PT\_VLAN',~~  ~~'INSTAR\_SDID\_PORT\_IDENTIFIER', 'PT\_SDID',~~    ~~'IPV4', 'PT\_IP\_V4',~~  ~~'IPV6', 'PT\_IP\_V6',~~  ~~'PORT\_SAP\_ID\_PORT\_IDENTIFIER', 'PT\_SAP\_ID', 'PT\_UNDEFINED')~~ |
| ~~GDB.Log\_Port\_identifier\_value.value~~ | ~~Prov\_Edge\_IP\_Data2\_Simple\_Network\_Address~~  ~~PROV\_EDGE\_IP\_DATA2\_NET\_ADDRESS~~ |  |
| ~~Nc3\_mat.cos\_profile.profile\_number~~  Icore.cos\_profile.profile\_id  GRID.SERVICE\_CONNECTION.PE\_INGRESS\_PROFILE\_ID | Prov\_Edge\_Ingress\_Profile\_Id | If Asset\_Ext\_Network\_Connection.has\_details=’Y’, go to GDB. If data is not there, go to ICORE and if not present go to GRID. |
| ~~Nc3\_mat.cos\_profile.name~~  Icore.cos\_profile.profile\_name | Prov\_Edge\_Ingress\_Profile\_Name | If prov\_edge\_ingress\_profile\_name is not present, pass ‘ ‘, only if Profile\_Id is present. |
| ~~Nc3\_mat.cos\_profile.profile\_number~~  Icore.cos\_profile.profile\_id  GRID.SERVICE\_CONNECTION.PE\_ENGRESS\_PROFILE\_ID | Prov\_Edge\_Egress\_Profile\_Id |  |
| ~~Nc3\_mat.cos\_profile.name~~  Icore.cos\_profile.profile\_name | Prov\_Edge\_Egress\_Profile\_Name | If prov\_edge\_ingress\_profile\_name is not present, pass ‘ ‘, only if Profile\_Id is present. |
|  | ~~Cos\_Detail~~ |  |
| ~~Trim(GDB.identifier\_type.type)~~ | ~~Cust\_Edge\_IP\_Data1\_Type~~ | ~~('DLCI\_PORT\_IDENTIFIER', 'PT\_FR\_DLCI',~~  ~~'VPI\_PORT\_IDENTIFIER', 'PT\_ATM\_VPI',~~  ~~'VCI\_PORT\_IDENTIFIER', 'PT\_ATM\_VCI',~~  ~~'UNDEFINED\_PORT\_IDENTIFIER', 'PT\_UNDEFINED',~~  ~~'ETH\_TOP\_PORT\_IDENTIFIER', 'PT\_ETH\_TOP',~~  ~~'ETH\_BOTTOM\_PORT\_IDENTIFIER', 'PT\_ETH\_BOTTOM',~~  ~~'VLAN\_PORT\_IDENTIFIER', 'PT\_VLAN',~~  ~~'INSTAR\_SDID\_PORT\_IDENTIFIER', 'PT\_SDID',~~    ~~'IPV4', 'PT\_IP\_V4',~~  ~~'IPV6', 'PT\_IP\_V6',~~  ~~'PORT\_SAP\_ID\_PORT\_IDENTIFIER', 'PT\_SAP\_ID', 'PT\_UNDEFINED')~~ |
| ~~GDB.Log\_Port\_identifier\_value.value~~ | ~~Cust\_Edge\_IP\_Data1\_Simple\_Network\_Address~~  ~~CUST\_EDGE\_IP\_DATA1\_NET\_ADDRESS~~ |  |
| ~~Trim(GDB.identifier\_type.type)~~ | ~~Cust\_Edge\_IP\_Data2\_Type~~ | ~~('DLCI\_PORT\_IDENTIFIER', 'PT\_FR\_DLCI',~~  ~~'VPI\_PORT\_IDENTIFIER', 'PT\_ATM\_VPI',~~  ~~'VCI\_PORT\_IDENTIFIER', 'PT\_ATM\_VCI',~~  ~~'UNDEFINED\_PORT\_IDENTIFIER', 'PT\_UNDEFINED',~~  ~~'ETH\_TOP\_PORT\_IDENTIFIER', 'PT\_ETH\_TOP',~~  ~~'ETH\_BOTTOM\_PORT\_IDENTIFIER', 'PT\_ETH\_BOTTOM',~~  ~~'VLAN\_PORT\_IDENTIFIER', 'PT\_VLAN',~~  ~~'INSTAR\_SDID\_PORT\_IDENTIFIER', 'PT\_SDID',~~    ~~'IPV4', 'PT\_IP\_V4',~~  ~~'IPV6', 'PT\_IP\_V6',~~  ~~'PORT\_SAP\_ID\_PORT\_IDENTIFIER', 'PT\_SAP\_ID', 'PT\_UNDEFINED')~~ |
| ~~GDB.Log\_Port\_identifier\_value.value~~ | ~~Cust\_Edge\_IP\_Data2\_Simple\_Network\_Address~~  ~~CUST\_EDGE\_IP\_DATA2\_NET\_ADDRESS~~ |  |
| Nc3\_mat.cos\_profile.profile\_number  Icore.cos\_profile.profile\_id | Cust\_Edge\_Ingress\_Profile\_Id |  |
| Nc3\_mat.cos\_profile.name  Icore.cos\_profile.profile\_name | Cust\_Edge\_Ingress\_Profile\_Name |  |
| Nc3\_mat.cos\_profile.profile\_number  Icore.cos\_profile.profile\_id | Cust\_Edge\_Egress\_Profile\_Id |  |
| Nc3\_mat.cos\_profile.name  Icore.cos\_profile.profile\_name | Cust\_Edge\_Egress\_Profile\_Name |  |
| GDB.Asset.Id | Asset\_Cust\_NW\_POK | Please refer to the notes below. |
| GDB.Association\_type.type | Asset\_Cust\_NW\_Name | Please refer to the notes below. |
| GDB.Asset\_Type.Type | Asset\_Cust\_NW\_Asset\_Type | Please refer to the notes below. |
|  | Asset\_Cust\_NW\_Asset\_Role | DECODE (AST.IS\_ROLE\_PRIMARY, ''Y'', ''ART\_PRIMARY'') AS "cdm:assetRole", DECODE (AST.IS\_ROLE\_ALTERNATE\_PRIMARY, ''Y'', ''ART\_ALTERNATE\_PRIMARY'' ) AS "cdm:assetRole", DECODE (AST.IS\_ROLE\_BACKUP, ''Y'', ''ART\_BACKUP'') AS "cdm:assetRole" |
| GDB.Network\_type.type | NW\_Sub\_Type | where NT.ID = GDB.Asset\_Ext\_Network\_Connection.ID\_NETWORK\_SUB\_TYPE |
| GDB.Asset\_ext\_network\_connection.bfd\_interval  Icore.ipfr.bfd\_heartbeat\_interval | Bfd\_Interval |  |
| pvc.pvc\_lqos | Quality\_Of\_Service |  |
| GDB.Asset.Id | Asset\_Group\_POK | Please refer to the notes below. |
| GDB.Association\_type.type | Asset\_Group\_Name | Please refer to the notes below. |
| GDB.Association\_type.type | Asset\_Group\_Asset\_Type | Please refer to the notes below. |
|  | Asset\_Group\_Asset\_Role | DECODE (AST.IS\_ROLE\_PRIMARY, ''Y'', ''ART\_PRIMARY'') AS "cdm:assetRole", DECODE (AST.IS\_ROLE\_ALTERNATE\_PRIMARY, ''Y'', ''ART\_ALTERNATE\_PRIMARY'' ) AS "cdm:assetRole", DECODE (AST.IS\_ROLE\_BACKUP, ''Y'', ''ART\_BACKUP'') AS "cdm:assetRole" |
| GDB.Log\_port.id\_asset\_equipment | Managed\_Eqp\_POK |  |
| GDB.Equipment\_type.type\_name | Managed\_Eqp\_Name |  |
|  | Managed\_Eqp\_Asset\_Type | ‘EQUIPMENT’ (Please refer the notes below) |
|  | Managed\_Eqp\_Asset\_Role | DECODE (AST.IS\_ROLE\_PRIMARY, ''Y'', ''ART\_PRIMARY'') AS "cdm:assetRole", DECODE (AST.IS\_ROLE\_ALTERNATE\_PRIMARY, ''Y'', ''ART\_ALTERNATE\_PRIMARY'' ) AS "cdm:assetRole", DECODE (AST.IS\_ROLE\_BACKUP, ''Y'', ''ART\_BACKUP'') AS "cdm:assetRole" |
|  | Service\_Option\_Name | If bvoip is enabled (check if bvoip flag is Y from ICORE.IPFR.BVOIP as being done in Network\_Connection\_Data) , serv\_opt\_name = ‘**~~SO\_MANAGED\_INDICATOR~~**’  ‘SO\_BVOIP\_ENABLED’. Else, NULL |
|  | Service\_Option\_Display\_Name | If bvoip is enabled, serv\_opt\_display\_name = ‘**~~SO\_MANAGED\_INDICATOR~~**’  ‘SO\_BVOIP\_ENABLED’. Else, NULL |
| GDB.Network\_type.type | Network\_Type |  |
| ipfr.dhv\_indicator | Dhv\_Flag | Note: If Y -> set as true,  N, Null -> set as false |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |

Note:

Find the "assetGroup" or “CustomerNetwork” or “NetworkConnection” or “Access\_Circuit” or “Equipment” by using the following association:

i. ASSET (assetID) <-> (IS\_GROUP\_MEMBER\_OF/(NULL)) <> ASSET (assetGroup, status <> ‘DELETED’)

Note: If ID\_Object\_What, which is passed in input and is a group member as mentioned above, then ID\_Object\_To will be the POK. Or

If ID\_Object\_To, which is passed in input and is a group member as mentioned above, then ID\_Object\_What will be the POK.

ii. For each asset object "anAsset":

1. Create an instance of AssetObjectSummaryType

2. populate AssetObjectSummaryType.assetType with the type of "anAsset"

3. populate AssetObjectSummaryType.name as follows based on the asset type:

a. CUSTOMER\_NETWORK: name = networkIdValue + "/" + networkName + “/” + networkType (from GDB.ASSET\_EXT\_CUSTOMER\_NETWORK.NETWORK\_NAME and NETWORK\_ID)

b. NETWORK\_CONNECTION: name = connectionIdValue (from Asset\_Ext\_Network\_Connection)

c. ACCESS\_CIRCUIT, TRUNK\_CIRCUIT, TRUNK\_CHANNEL\_CIRCUIT: name = circuitIdValue (Asset\_Ext\_Access\_Circuit or Asset\_Ext\_Trunk\_Circuit or Asset\_Ext\_Trunk\_Channel\_Circuit)

d. EQUIPMENT: name = hostname + "/" + type + “/” + customerAssetAliasName (use a "?" character for any missing value) (<Alias-Correction> update to include ‘customerAssetAliasName’) (from Asset\_Ext\_Equipment)

e. ASSET\_GROUP: name = name + "/" + type (use a "?" character for any missing value)

f. SIMPLE\_ASSET: hostname + “/” + type (use a “?” character for any missing value)

g. TRUNK\_GROUP: name = centralOfficeCLLI + "/" + customerCLLI + "/" + baseTrafficNumber (use a "?" character for any missing value)

h. VIRTUAL\_NETWORK\_CONNECTION: name = vncName + “/” + type

4. populate the idObj with "anAsset" ID

**Managed Equipment:**

populate "managedEquipment" by using ID\_ASSET\_EQUIPMENT to create AssetObjectSummaryType instance as was done above.

* + - 1. Find the associated port ID for the asset by find ~~PHY\_PORT/~~LOG\_PORT where:

~~PHY\_PORT/~~LOG\_PORT(Type=’CE’) 🡪 (USED\_BY) 🡪 ASSET(status <> ‘DELETED’)

Always use the “unified” ~~PHY\_PORT/~~LOG\_PORT

If retrieved ASSET is of type ACCESS\_CIRCUIT, PRIVATE\_LINE\_CIRCUIT, TRUNK\_CIRCUIT or TRUNK\_CHANNEL\_CIRCUIT, get the CIRCUIT\_ID\_VALUE, and ID\_CIRCUIT\_ID\_IDENTIFIER\_TYPE from the appropriate table as listed below:

* 1. ASSET\_EXT\_ACCESS\_CIRCUIT
  2. ASSET\_EXT\_PL\_CIRCUIT
  3. ASSET\_EXT\_TRUNK\_CIRCUIT
  4. ASSET\_EXT\_TRUNK\_CHNL\_CIRCUIT

For each asset object "anAsset":

1. Create an instance of AssetObjectSummaryType

2. populate AssetObjectSummaryType.assetType with the type of "anAsset"

3. populate AssetObjectSummaryType.name as follows based on the asset type:

a. name = hostname + “/” + type + “/” + Alias\_Value (use a "?" character for any missing value) + “/” + DEVICE\_CATEGORY (use a "?" character for any missing value)

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| ASSET.ID | ASSET\_EXT\_NETWORK\_CONNECTION.ID\_ASSET |
| ICORE.PVC.PVC\_ID | ICORE.NETWORK\_CONNECTION.PVC\_ID |
| ICORE.NETWORK\_CONNECTION.PVC\_ID | GDB.ASSET\_EXT\_NETWORK\_CONNECTION.NETWORK\_CONNECTION\_ID |
| ICORE.PVC.PVC\_ID | ICORE.IPFR.PVC\_ID |
| GDB.Log\_Port.ID | GDB.Association.ID\_Object\_What |
| GDB.Association.ID\_Object\_To | GDB.Asset\_Ext\_Network\_connection.id\_asset |
| GDB.Asset\_Ext\_Network\_connection.id\_asset | GDB.Log\_Port.id\_log\_port\_unified |
| GDB.Log\_port.id | GDB.Log\_Port\_Identifier.Id\_Log\_Port |
| GDB.Log\_port.Id\_Port\_Type | GDB.Port\_Type.Id |
| GDB.Log\_Port\_Identifier.Id | GDB.Log\_Port\_Identifier\_Value.id\_log\_port\_identifier |
| NC3\_MAT.COS\_PROFILE.ODBID\_LOG\_PORT | NC3\_MAT.COS\_DATA.ODBID\_LOG\_PORT |
| NC3\_MAT.COS\_PROFILE.ODBID\_LOG\_PORT | NC3\_MAT.LOG\_PORT.ODBID |
| NC3\_MAT.IP\_INTERFACE.ODBID\_LOG\_PORT | NC3\_MAT.LOG\_PORT.ODBID |
| NC3\_MAT.LOG\_PORT.ODBID | GDB.LOG\_PORT.ID\_LOG\_PORT\_UNIFIED |
|  |  |

log\_port.id ( nc3 log\_port)= metadata.source\_key.id  
and metadata.source\_key.meta\_table = meta\_table.id  
and meta\_table.id\_meta\_system = meta\_system.id  
and meta\_table.name='LOG\_PORT'  
and meta\_system.name='LPP-CPE'

Port Level CoS details:  
1. When Port Level CoS is true, the CoS ingress and Egress profiles (allocations and advanced options) are all returned in the **AccessCircuitDetailInstance** in the API.   
2. When Port Level CoS is false, the CoS ingress and Egress profiles (allocations and advanced options) are all returned in the **NetworkConnectionDetailInstance** in the API.  
3. For Managed Assets, Port Level CoS is separately indicated for PE and CE – in which case, the CE CoS profile info is passed in CustomerPortDetails.  
4. Also for Managed Assets, PE Port Level CoS (PLC) and CE Port Level CoS (PLC) can have the below combinations possible for Fr Encap and Ethernet Port.  
i. Frame Encap –   
1. PE PLC = Y and CE PLC = Y (Both PE and CE CoS Profiles are at the Port Level)  
2. PE PLC = Y and CE PLC = N (PE CoS Profile is at the Port Level and CE CoS Profile is at the Channel Level)  
ii. Ethernet –   
1. PE PLC = Y and CE PLC = Y (Both PE and CE CoS Profiles are at the Port Level)  
---------------------------------------------------------------------------------------------------  
**Access Circuit: Customer Port-> find NC3 PHY\_PORT-> find COS details...**  
Find the CE side PHY\_PORT by checking whether an NC3 port record exists (that has been unified into the ICORE record) **using 'stopDate > sysdate' filter** :  
1. Derive this information by using the Master PHY\_PORT from the GDB schema and check all the associated child ports (unification slave ports) to determine which one was loaded from NC3.  
2. Once the PHY\_PORT loaded from NC3 is found, use it to go to GDB METADATA schema to find the ODBID (NC3 key)  
3. Once the NC3.port.ODBID value is obtained, use it to pull the corresponding NC3.PHY\_PORT   
USING IP\_INTERFACE.PORT\_LEVEL\_COS having portLevelCos == 'Y'  
Join into log\_port and ip\_interface; if at least one record has portLevelCos == 'Y' then pull the corresponding COS\_ PROFILE and COS\_DATA data. With port level cos, find the port record that has the COS\_PROFILE/COS\_DATA data associated to it to pull USING IP\_INTERFACE.PORT\_LEVEL\_INTERFACE == 'Y'. If no record with portLevelCos == 'Y' is found, this is logical connection level Cos so nothing needs to be provided.  
**Network Connection: Customer Port-> find NC3 LOG\_PORT-> find COS details...**  
Find the CE side LOG\_PORT by checking whether an NC3 port record exists (that has been unified into the ICORE record) **using 'stopDate > sysdate' filter:**  
1. Derive this information by using the Master LOG\_PORT from the GDB schema and check all the associated child ports (unification slave ports) to determine which one was loaded from NC3.  
2. Once the LOG\_PORT loaded from NC3 is found, go to GDB METADATA schema to find the ODBID (NC3 key)  
3. Once the NC3.log\_port.ODBID value is obtained, use it to pull the corresponding NC3.LOG\_PORT   
USING IP\_INTERFACE.PORT\_LEVEL\_COS having portLevelCos == 'Y'. If True, we are done as we need not provide Cos data, if False, pull the COS Profile data from the COS\_PROFILE table USING the log port ODBID

A one time full load of the GDB NETWORK\_IP\_DATA table for all Access Circuit POKs will be needed to initialize this table.

On subsequent loads, a new record should be inserted if a new Access Circuit POK was added to GDB NETWORK\_IP\_DATA.

|  |  |  |
| --- | --- | --- |
| **ICORE table.column** | **GDB NETWORK\_IP\_DATA column** | **Processing comment** |
| GDB.ASSET.ID | NETWORK\_CONNECTION\_POK | Where asset\_type=’NETWORK\_CONNECTION’. |
| Trim(GDB.identifier\_type.type) | Prov\_Edge\_IP\_Data\_Type | ('DLCI\_PORT\_IDENTIFIER', 'PT\_FR\_DLCI',  'VPI\_PORT\_IDENTIFIER', 'PT\_ATM\_VPI',  'VCI\_PORT\_IDENTIFIER', 'PT\_ATM\_VCI',  'UNDEFINED\_PORT\_IDENTIFIER', 'PT\_UNDEFINED',  'ETH\_TOP\_PORT\_IDENTIFIER', 'PT\_ETH\_TOP',  'ETH\_BOTTOM\_PORT\_IDENTIFIER', 'PT\_ETH\_BOTTOM',  'VLAN\_PORT\_IDENTIFIER', 'PT\_VLAN',  'INSTAR\_SDID\_PORT\_IDENTIFIER', 'PT\_SDID',    'IPV4', 'PT\_IP\_V4',  'IPV6', 'PT\_IP\_V6',  'PORT\_SAP\_ID\_PORT\_IDENTIFIER', 'PT\_SAP\_ID', 'PT\_UNDEFINED') |
| GDB.Log\_Port\_identifier\_value.value | PROV\_EDGE\_IP\_DATA\_NET\_ADDRESS | if 'IPV4', 'PT\_IP\_V4', 'IPV6', 'PT\_IP\_V6', do not load any data. |
| Trim(GDB.identifier\_type.type) | Cust\_Edge\_IP\_Data\_Type | ('DLCI\_PORT\_IDENTIFIER', 'PT\_FR\_DLCI',  'VPI\_PORT\_IDENTIFIER', 'PT\_ATM\_VPI',  'VCI\_PORT\_IDENTIFIER', 'PT\_ATM\_VCI',  'UNDEFINED\_PORT\_IDENTIFIER', 'PT\_UNDEFINED',  'ETH\_TOP\_PORT\_IDENTIFIER', 'PT\_ETH\_TOP',  'ETH\_BOTTOM\_PORT\_IDENTIFIER', 'PT\_ETH\_BOTTOM',  'VLAN\_PORT\_IDENTIFIER', 'PT\_VLAN',  'INSTAR\_SDID\_PORT\_IDENTIFIER', 'PT\_SDID',    'IPV4', 'PT\_IP\_V4',  'IPV6', 'PT\_IP\_V6',  'PORT\_SAP\_ID\_PORT\_IDENTIFIER', 'PT\_SAP\_ID', 'PT\_UNDEFINED') |
| GDB.Log\_Port\_identifier\_value.value | CUST\_EDGE\_IP\_DATA~~1~~\_NET\_ADDRESS |  |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |

A one time full load of the GDB IPV4\_DATA table for all Network Connection POKs will be needed to initialize this table.

On subsequent loads, a new record should be inserted if a new Network Connection POK was added to GDB IPV4\_DATA.

|  |  |  |
| --- | --- | --- |
| **ICORE table.column** | **GDB IPV4\_DATA column** | **Processing comment** |
| GDB.ASSET.ID | NETWORK\_CONNECTION\_POK | Where asset\_type=’NETWORK\_CONNECTION’,  Associated to access\_circuit POK related to AVPN service. |
| ~~GDB.Static\_route\_v4.ip\_address~~  ~~static\_routing.static\_route~~ | ~~IP\_Address\_V4~~ |  |
| ~~GDB.Static\_route\_v4.subnet\_mask~~  ~~static\_routing.route\_mask~~ | ~~subnet\_Mask\_V4~~ |  |
| bgp\_routing.asn  GDB.Log\_port.ipv4\_asn\_number | Asn\_Number |  |
| bgp\_routing.strip\_asn  GDB.Log\_port.ipv4\_asn\_override | Asn\_Override |  |
| bgp\_routing.restrictive\_route  GDB.Log\_port.ipv4\_restrictive\_routing | Restrictive\_Routing\_Fg | Note: If Y -> set as true,  N, Null -> set as false |
| ipfr.md5  GDB.Log\_port.ipv4\_md5\_encryption | md5\_Encryption\_Fg | Note: If Y -> set as true,  N, Null -> set as false |
| ipfr.v4\_mtu  GDB.Log\_port.ipv4\_mtu\_size | Mtu\_Size |  |
| ipfr.pmtu | Provider\_Mtu\_Size |  |
| GDB.Log\_port\_identifier\_value.value  routing\_type.routing\_type | Routing\_Protocol\_Ipv4\_Pe | identifier value 'IPV4\_ROUTING\_PROTOCOL\_PORT\_IDENTIFIER' |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| ASSET.ID | ASSET\_EXT\_NETWORK\_CONNECTION.ID\_ASSET |
| ASSET\_EXT\_NETWORK\_CONNECTION.ID\_ASSET | STATIC\_ROUTE\_V4.ID\_ASSET |
| ICORE.PVC.PVC\_ID | ICORE.NETWORK\_CONNECTION.PVC\_ID |
| ICORE.NETWORK\_CONNECTION.PVC\_ID | GDB.ASSET\_EXT\_NETWORK\_CONNECTION.NETWORK\_CONNECTION\_ID |
| ICORE.PVC.PVC\_ID | ICORE.IPFR.PVC\_ID |
| ICORE.IPFR.IPFR\_ID | ICORE.STATIC\_ROUTING.IPFR\_ID |
| ROUTING\_TYPE.ROUTING\_TYPE\_ID | ICORE.IPFR.ROUTING\_TYPE\_ID |
| ICORE.IPFR.IPFR\_ID | ICORE.BGP\_Routing.IPFR\_ID |
| GDB.Log\_Port.ID | GDB.Association.ID\_Object\_What |
| GDB.Association.ID\_Object\_To | GDB.Asset\_Ext\_Network\_connection.id\_asset |
| GDB.Log\_port.id | GDB.Log\_Port\_Identifier.Id\_Log\_Port |
| GDB.Log\_port.Id\_Port\_Type | GDB.Port\_Type.Id |
| GDB.Log\_Port\_Identifier.Id | GDB.Log\_Port\_Identifier\_Value.id\_log\_port\_identifier |

Note:

* 1. ~~Find the CE side LOG\_PORT by using:~~

~~LOG\_PORT.id\_asset\_equipment = ASSET.id (for EQUIPMENT). If LOG\_PORT.id\_log\_port\_unified is not NULL, use the id\_log\_port\_unified value~~

~~LOG\_PORT ('CE') identifier\_value of type IPV4\_IP\_ADDR\_PORT\_IDENTIFIER for identifier type of IPV4\_PORT\_IDENTIFIER~~

* 1. Find the PE side LOG\_PORT by using:

LOG\_PORT --> (USED\_BY) --> ASSET (type = 'NETWORK\_CONNECTION')

ASSET <-- (USED\_BY) <-- LOG\_PORT (port\_type = 'PE')

LOG\_PORT ('PE') identifier\_value of type IPV4\_PORT\_IDENTIFIER.

* 1. Please use port\_type of ‘PE’ as client is looking for PE related data.

A one time full load of the GDB IPV4\_ADDRESS\_DATA table for all Network Connection POKs will be needed to initialize this table.

On subsequent loads, a new record should be inserted if a new Network Connection POK was added to GDB IPV4\_ADDRESS\_DATA.

|  |  |  |
| --- | --- | --- |
| **ICORE table.column** | **GDB IPV4\_ADDRESS\_DATA column** | **Processing comment** |
| GDB.ASSET.ID | NETWORK\_CONNECTION\_POK | Where asset\_type=’NETWORK\_CONNECTION’, associated with the access\_circuit POK related to ‘AVPN’ service. |
|  | ID | Sequential Incrementor <Defect 388187> |
| Trim(GDB.identifier\_type.type) | Type | Valid Type: ('IPV4', 'PT\_IP\_V4') <Defect 388187> |
| GDB.Log\_Port\_identifier\_value.value  ~~GDB.Static\_route\_v4.ip\_address~~  static\_routing.static\_route | IP\_Address\_V4 | where GDB.identifier\_type is 'IPV4\_IP\_ADDR\_PORT\_IDENTIFIER' |
| GDB.Log\_Port\_identifier\_value.value  ~~GDB.Static\_route\_v4.subnet\_mask~~  static\_routing.route\_mask | subnet\_Mask\_V4 | Where GDB.Identifier\_Type is 'IPV4\_SUBNET\_MASK\_PORT\_IDENTIFIER' |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| ASSET.ID | ASSET\_EXT\_NETWORK\_CONNECTION.ID\_ASSET |
| ASSET\_EXT\_NETWORK\_CONNECTION.ID\_ASSET | STATIC\_ROUTE\_V6.ID\_ASSET |
| ICORE.PVC.PVC\_ID | ICORE.NETWORK\_CONNECTION.PVC\_ID |
| ICORE.IPFR.IPFR\_ID | ICORE.STATIC\_ROUTING.IPFR\_ID |
| GDB.Log\_Port.ID | GDB.Association.ID\_Object\_What |
| GDB.Association.ID\_Object\_To | GDB.Asset\_Ext\_Network\_connection.id\_asset |
| GDB.Log\_port.id | GDB.Log\_Port\_Identifier.Id\_Log\_Port |
| GDB.Log\_port.Id\_Port\_Type | GDB.Port\_Type.Id |
| GDB.Log\_Port\_Identifier.Id | GDB.Log\_Port\_Identifier\_Value.id\_log\_port\_identifier |

A one time full load of the GDB IPV6\_DATA table for all Network Connection POKs will be needed to initialize this table.

On subsequent loads, a new record should be inserted if a new Network Connection POK was added to GDB IPV6\_DATA.

|  |  |  |
| --- | --- | --- |
| **ICORE table.column** | **GDB IPV6\_DATA column** | **Processing comment** |
| GDB.ASSET.ID | NETWORK\_CONNECTION\_POK | Where asset\_type=’NETWORK\_CONNECTION’, associated with the access\_circuit POK related to ‘AVPN’ service. |
| ~~GDB.Static\_route\_v46.ip\_address~~  ~~instar.ipv6\_assign\_ip.ipv6\_ip\_compress~~ | ~~IP\_Address\_V6~~ |  |
| ~~GDB.Static\_route\_v46.subnet\_mask~~  ~~instar.ipv6\_assign\_ip.length~~ | ~~subnet\_Mask\_V6~~ |  |
| ~~GRID.SC\_STATIC\_ROUTING.ipv6\_prefix\_length~~ | ~~IP\_Prefix\_Length\_V6~~ |  |
| ipfr.v6\_wan\_ip  GDB.Log\_port.wan\_ipv6\_ip\_address | Wan\_IP\_Address |  |
| v6\_rtg\_protocol.asn  GDB.Log\_port.ipv6\_asn\_number | Asn\_Number |  |
| v6\_rtg\_protocol.asnoverride  GDB.Log\_port.ipv6\_asn\_override | Asn\_Override |  |
| v6\_rtg\_protocol.send\_default  GDB.Log\_port.ipv6\_restricitve\_routing | Restrictive\_Routing\_Flag | Note: If Y -> set as true,  N, Null -> set as false |
| v6\_rtg\_protocol.md5\_password  GDB.Log\_port.ipv6\_md5\_encryption | md5\_Encryption\_Flag | Note: If Y -> set as true,  N, Null -> set as false |
| ipfr.v6\_mtu  GDB.Log\_port.ipv6\_mtu\_size | Mtu\_Size |  |
| ipfr.pmtu | Provider\_Mtu\_Size |  |
| GDB.Log\_port\_identifier\_value.value  V6\_RTG\_PROTOCOL.PROTOCOL | Routing\_Protocol\_Ipv6\_Pe | identifier value 'IPV6\_ROUTING\_PROTOCOL\_PORT\_IDENTIFIER' |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| ASSET.ID | ASSET\_EXT\_NETWORK\_CONNECTION.ID\_ASSET |
| ASSET\_EXT\_NETWORK\_CONNECTION.ID\_ASSET | STATIC\_ROUTE\_V6.ID\_ASSET |
| ICORE.PVC.PVC\_ID | ICORE.NETWORK\_CONNECTION.PVC\_ID |
| ICORE.NETWORK\_CONNECTION.PVC\_ID | GDB.ASSET\_EXT\_NETWORK\_CONNECTION.NETWORK\_CONNECTION\_ID |
| ICORE.PVC.PVC\_ID | ICORE.IPFR.PVC\_ID |
| ICORE.IPFR.IPFR\_ID | ICORE.STATIC\_ROUTING.IPFR\_ID |
| V6\_RTG\_PROTOCOL.IPFR\_ID | ICORE.IPFR.IPFR\_ID |
| ICORE.IPFR.IPFR\_ID | ICORE.BGP\_Routing.IPFR\_ID |
| GDB.Log\_Port.ID | GDB.Association.ID\_Object\_What |
| GDB.Association.ID\_Object\_To | GDB.Asset\_Ext\_Network\_connection.id\_asset |
| GDB.Log\_port.id | GDB.Log\_Port\_Identifier.Id\_Log\_Port |
| GDB.Log\_port.Id\_Port\_Type | GDB.Port\_Type.Id |
| GDB.Log\_Port\_Identifier.Id | GDB.Log\_Port\_Identifier\_Value.id\_log\_port\_identifier |
| ~~Icore.map\_instar\_port.pvc\_id~~ | ~~Icore.pvc.pvc\_id~~ |
| Icore.map\_instar\_port.instar\_port\_asgmt | Instar.ip\_port\_asgmt.ip\_port\_asgmt\_id |
| ICORE.PVC.PVC\_ID | INSTAR.IP\_SERV\_ACCT\_PT.ICORE\_PVC\_ID |
| ~~Instar.ip\_port\_asgmt.sdid~~  Instar.IP\_SERV\_ACCT\_PT.sdid | instar.ipv6\_lan\_port\_map. Sdid |
| instar.ipv6\_assign\_ip. ipv6\_ip\_id | instar.ipv6\_lan\_port\_map. ipv6\_ip\_id |
| GDB.ASSET\_EXT\_NETWORK\_CONNECTION.ETE\_SERVICE\_CONNECTION\_KEY | GRID.SC\_STATIC\_ROUTING.ETE\_SERVICE\_CONNECTION\_KEY |

Note:

1. ~~Find the CE side LOG\_PORT by using:~~

~~LOG\_PORT.id\_asset\_equipment = ASSET.id (for EQUIPMENT). If LOG\_PORT.id\_log\_port\_unified is not NULL, use the id\_log\_port\_unified value~~

~~LOG\_PORT ('CE') identifier\_value of type IPV4\_IP\_ADDR\_PORT\_IDENTIFIER for identifier type of IPV4\_PORT\_IDENTIFIER~~

1. Find the PE side LOG\_PORT by using:

LOG\_PORT --> (USED\_BY) --> ASSET (type = 'NETWORK\_CONNECTION')

ASSET <-- (USED\_BY) <-- LOG\_PORT (port\_type = 'PE')

LOG\_PORT ('PE') identifier\_value of type ~~IPV6\_IP\_ADDR\_PORT\_IDENTIFIER~~ IPV6\_PORT\_IDENTIFIER

1. Please use port\_type of ‘PE’ as client is looking for PE related data.

A one time full load of the GDB IPV6\_ADDRESS\_DATA table for all Network Connection POKs will be needed to initialize this table.

On subsequent loads, a new record should be inserted if a new Network Connection POK was added to GDB IPV6\_ADDRESS\_DATA.

|  |  |  |
| --- | --- | --- |
| **ICORE table.column** | **GDB IPV6\_ADDRESS\_DATA column** | **Processing comment** |
| GDB.ASSET.ID | NETWORK\_CONNECTION\_POK | Where asset\_type=’NETWORK\_CONNECTION’, associated with the access\_circuit POK related to ‘AVPN’ service. |
|  | ID | Sequential Incrementor <Defect 388187> |
| Trim(GDB.identifier\_type.type) | Type | Valid Type: ('IPV6', 'PT\_IP\_V6') <Defect 388187> |
| GDB.Log\_Port\_identifier\_value.value  ~~GDB.Static\_route\_v46.ip\_address~~  instar.ipv6\_assign\_ip.ipv6\_ip\_compress | IP\_Address\_V6 | Where GDB.Identifier\_Type is 'IPV6\_IP\_ADDR\_PORT\_IDENTIFIER' |
| GDB.Log\_Port\_identifier\_value.value  ~~GDB.Static\_route\_v46.subnet\_mask~~  instar.ipv6\_assign\_ip.length | subnet\_Mask\_V6 | Where GDB.Identifier\_Type is 'IPV6\_SUBNET\_MASK\_PORT\_IDENTIFIER' |
| GRID.SC\_STATIC\_ROUTING.ipv6\_prefix\_length | IP\_Prefix\_Length\_V6 |  |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| ASSET.ID | ASSET\_EXT\_NETWORK\_CONNECTION.ID\_ASSET |
| ASSET\_EXT\_NETWORK\_CONNECTION.ID\_ASSET | STATIC\_ROUTE\_V6.ID\_ASSET |
| ICORE.PVC.PVC\_ID | ICORE.NETWORK\_CONNECTION.PVC\_ID |
| ICORE.PVC.PVC\_ID | INSTAR.IP\_SERV\_ACCT\_PT.ICORE\_PVC\_ID |
| Instar.IP\_SERV\_ACCT\_PT.sdid | instar.ipv6\_lan\_port\_map. Sdid |
| instar.ipv6\_assign\_ip. ipv6\_ip\_id | instar.ipv6\_lan\_port\_map. ipv6\_ip\_id |
| GDB.ASSET\_EXT\_NETWORK\_CONNECTION.ETE\_SERVICE\_CONNECTION\_KEY | GRID.SC\_STATIC\_ROUTING.ETE\_SERVICE\_CONNECTION\_KEY |
| GDB.Log\_Port.ID | GDB.Association.ID\_Object\_What |
| GDB.Association.ID\_Object\_To | GDB.Asset\_Ext\_Network\_connection.id\_asset |
| GDB.Log\_port.id | GDB.Log\_Port\_Identifier.Id\_Log\_Port |
| GDB.Log\_port.Id\_Port\_Type | GDB.Port\_Type.Id |
| GDB.Log\_Port\_Identifier.Id | GDB.Log\_Port\_Identifier\_Value.id\_log\_port\_identifier |

A one time full load of the GDB RELATED\_ASSET\_DATA table for all Network Connection POKs will be needed to initialize this table.

On subsequent loads, a new record should be inserted if a new Network Connection POK was added to GDB RELATED\_ASSET\_DATA.

|  |  |  |
| --- | --- | --- |
| **ICORE table.column** | **GDB RELATED\_ASSET\_DATA column** | **Processing comment** |
| GDB.ASSET.ID | NETWORK\_CONNECTION\_POK | Where asset\_type=’NETWORK\_CONNECTION’ |
| GDB.Asset.Id | Related\_Asset\_Primary\_Obj\_Key | Retrieved from the Asset <-> Asset association. |
| GDB.Association\_Type.Type | Related\_Asset\_Name | ('ACCESS\_CIRCUIT', (SELECT asec.circuit\_id\_value  FROM asset\_ext\_access\_circuit asec  WHERE asec.id\_asset =  astid)) |
| GDB.Association\_Type.Type | Asset\_Type | Please refer to notes given below. |
|  | Asset\_Role | DECODE (AST.IS\_ROLE\_PRIMARY, ''Y'', ''ART\_PRIMARY'') AS "cdm:assetRole", DECODE (AST.IS\_ROLE\_ALTERNATE\_PRIMARY, ''Y'', ''ART\_ALTERNATE\_PRIMARY'' ) AS "cdm:assetRole", DECODE (AST.IS\_ROLE\_BACKUP, ''Y'', ''ART\_BACKUP'') AS "cdm:assetRole" |
| GDB.Function\_Type.Name | Relation\_Type | ('IS\_BACKUP\_FOR', 'IS\_BACKUP\_FOR', ‘IS\_NETWORK\_CONNECTION\_FOR\_ANIRA\_PRESENCE', 'IS\_NETWORK\_CONNECTION\_FOR\_ANIRA\_PRESENCE',  'IS\_NETWORK\_CONNECTION\_FOR\_DEVICE', 'IS\_NETWORK\_CONNECTION\_FOR\_DEVICE', 'IS\_VLAN\_FOR\_VNC','IS\_VNC\_FOR\_VLAN', 'IS\_NETWORK\_CONNECTION\_FOR\_BVOIP\_PRESENCE', 'IS\_NETWORK\_CONNECTION\_FOR\_BVOIP\_PRESENCE')  Please refer the notes given below. |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |

Notes:

**~~Equipment:~~**

~~populate relatedAsset by retrieving any non-equipment asset that is 'associated with' this asset.~~

~~Note that the maxOccurs for this is increased to 200 (from previous 100) – so upto 200 related assets should be sent back in response:~~

~~a. ASSET <-> ASSET (relatedAsset, STATUS <> ‘DELETED’, assetType <> ‘EQUIPMENT’)~~

~~b. Create an instance of RelatedAssetObjectSummaryType. Populate the attributes using AssetObjectSummaryType steps as was done above. Populate the "relationType" attribute using the AssetRelationshipValidValue where the value matches GDB.FUNCTION\_TYPE.NAME - use the following joins to find FUNCTION\_TYPE.NAME:~~

~~FUNCTION\_TYPE.ID == ROLE.ID\_FUNCTION\_TYPE~~

~~ROLE.ID == ASSOCIATION\_TYPE.ID\_ROLE~~

~~ASSOCIATION\_TYPE.ID = ASSOCIATION.ID\_ASSOCIATION\_TYPE (for the ASSOCIATION record found above)~~

~~If GDB.FUNCTION\_TYPE.NAME has the value:~~

~~“IS\_ACCESS\_ROUTER\_FOR\_TRINITY\_PRESENCE”,~~

~~then for input 'assetID' matching 'id\_object\_what', the relationType of RelatedAssetObjectSummaryType should be set to IS\_TRINITY\_PRESENCE\_FOR\_ASSET, RelatedAssetObjectSummaryType’s idObj should be set to ‘id\_object\_to’.~~

~~If GDB.FUNCTION\_TYPE.NAME has the following values:~~

~~“IS\_ACCESS\_CIRCUIT\_FOR\_UCPE”, “IS\_ACCESS\_CIRCUIT\_PRIMARY\_FOR\_UCPE”, or “IS\_ACCESS\_CIRCUIT\_BACKUP\_FOR\_UCPE”~~

~~Populate relationType to FUNCTION\_TYPE.NAME, and RelatedAssetObjectSummaryType’s idObj should be set to ‘id\_object\_what’.~~

~~Include FUNCTION\_TYPE of ‘IS\_BVOIP\_PRESENCE\_FOR\_ASSET’ function\_type~~

~~Include FUNCTION\_TYPE of “IS\_WAN2\_INTERNET\_ACCESS\_CIRCUIT\_FOR\_UCPE”~~

~~Include FUNCTION\_TYPE of “IS\_DEVICE\_FOR\_ANIRA\_PRESENCE” (id\_object\_to), “IS\_NETWORK\_CONNECTION\_FOR\_DEVICE” (id\_object\_what) and “IS\_ACCESS\_CIRCUIT\_FOR\_DEVICE” (id\_object\_what)~~

~~Include FUNCTION\_TYPE of “IS\_VQM\_DEVICE\_FOR\_BVOIP\_PRESENCE” (input ‘assetID’ matching ‘id\_object\_what’). If found, set the RelatedAsset.relationType for those as ‘IS\_BVOIP\_PRESENCE\_FOR\_ASSET’~~

**Network Connection:**

populate relatedAsset by retrieving any non-access circuit asset that is 'associated with' this asset:

a. ASSET <- ->ASSET (relatedAsset, STATUS <> ‘DELETED’)

b. Create an instance of RelatedAssetObjectSummaryType. Populate the attributes using AssetObjectSummaryType steps as was done above. Populate the "relationType" attribute using the AssetRelationshipValidValue where the value matches GDB.FUNCTION\_TYPE.NAME - use the following joins to find FUNCTION\_TYPE.NAME:

c. FUNCTION\_TYPE.ID == ROLE.ID\_FUNCTION\_TYPE

d. ROLE.ID == ASSOCIATION\_TYPE.ID\_ROLE

e. ASSOCIATION\_TYPE.ID = ASSOCIATION.ID\_ASSOCIATION\_TYPE (for the ASSOCIATION record found above)

f. Don’t include related network connection, access circuit, and customer network.

g. Include the FUNCTION\_TYPE of “IS\_NETWORK\_CONNECTION\_FOR\_ANIRA\_PRESENCE”, “IS\_NETWORK\_CONNECTION\_FOR\_DEVICE”, and “IS\_NETWORK\_CONNECTION\_FOR\_BVOIP\_PRESENCE”.

h. If GDB.FUNCTION\_TYPE is “IS\_BACKUP\_FOR” and current network connection’s platformObjectKey equals id\_object to, take id\_object\_what (function\_type is “IS\_BACKUP\_FOR”), if current network connection’s platformObjectKey equals id\_object\_what, take the id\_object\_to (function\_type is “IS\_PRIMARY\_FOR”).

i. Include the FUNCTION\_TYPE of “IS\_VLAN\_FOR\_VNC” when searching. If any found, populate RelatedAsset.relationType with “IS\_VNC\_FOR\_VLAN” (notice the reversing of the name)

**Access Circuit:**

populate relatedAsset by retrieving any non-access circuit asset that is 'associated with' this asset:

a. ASSET <- -> ASSET (relatedAsset, STATUS <> ‘DELETED’)

b. Create an instance of RelatedAssetObjectSummaryType. Populate the attributes using AssetObjectSummaryType steps as was done above. Populate the "relationType" attribute using the AssetRelationshipValidValue where the value matches GDB.FUNCTION\_TYPE.NAME - use the following joins to find FUNCTION\_TYPE.NAME:

c. FUNCTION\_TYPE.ID == ROLE.ID\_FUNCTION\_TYPE

d. ROLE.ID == ASSOCIATION\_TYPE.ID\_ROLE

e. ASSOCIATION\_TYPE.ID = ASSOCIATION.ID\_ASSOCIATION\_TYPE (for the ASSOCIATION record found above)

f. If found a IS\_ACCESS\_CIRCUIT\_FOR\_TRINITY\_PRESENCE, then for input 'assetID' matching 'id\_object\_what', the relationType should be set to IS\_TRINITY\_PRESENCE\_FOR\_ASSET, RelatedAssetObjectSummaryType’s idObj should be set to ‘id\_object\_to’.

g. If GDB.FUNCTION\_TYPE.NAME has the following values:

“IS\_ACCESS\_CIRCUIT\_FOR\_UCPE”, “IS\_ACCESS\_CIRCUIT\_PRIMARY\_FOR\_UCPE”, or “IS\_ACCESS\_CIRCUIT\_BACKUP\_FOR\_UCPE”, <270198g> “IS\_ACCESS\_CIRCUIT\_FOR\_DEVICE”

Populate relationType to FUNCTION\_TYPE.NAME, and RelatedAssetObjectSummaryType’s idObj should be set to ‘id\_object\_to’.

h. If GDB.FUNCTION\_TYPE.NAME has the following values:

“IS\_ACCESS\_CIRCUIT\_FOR\_UCPE”, “IS\_ACCESS\_CIRCUIT\_PRIMARY\_FOR\_UCPE”, or “IS\_ACCESS\_CIRCUIT\_BACKUP\_FOR\_UCPE”, or

“IS\_WAN2\_INTERNET\_ACCESS\_CIRCUIT\_FOR\_UCPE”

Populate relationType as “IS\_UCPE\_FOR\_ACCESS\_CIRCUIT”, and RelatedAssetObjectSummaryType’s idObj should be set to ‘id\_object\_to’.

Pull data for “IS\_UCPE\_FOR\_ACCESS\_CIRCUIT”

i. If GDB.FUNCTION\_TYPE.NAME has the following values:

“IS\_RELATED\_CSU\_DSU”

Populate relationType to ‘IS\_CSU\_DSU\_FOR\_ACCESS\_CIRCUIT’, and RelatedAssetObjectSummaryType’s idObj should be set to ‘id\_object\_to’.

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| ASSET.ID <Access\_Circuit> | ASSOCIATION.ID\_OBJECT\_WHAT |
| ASSOCIATION.ID\_Object\_To | ASSET\_ID <Network\_Connection> |
| ASSOCIATION.Id\_Association\_Type | AsseT\_Type.Id |
| Asset\_type.id\_role | Role.id |

A one time full load of the GDB AVPN\_ACCESS\_CIRCUIT\_DATA table for all Access Circuit POKs will be needed to initialize this table.

On subsequent loads, a new record should be inserted if a new Access Circuit POK was added to GDB AVPN\_ACCESS\_CIRCUIT\_DATA.

|  |  |  |
| --- | --- | --- |
| **ICORE table.column** | **GDB AVPN\_ACCESS\_CIRCUIT\_DATA column** | **Processing comment** |
| GDB.ASSET.ID | ACCESS\_CIRCUIT\_POK | Where asset\_type.type=’ACCESS\_CIRCUIT’, and id\_status is other than ‘DELETED’ |
| GDB.Address\_Notation.Subdivision | State |  |
| GDB.asset\_ext\_access\_circuit.circuit\_id\_value  (1) Use “CUST\_ACCESS.acc\_ckt” if no CLCI-formatted value could be converted; (2) Use the CLCI-formatted value of “CUST\_ACCESS.acc\_ckt” if a CLCI-formatted value could be converted; | Circuit\_Id |  |
| GDB.Asset\_Alias.Value | Customer\_Asset\_Alias\_Name |  |
| GDB.Asset\_Ext\_Access\_Circuit.speed | Speed |  |
| GDB.Phy\_port.speed | Prov\_Port\_Speed |  |
| GDB.Phy\_port.protocol | Prov\_port\_protocol |  |
| IPFR.V4\_MTU | Prov\_Mtu\_Size\_V4 |  |
| IPFR.V6\_MTU | Prov\_Mtu\_Size\_V6 |  |
| GDB.Phy\_port.port\_level\_cos\_ind | Prov\_port\_Level\_Cos\_Exists\_Flag |  |
| GDB. Asset\_ext\_access\_circuit.subrate\_ethernet  icore.service\_option | Prov\_port\_subRate\_Ethernet\_Flag | service\_option=true when service\_option.serv\_opt\_long=’SUBRATE ETH’ |
| GDB.Asset\_ext\_access\_circuit.is\_vlan  site.vlan\_asr | Prov\_port\_NW\_Vlan\_Stacking\_Flag | If source is ICORE, then decode as follows: PORT -> true, else, false.  In GDB, if Y -> true, else false. |
| site.vlan\_tag\_control | Prov\_port\_vlan\_Tag\_Control |  |
| Usrp.logical\_port.cpe\_interface\_type | Prov\_port\_physical\_Interface |  |
| site.ingress\_profile\_id | Prov\_port\_ingress\_profile\_Id |  |
|  | Prov\_port\_ingress\_profile\_Name | Note: If values are not available, then dev team has to embed it with space. |
| site.egress\_profile\_id | Prov\_port\_egress\_profile\_Id |  |
|  | Prov\_port\_egress\_profile\_Name | Note: If values are not available, then dev team has to embed it with space. |
|  | ~~Prov\_port\_cos\_Detail~~ | ~~<Need to add the details in a new table>~~ |
| GDB.Phy\_port.speed | Cust\_Port\_Speed |  |
| GDB.Phy\_port.protocol | Cust\_Port\_protocol |  |
| GDB.Phy\_port.icore\_site\_id\_value | Cust\_Port\_inv\_Site\_Id |  |
| Nc3\_mat.cos\_profile.profile\_number | Cust\_Port\_ingress\_profile\_Id |  |
| Nc3\_mat.cos\_profile.name | Cust\_Port\_ingress\_profile\_Name |  |
| Nc3\_mat.cos\_profile.profile\_number | Cust\_Port\_egress\_profile\_Id |  |
| Nc3\_mat.cos\_profile.name | Cust\_Port\_egress\_profile\_Name |  |
| Usrp.ckt\_loc.how\_access\_provided | Provisioning\_Access\_Arngmt\_Cd |  |
| Grid.access\_customer\_prem\_endpoint.network\_interface\_jack | Cust\_Port\_NW\_Interface\_Jack |  |
|  | ~~Available\_Bandwidth~~ | ~~<Not available>~~ |
| Icore.service\_option.serv\_opt\_long | Cos\_Package | Where TRIM (SERVICE.SERV\_NAME) = 'COS PACKAGE'  Note: Remove all underscores and load. |
| GDB.Service\_Opt\_Name | Service\_Option\_Name | Service option SO\_BVOIP\_ENABLED is to be supported. |
| GDB.Service\_Opt\_Display\_Name | Service\_Option\_Display\_Name |  |
| GDB.Service\_Opt\_Name | Management\_Option\_Name | If SO\_MANAGED\_INDICATOR is True, then Managed, otherwise Transport |
| GDB.Service\_Opt\_Display\_Name | Management\_Option\_Display\_Name |  |
| Usrp.ckt\_loc.access\_type | Access\_Type |  |
| access\_method.access\_method\_desc  Usrp.logical\_port.access\_method\_type | Access\_Method\_Type |  |
| GDB.Asset\_ext\_access\_circuit.customer\_mtu\_size\_v4  site.v4\_mtu  Usrp.logical\_port.cmtu | Cust\_MTU\_Size\_V4 |  |
| GDB.Asset\_ext\_access\_circuit.customer\_mtu\_size\_v6  site.v6\_mtu | Cust\_MTU\_Size\_V6 |  |
| GDB.Asset\_ext\_access\_circuit.billing\_option  Usrp.logical\_port.ubb | Billing\_Option |  |
| GDB.Asset\_ext\_access\_circuit.minimum\_bandwidth\_commitment  Icore.service\_asgmt.ubb\_mbc | Min\_Bandwidth\_Commitment |  |
| GDB.Asset\_ext\_access\_circuit.interconnect\_type  intercon\_tech.intercon\_techname  Usrp.logical\_port.interconnect\_type | Access\_Inter\_connect |  |
| site.vendor\_name | Vendor |  |
| Cust\_access ~~site~~.vendor\_cktname  Grid.circuit.vendor\_clci | Vendor\_Circuit\_Id\_Value |  |
| ipfr.ip\_version | IP\_Version |  |
| cust\_access.turnup\_protocol | Turn\_up\_Protocol |  |
| site.site\_glbl\_dlci  Usrp.logical\_port.dlci | Dlci |  |
| Organization\_identifier\_Value.value | Mcn | where identifier\_type=’MCN’ |
| Organization\_identifier\_Value.value | Grc | where identifier\_type=’GRC’ |
| Organization\_identifier\_Value.value | Soc | where identifier\_type=’SOC’ |
| usrp.CKT\_LOC.cust\_spec\_access\_vend  Grid.access\_circuit.access\_provider\_code | Access\_Prov\_Cd |  |
| Usrp.logical\_ip.access\_method\_type  Grid.access\_customer\_prem\_endpoint.tail\_technology | Ethernet\_Type | where access\_method\_type in (6, 7, 8, 11, 13)  Decode 6,7,11 as Dedicated and 8,13 as Switched.  Grid.access\_customer\_prem\_endpoint.tail\_technology where tail\_technology in (ZERO\_MAIL\_ACCESS – Dedicated, ESP\_ETHERNET\_DEDICATED – Dedicated, ZERO\_MILE\_ACCESS\_SHARED – Dedicated, ESP\_ETHERNET\_SHARED – Switched, SEAMLESS\_CONTROL\_PANEL – Switched)  Use source\_key.id |
| cust\_access.itu\_carrier\_code | Itu\_Carrier\_Cd | TRIM(Substr(Itu\_Carrier\_Code,Instr(Itu\_Carrier\_Code,'-',1)+1)) |
| ~~GCP\_CIRCUIT.AAFDAROLE~~  Icore.service\_option.serv\_opt  GRID.PORT.SERVICE\_DIVERSITY\_OPTION | Diversity\_Option | ~~Derive the value as follows:~~  ~~AAF – ‘Advanced Access \Failover’~~  ~~DA – ‘Diverse Access’~~  ~~Null – ‘None’~~  SELECT so.serv\_opt from icore.service\_asgmt sa,icore.service\_option so, icore.service s where so.serv\_opt\_id=sa.serv\_opt\_id and so.serv\_id=s.serv\_id and trim(serv\_name)='SDO' and site\_id=<icoreSiteId>  'Switch' if serv\_opt = 'S', 'POP' if serv\_opt='P' |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| ASSET.ID | ASSET\_EXT\_ACCESS\_CIRCUIT.ID\_ASSET |
| CUST\_ACCESS.site\_id | PORT\_ASGMT.site\_id |
| ICORE.PVC.PVC\_ID | ICORE.NETWORK\_CONNECTION.PVC\_ID |
| ICORE.CUST\_ACCESS.site\_id | SITE.site\_id |
| ~~CUST\_ACCESS\_CKT.SITE\_ID~~  ICORE.CUST\_ACCESS.SITE\_ID | PVC.PVC\_LSITE\_ID or PVC.RSITE\_ID |
| ICORE.PVC.PVC\_ID | ICORE.IPFR.PVC\_ID |
| ICORE.IPFR.IPFR\_ID | ICORE.STATIC\_ROUTING.IPFR\_ID |
| V6\_RTG\_PROTOCOL.IPFR\_ID | ICORE.IPFR.IPFR\_ID |
| ICORE.IPFR.IPFR\_ID | ICORE.BGP\_Routing.IPFR\_ID |
| GDB.Log\_Port.ID | GDB.Association.ID\_Object\_What |
| GDB.Association.ID\_Object\_To | GDB.Asset\_Ext\_Network\_connection.id\_asset |
| GDB.Log\_port.id | GDB.Log\_Port\_Identifier.Id\_Log\_Port |
| GDB.Log\_port.Id\_Port\_Type | GDB.Port\_Type.Id |
| GDB.Log\_Port\_Identifier.Id | GDB.Log\_Port\_Identifier\_Value.id\_log\_port\_identifier |
| GDB.ASSET.ID | GDB.Association.id\_object\_what |
| GDB.Site.Id | GDB.Association.Id\_Object\_to |
| GDB.Site.ID\_Address\_Notation | GDB.Address\_Notation.Id |
| GDB.Phy\_Port. ID\_PORT\_TYPE | GDB.Port\_Type.Id |
| GDB. Phy\_Port. ID | GDB.Association.id\_object\_what |
| GDB.ASSET\_EXT\_ACCESS\_CIRCUIT.ID\_ASSET | GDB. Association.id\_object\_To |
| GDB.ASSET\_EXT\_ACCESS\_CIRCUIT.ETE\_ACCESS\_KEY | ~~GRID.ACCESS\_CUSTOMER\_PREM\_ENDPOINT.ETE\_ACCESS\_KEY~~  GRID.ACCESS\_CIRCUIT.ETE\_ACCESS\_KEY |
| USRP.LOGICAL\_PORT.LOG\_PORT\_ID | USRP.PORT\_INFO.BUF\_LOG\_PORT\_ID |
| Icore.cust\_access.site\_id | Usrp.port\_info.icore\_site\_id |
| Usrp.port\_info.ckt\_id | Usrp.circuit.circuit\_id |
| Usrp.circuit.circuit\_id | Usrp.ckt\_loc.circuit\_id |
| Usrp.logical\_ip.log\_ip\_id | Usrp.connection\_ip\_ckt.log\_ip\_id |
| USPR.PORT\_INFO.ICORE\_SITE\_ID | ICORE.SITE.SITE\_ID |
| ICORE.SITE.SITE\_ID | ICORE.cust\_access.SITE\_ID |
| ICORE.SITE.ACCESS\_METHOD\_ID | ICORE. ACCESS\_METHOD.ACCESS\_METHOD\_ID |
| Icore. cust\_access.site\_id | Icore. SERVICE\_ASGMT.SITE\_ID |
| GDB.ASSET\_EXT\_ACCESS\_CIRCUIT.ETE\_ACCESS\_KEY | GRID.CIRCUIT.ETE\_ACCESS\_KEY |
| ICORE.it.intercon\_tech\_id | ICORE.site.intercon\_tech\_id |
| ICORE.site.site\_id | ICORE.cust\_access.site\_id |
| ICORE.SERVICE\_OPTION.SERV\_OPT\_ID | ICORE.SERVICE\_ASGMT.SERV\_OPT\_ID |
| GDB.ASSET\_EXT\_ACCESS\_CIRCUIT.id\_asset | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id\_meta\_table | METADATA.META\_TABLE.id |
| METADATA.META\_SYSTEM.id | METADATA.META\_TABLE.id\_meta\_system |

Note: Find the CE side PHY\_PORT by checking whether an NC3 port record exists (that has been unified into the ICORE record) using 'stopDate > sysdate' filter :  
1. Derive this information by using the Master PHY\_PORT from the GDB schema and check all the associated child ports (unification slave ports) to determine which one was loaded from NC3.  
2. Once the PHY\_PORT loaded from NC3 is found, use it to go to GDB METADATA schema to find the ODBID (NC3 key)  
3. Once the NC3.port.ODBID value is obtained, use it to pull the corresponding NC3.PHY\_PORT  
USING IP\_INTERFACE.PORT\_LEVEL\_COS having portLevelCos == 'Y'  
Join into log\_port and ip\_interface; if at least one record has portLevelCos == 'Y' then pull the corresponding COS\_ PROFILE and COS\_DATA data. With port level cos, find the port record that has the COS\_PROFILE/COS\_DATA data associated to it to pull USING IP\_INTERFACE.PORT\_LEVEL\_INTERFACE == 'Y'. If no record with portLevelCos == 'Y' is found, this is logical connection level Cos so nothing needs to be provided.

~~A one time full load of the GDB EQUIPMENT\_DATA table for all Access Circuit POKs will be needed to initialize this table.~~

~~On subsequent loads, a new record should be inserted if a new Access Circuit POK was added to GDB EQUIPMENT\_DATA.~~

|  |  |  |
| --- | --- | --- |
| **~~ICORE table.column~~** | **~~GDB EQUIPMENT\_DATA column~~** | **~~Processing comment~~** |
| ~~GDB.ASSET.ID~~ | ~~ACCESS\_CIRCUIT\_POK~~ | ~~Where asset\_type.type=’ACCESS\_CIRCUIT’~~ |
| ~~INSTAR\_AAI\_DEVICE\_VW.equipment\_name for uCPE~~ | ~~Router\_host\_name~~ |  |
| ~~INSTAR\_AAI\_DEVICE.PTNII\_Name~~ | ~~vHNF\_Host\_name~~ |  |
| ~~GDB.Asset\_Ext\_Equipment.Customer\_Internal\_Alias~~ | ~~vHNF\_Customer\_Asset\_Alias~~ |  |
| ~~AAI\_PHYSICAL\_LINK.CIRCUIT\_ID~~ | ~~Connected\_Circuit\_Id\_Value~~ |  |
| ~~AAI\_PHYSICAL\_LINK.SERVICE\_PROVIDER\_NAME~~ | ~~Isp\_Name~~ |  |
| ~~AAI\_PHYSICAL\_LINK.svc\_provider\_bandwidth\_up\_val + AAI\_PHYSICAL\_LINK.svc\_provider\_bandwidth\_up\_unit~~ | ~~Isp\_Speed\_Up~~ |  |
| ~~AAI\_PHYSICAL\_LINK.svc\_provider\_bandwidth\_down\_val + AAI\_PHYSICAL\_LINK.svc\_provider\_bandwidth\_down\_unit~~ | ~~Isp\_Speed\_Down~~ |  |
| ~~AAI\_TUNNEL\_XCONNECT.BANDWIDTH\_UP\_WAN1 + AAI\_TUNNEL\_XCONNECT.BANDWIDTH\_DOWN\_WAN1~~  ~~or~~  ~~AAI\_TUNNEL\_XCONNECT.BANDWIDTH\_UP\_WAN2 + AAI\_TUNNEL\_XCONNECT.BANDWIDTH\_DOWN\_WAN2~~ | ~~Sd\_WAN\_Throughput~~ |  |
| ~~GPS.SW\_PROD\_RELEASE~~ | ~~Vhnf\_Part\_Number~~ | ~~select SPR.swpartnumber vNF\_part\_Number,~~  ~~sip.atspecassetname vnf\_hostname~~  ~~from GPS.SW\_PROD\_RELEASE SPR,~~  ~~GPS.SW\_CUSTOMER SC,~~  ~~GPS.SW\_INST\_PRODUCT SIP~~  ~~WHERE SIP.SWPRODRELEASEID = SPR.SWPRODRELEASEID~~  ~~AND SPR.SWCUSTOMERID = SC.SWCUSTOMERID~~ |
| ~~AAI\_Service\_Subscription.temp\_ub\_sub\_account~~ | ~~Ub\_Sub\_Account\_Id~~ |  |
| ~~GCP\_NODE.name~~ | ~~uCPE\_Host\_name~~ |  |
| ~~GDB.alias\_value.value~~ | ~~uCPE\_Customer\_Asset\_Alias~~ |  |
| ~~GCP\_NODE.ubsubaccount~~ | ~~Ub\_Sub\_Account\_Id\_UCPE~~ |  |
| ~~IDIS. GCP\_NODE. MAINTENANCEPROVIDER~~ | ~~Management\_Option~~ |  |
| ~~-~~ | ~~id\_change\_tracking~~ | ~~See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process]~~ |

~~Database traversals:~~

|  |  |
| --- | --- |
| **~~From table.column~~** | **~~To table.column~~** |
| ~~ASSET.ID~~ | ~~ASSET\_EXT\_EQUIPMENT.ID\_ASSET~~ |
| ~~ASSET\_EXT\_EQUIPMENT.HOST\_NAME~~ | ~~INSTAR\_AAI\_DEVICE\_VW.PTNII\_NAME~~ |
| ~~INSTAR\_AAI\_DEVICE\_VW.PTNII\_NAME~~ | ~~AAI\_GENERIC\_VNF.VNF\_NAME~~ |
| ~~AAI\_GENERIC\_VNF.VNF\_ID~~ | ~~AAI\_L\_INTERFACE.VNF\_ID~~ |
| ~~AAI\_L\_INTERFACE.INTERFACE\_NAME~~ | ~~AAI\_VLAN.INTERFACE\_NAME~~ |
| ~~AAI\_VLAN.VLAN\_INTERFACE~~ | ~~AAI\_MULTICAST\_CONFIGURATION.VLAN\_INTERFACE~~ |
| ~~ASSET\_EXT\_EQUIPMENT.HOST\_NAME~~ | ~~IDIS.GCP\_NODE.NAME~~ |
| ~~AAI\_L\_INTERFACE.HOST\_NAME~~ | ~~AAI\_P\_INTERFACE.HOST\_NAME~~ |
| ~~AAI\_P\_INTERFACE.HOST\_NAME~~ | ~~AAI\_P\_INTERFACE\_PHYSICAL\_LINK.HOST\_NAME~~ |
| ~~AAI\_P\_INTERFACE\_PHYSICAL\_LINK.LINK\_NAME~~ | ~~AAI\_PHYSICAL\_LINK.LINK\_NAME~~ |
| ~~instar\_aai\_device\_vw.ptnii\_name~~ | ~~gps.atspecassetname~~ |

A one time full load of the GDB COS\_DETAIL\_INGRESS\_DATA table for all Access Circuit POKs will be needed to initialize this table.

On subsequent loads, a new record should be inserted if a new Access Circuit POK was added to GDB COS\_DETAIL\_INGRESS\_DATA.

|  |  |  |
| --- | --- | --- |
| **ICORE table.column** | **GDB COS\_DETAIL\_INGRESS\_DATA column** | **Processing comment** |
| ~~Nc3\_mat.cos\_profile.profile\_number~~ or  cos\_profile.profile\_id, for Network Connection  GRID.SERVICE\_CONNECTION.PE\_INGRESS\_PROFILE\_ID, for Network Connection  site.ingress\_profile\_id, for Access Circuit | Ingress\_profile\_Id |  |
| Cos\_classes.class\_name | Ingress\_COS\_Class | Present in 'COS1','COS2','COS3','COS4', ‘COS2V’, ‘COS5’ |
| Cos\_allocations.class\_allocation | Ingress\_Value |  |
| Cos\_Option\_Values.OPTION\_VALUE\_NAME | Ingress\_Policing | where Cos\_Options.OPTION\_NAME = 'POLICING'.  Convert 'ENABLE' to 'ON', 'DISABLE' to 'OFF' |
| GRID.PORT.PE\_INGRESS\_COS\_ELEMENT for access circuit  or  GRID.SERVICE\_CONNECTION.PE\_INGRESS\_COS\_ELEMENT for service\_connection | Ingress\_Cos\_Element |  |
| GRID.PORT.PE\_INGRESS\_COS1\_PERCENTAGE for access circuit or  GRID.PORT.PE\_INGRESS\_COS2V\_PERCENTAGE for access circuit or  GRID.PORT.PE\_INGRESS\_COS2\_PERCENTAGE for access circuit or  GRID.PORT.PE\_INGRESS\_COS3\_PERCENTAGE for access circuit or  GRID.PORT.PE\_INGRESS\_COS4\_PERCENTAGE for access circuit or  GRID.PORT.PE\_INGRESS\_COS5\_PERCENTAGE for access circuit or  GRID.SERVICE\_CONNECTION.PE\_INGRESS\_COS1\_PERCENTAGE for service\_connection or  GRID.SERVICE\_CONNECTION.PE\_INGRESS\_COS2V\_PERCENTAGE for service\_connection or  GRID.SERVICE\_CONNECTION.PE\_INGRESS\_COS2\_PERCENTAGE for service\_connection or  GRID.SERVICE\_CONNECTION.PE\_INGRESS\_COS3\_PERCENTAGE for service\_connection or  GRID.SERVICE\_CONNECTION.PE\_INGRESS\_COS4\_PERCENTAGE for service\_connection or  GRID.SERVICE\_CONNECTION.PE\_INGRESS\_COS5\_PERCENTAGE for service\_connection | Ingress\_Cos\_Percentage |  |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| ASSET.ID | GDB.ASSET\_EXT\_NETWORK\_CONNECTION.ETE\_SERVICE\_CONNECTION\_KEY |
| GDB.ASSET\_EXT\_NETWORK\_CONNECTION.ETE\_SERVICE\_CONNECTION\_KEY | GRID.SERVICE\_CONNECTION.ETE\_SERVICE\_CONNECTION\_KEY |
| GDB.ASSET\_EXT\_ACCESS\_CIRCUIT.ETE\_ACCESS\_KEY | GRID.CIRCUIT.ETE\_ACCESS\_KEY |
| GRID.CIRCUIT.ETE\_ACCESS\_KEY | GRID.ACCESS\_CIRCUIT.ETE\_ACCESS\_KEY |
| GRID.ACCESS\_CIRCUIT.ETE\_PORT\_KEY | GRID.PORT.ETE\_PORT\_KEY |
| ICORE.COS\_CLASSES.Class\_Id | Icore.Cos\_Allocations.Class\_Id |
| Icore.Cos\_Allocations.Profile\_Id | Icore.Site.Ingress\_Profile\_Id |
| Icore.Cos\_Profile.Profile\_Id | Icore.Site.Ingress\_Profile\_Id |
| ICORE.PVC.PVC\_ID | ICORE.NETWORK\_CONNECTION.PVC\_ID |
| ICORE.NETWORK\_CONNECTION.PVC\_ID | GDB.ASSET\_EXT\_NETWORK\_CONNECTION.NETWORK\_CONNECTION\_ID |
| ICORE.PVC.PVC\_ID | ICORE.IPFR.PVC\_ID |
| Icore.ipfr.ingress\_profile\_id | Icore.Cos\_Profile.Profile\_Id |
| Icore.cos\_class2opt\_map.class\_option\_id | Icore.cos\_options.class\_option\_id |
| Icore.cos\_class2opt\_map.option\_value\_id | Icore.cos\_option\_values.option\_value\_id |
| Icore.site.site\_id | Icore.cos\_selected\_opt.key\_id |
| Icore.cos\_selected\_opt.cos\_option\_map\_id | Icore.cos\_class2opt\_map.cos\_option\_map\_id |

Note: Please find the attachments for queries which can be used as reference. COS\_PVC.txt can be referred for Network Connection, COS\_SITE.txt can be referred for Access Circuit

A one time full load of the GDB COS\_DETAIL\_EGRESS\_DATA table for all Access Circuit POKs will be needed to initialize this table.

On subsequent loads, a new record should be inserted if a new Access Circuit POK was added to GDB COS\_DETAIL\_EGRESS\_DATA.

|  |  |  |
| --- | --- | --- |
| **ICORE table.column** | **GDB COS\_DETAIL\_EGRESS\_DATA column** | **Processing comment** |
| ~~Nc3\_mat.cos\_profile.profile\_number~~ or  cos\_profile.profile\_id , for network connection  GRID.SERVICE\_CONNECTION.PE\_EGRESS\_PROFILE\_ID, for network connection  site.ingress\_profile\_id, for access circuit | Egress\_profile\_Id |  |
| Cos\_classes.class\_name | Egress\_COS\_Class | Present in 'COS1','COS2','COS3','COS4', ‘COS2V’, ‘COS5’ |
| Cos\_allocations.class\_allocation | Egress\_Value |  |
| Cos\_Option\_Values.OPTION\_VALUE\_NAME | Egress\_Queuing | where Cos\_Options.OPTION\_NAME = 'QUEUING'.  Convert 'ENABLE' to 'ON', 'DISABLE' to 'OFF' |
| Cos\_Option\_Values.OPTION\_VALUE\_NAME | Egress\_Shaping | where Cos\_Options.OPTION\_NAME = 'SHAPING'.  Convert 'ENABLE' to 'ON', 'DISABLE' to 'OFF' |
| GRID.PORT.PE\_EGRESS\_COS\_ELEMENT for access circuit or  GRID.SERVICE\_CONNECTION.PE\_EGRESS\_COS\_ELEMENT for service\_connection | Egress\_Cos\_Element |  |
| GRID.PORT.PE\_EGRESS\_COS1\_PERCENTAGE for access circuit or  GRID.PORT.PE\_EGRESS\_COS2V\_PERCENTAGE for access circuit or  GRID.PORT.PE\_EGRESS\_COS2\_PERCENTAGE for access circuit or  GRID.PORT.PE\_EGRESS\_COS3\_PERCENTAGE for access circuit or  GRID.PORT.PE\_EGRESS\_COS4\_PERCENTAGE for access circuit or  GRID.PORT.PE\_EGRESS\_COS5\_PERCENTAGE for access circuit or  GRID.SERVICE\_CONNECTION.PE\_EGRESS\_COS1\_PERCENTAGE for service\_connection or  GRID.SERVICE\_CONNECTION.PE\_EGRESS\_COS2V\_PERCENTAGE for service\_connection or  GRID.SERVICE\_CONNECTION.PE\_EGRESS\_COS2\_PERCENTAGE for service\_connection or  GRID.SERVICE\_CONNECTION.PE\_EGRESS\_COS3\_PERCENTAGE for service\_connection or  GRID.SERVICE\_CONNECTION.PE\_EGRESS\_COS4\_PERCENTAGE for service\_connection or  GRID.SERVICE\_CONNECTION.PE\_EGRESS\_COS5\_PERCENTAGE for service\_connection | Egress\_Cos\_Percentage |  |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| ASSET.ID | GDB.ASSET\_EXT\_NETWORK\_CONNECTION.ETE\_SERVICE\_CONNECTION\_KEY |
| GDB.ASSET\_EXT\_NETWORK\_CONNECTION.ETE\_SERVICE\_CONNECTION\_KEY | GRID.SERVICE\_CONNECTION.ETE\_SERVICE\_CONNECTION\_KEY |
| GDB.ASSET\_EXT\_ACCESS\_CIRCUIT.ETE\_ACCESS\_KEY | GRID.CIRCUIT.ETE\_ACCESS\_KEY |
| GRID.CIRCUIT.ETE\_ACCESS\_KEY | GRID.ACCESS\_CIRCUIT.ETE\_ACCESS\_KEY |
| GRID.ACCESS\_CIRCUIT.ETE\_PORT\_KEY | GRID.PORT.ETE\_PORT\_KEY |
| ICORE.COS\_CLASSES.Class\_Id | Icore.Cos\_Allocations.Class\_Id |
| Icore.Cos\_Allocations.Profile\_Id | Icore.Site.Egress\_Profile\_Id |
| Icore.Cos\_Profile.Profile\_Id | Icore.Site.Egress\_Profile\_Id |
| ICORE.PVC.PVC\_ID | ICORE.NETWORK\_CONNECTION.PVC\_ID |
| ICORE.NETWORK\_CONNECTION.PVC\_ID | GDB.ASSET\_EXT\_NETWORK\_CONNECTION.NETWORK\_CONNECTION\_ID |
| ICORE.PVC.PVC\_ID | ICORE.IPFR.PVC\_ID |
| Icore.ipfr.Egress\_profile\_id | Icore.Cos\_Profile.Profile\_Id |
| Icore.cos\_class2opt\_map.class\_option\_id | Icore.cos\_options.class\_option\_id |
| Icore.cos\_class2opt\_map.option\_value\_id | Icore.cos\_option\_values.option\_value\_id |
| Icore.site.site\_id | Icore.cos\_selected\_opt.key\_id |
| Icore.cos\_selected\_opt.cos\_option\_map\_id | Icore.cos\_class2opt\_map.cos\_option\_map\_id |
| GDB.ASSET\_EXT\_NETWORK\_CONNECTION.ETE\_SERVICE\_CONNECTION\_KEY | GRID.SERVICE\_CONNECTION.ETE\_SERVICE\_CONNECTION\_KEY |

Note: Please refer to attachments under COS\_DETAIL\_INGRESS\_DATA loading which has queries that can be used as reference.

</295359-US325561>

Note:

If any POK is missing in parent table when trying to insert into child table, then first parent table should be inserted and then child table should be inserted so that Parent Child relationship is maintained while loading into tables.

### Data Index Maintenance for PORT Data [254035b]

#### HLD-254035b-GCP-DATA-INDEX-001 [Data Index Processing Rules]

This is a “placeholder” requirement. The real requirement is contained in the HLD “**GCP-SA-HLD-for-GCP-GDB.254035b.docx**” (outdated !) and “**GCP-SA-HLD-for-GCP-GDB.254035c.For-UIS-et-al-Load.docx**”.

#### HLD-254035b-GCP-DATA-INDEX-100 [Data Index initial population for PORT data]

This is a “placeholder” requirement. The real requirement is contained in the HLD “**GCP-SA-HLD-for-GCP-GDB.254035b.docx**” (outdated !) and “**GCP-SA-HLD-for-GCP-GDB.254035c.For-UIS-et-al-Load.docx**”.

#### HLD-254035b-GCP-DATA-INDEX-110 [Data Index codes tables prepopulation for PORT data]

This is a “placeholder” requirement. The real requirement is contained in the HLD “**GCP-SA-HLD-for-GCP-GDB.254035b.docx**” (outdated !) and “**GCP-SA-HLD-for-GCP-GDB.254035c.For-UIS-et-al-Load.docx**”.

### Service Delivery Data Flow into Golden Database [Icore Data Flow] for Asset-to-Service-Option Associations (254035c)

#### HLD-254035c- GCP-FLOW-ICORE-300 [Icore Data Flow into (“Access Circuit”) Asset-to-Service-Option: SO\_MANAGED\_INDICATOR]

<BEGIN 254035c.CR-108224>

This requirement is for covering the relationship between an “access circuit” asset and the “SO\_MANAGED\_INDICATOR” service option.

Implement the following:

Process the following ICORE records into the GDB asset to service option (SO\_MANAGED\_INDICATOR) association:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**SERVICE\_OPTION**’, ROLE with FUNCTION\_TYPE ‘**HAS\_SERVICE\_OPTION**’ and FUNCTION\_ROLE NULL |
| CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id +  CUST\_ACCESS\_CKT.seq\_no | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| - | Id\_object\_to [via lookup in GDB SERV\_OPT and SERV\_OPT\_TYPE] | Use SERV\_OPT.id which references SERV\_OPT\_TYPE having serv\_opt\_name = ‘**SO\_MANAGED\_INDICATOR**’. |
| CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id +  CUST\_ACCESS\_CKT.seq\_no | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

##### Logic used for creating the association

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SERVICE\_ASGMT.site\_id | CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| SERVICE | SERV\_NAME | in ( ‘Managed Srvc’) |

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that all asset-loading steps have been executed.*

*Note:*

*Ensure that the process implemented by this design item (requirement) is added to and executed in the context of currently deployed processes/jobs that create association data for ICORE.*

#### HLD-254035c- GCP-FLOW-ICORE-300.1 [ICORE Data Flow into (“Access Circuit”) Asset-to-Service-Option: SO\_MANAGED\_INDICATOR <INIT>]

This requirement is for doing an initial load for initially building the relationship an “access circuit” asset and the “SO\_MANAGED\_INDICATOR” service option.

This initial load needs to have been completed before requirement “HLD-254035c- GCP-FLOW-ICORE-300 [Icore Data Flow into (“Access Circuit”) Asset-to-Service-Option: SO\_MANAGED\_INDICATOR]” can be executed for the first time.

The execution of this process requires the maintenance of a change tracking event into GDB.CHANGE\_TRACKING, CHANGE\_SYSTEM and CHANGE\_USER. The change tracking event for the service delivery data flow must use CHANGE\_SYSTEM.name = ’INITIAL\_LOAD’ and CHANGE\_USER.name = ’ INITIAL\_LOAD’. It is sufficient to have one change tracking event for all affected records during a single execution of this process.

NOTE that this requirement is a “one-time” load.

The set of assets for which this initial load applies is as follows  
(all “access circuits” loaded from ICORE):

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| GDB.ASSET\_EXT\_ACCESS\_CIRCUIT.id\_asset | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id\_meta\_table | METADATA.META\_TABLE.id |
| METADATA.META\_SYSTEM.id | METADATA.META\_TABLE.id\_meta\_system |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| METADATA.META\_SYSTEM | NAME | ICORE |

For the initial load the logic from requirement “HLD-254035c- GCP-FLOW-ICORE-300 [Icore Data Flow into (“Access Circuit”) Asset-to-Service-Option: SO\_MANAGED\_INDICATOR]” has to be used.

<END 254035c.CR-108224>

#### HLD-254035c- GCP-FLOW-ICORE-301 [Icore Data Flow into (“Access Circuit”) Asset-to-Service-Option: SO\_MULTICAST\_ENABLED\_INDICATOR]

<BEGIN 254035c.CR-108224>

This requirement is for covering the relationship between an “access circuit” asset and the “SO\_MULTICAST\_ENABLED\_INDICATOR” service option.

Implement the following:

Process the following ICORE records into the GDB asset to service option (SO\_MULTICAST\_ENABLED\_INDICATOR) association:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**SERVICE\_OPTION**’, ROLE with FUNCTION\_TYPE ‘**HAS\_SERVICE\_OPTION**’ and FUNCTION\_ROLE NULL |
| CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id +  CUST\_ACCESS\_CKT.seq\_no | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| - | Id\_object\_to [via lookup in GDB SERV\_OPT and SERV\_OPT\_TYPE] | Use SERV\_OPT.id which references SERV\_OPT\_TYPE having serv\_opt\_name = ‘**SO\_MULTICAST\_ENABLED\_INDICATOR**’. |
| CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id +  CUST\_ACCESS\_CKT.seq\_no | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

##### Logic used for creating the association

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SERVICE\_ASGMT.site\_id | CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| SERVICE | SERV\_NAME | in ( ‘MULTICAST’) |

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that all asset-loading steps have been executed.*

*Note:*

*Ensure that the process implemented by this design item (requirement) is added to and executed in the context of currently deployed processes/jobs that create association data for ICORE.*

#### HLD-254035c- GCP-FLOW-ICORE-301.1 [ICORE Data Flow into (“Access Circuit”) Asset-to-Service-Option: SO\_MULTICAST\_ENABLED\_INDICATOR <INIT>]

This requirement is for doing an initial load for initially building the relationship an “access circuit” asset and the “SO\_MULTICAST\_ENABLED\_INDICATOR” service option.

This initial load needs to have been completed before requirement “HLD-254035c- GCP-FLOW-ICORE-301 [Icore Data Flow into (“Access Circuit”) Asset-to-Service-Option: SO\_MULTICAST\_ENABLED\_INDICATOR]” can be executed for the first time.

The execution of this process requires the maintenance of a change tracking event into GDB.CHANGE\_TRACKING, CHANGE\_SYSTEM and CHANGE\_USER. The change tracking event for the service delivery data flow must use CHANGE\_SYSTEM.name = ’INITIAL\_LOAD’ and CHANGE\_USER.name = ’ INITIAL\_LOAD’. It is sufficient to have one change tracking event for all affected records during a single execution of this process.

NOTE that this requirement is a “one-time” load.

The set of assets for which this initial load applies is as follows  
(all “access circuits” loaded from ICORE):

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| GDB.ASSET\_EXT\_ACCESS\_CIRCUIT.id\_asset | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id\_meta\_table | METADATA.META\_TABLE.id |
| METADATA.META\_SYSTEM.id | METADATA.META\_TABLE.id\_meta\_system |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| METADATA.META\_SYSTEM | NAME | ICORE |

For the initial load the logic from requirement “HLD-254035c- GCP-FLOW-ICORE-301 [Icore Data Flow into (“Access Circuit”) Asset-to-Service-Option: SO\_MULTICAST\_ENABLED\_INDICATOR]” has to be used.

<END 254035c.CR-108224>

#### HLD-254035c- GCP-FLOW-ICORE-302 [Icore Data Flow into (“Access Circuit”) Asset-to-Service-Option: SO\_UNILINK]

<BEGIN 254035c.CR-108224>

This requirement is for covering the relationship between an “access circuit” asset and the “SO\_UNILINK” service option.

Implement the following:

Process the following ICORE records into the GDB asset to service option (SO\_UNILINK) association:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**SERVICE\_OPTION**’, ROLE with FUNCTION\_TYPE ‘**HAS\_SERVICE\_OPTION**’ and FUNCTION\_ROLE NULL |
| CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id +  CUST\_ACCESS\_CKT.seq\_no | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| - | Id\_object\_to [via lookup in GDB SERV\_OPT and SERV\_OPT\_TYPE] | Use SERV\_OPT.id which references SERV\_OPT\_TYPE having serv\_opt\_name = ‘**SO\_UNILINK**’. |
| CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id +  CUST\_ACCESS\_CKT.seq\_no | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

##### Logic used for creating the association

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SERVICE\_ASGMT.site\_id | CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| SERVICE | SERV\_NAME | in ( ‘UNILINK’) |

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that all asset-loading steps have been executed.*

*Note:*

*Ensure that the process implemented by this design item (requirement) is added to and executed in the context of currently deployed processes/jobs that create association data for ICORE.*

#### HLD-254035c- GCP-FLOW-ICORE-302.1 [ICORE Data Flow into (“Access Circuit”) Asset-to-Service-Option: SO\_UNILINK <INIT>]

This requirement is for doing an initial load for initially building the relationship an “access circuit” asset and the “SO\_UNILINK” service option.

This initial load needs to have been completed before requirement “HLD-254035c- GCP-FLOW-ICORE-302 [Icore Data Flow into (“Access Circuit”) Asset-to-Service-Option: SO\_UNILINK]” can be executed for the first time.

The execution of this process requires the maintenance of a change tracking event into GDB.CHANGE\_TRACKING, CHANGE\_SYSTEM and CHANGE\_USER. The change tracking event for the service delivery data flow must use CHANGE\_SYSTEM.name = ’INITIAL\_LOAD’ and CHANGE\_USER.name = ’ INITIAL\_LOAD’. It is sufficient to have one change tracking event for all affected records during a single execution of this process.

NOTE that this requirement is a “one-time” load.

The set of assets for which this initial load applies is as follows  
(all “access circuits” loaded from ICORE):

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| GDB.ASSET\_EXT\_ACCESS\_CIRCUIT.id\_asset | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id\_meta\_table | METADATA.META\_TABLE.id |
| METADATA.META\_SYSTEM.id | METADATA.META\_TABLE.id\_meta\_system |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| METADATA.META\_SYSTEM | NAME | ICORE |

For the initial load the logic from requirement “HLD-254035c- GCP-FLOW-ICORE-302 [Icore Data Flow into (“Access Circuit”) Asset-to-Service-Option: SO\_UNILINK]” has to be used.

<END 254035c.CR-108224>

#### HLD-254035c- GCP-FLOW-ICORE-303 [Icore Data Flow into (“Access Circuit”) Asset-to-Service-Option: SO\_BVOIP\_ENABLED]

<BEGIN 254035c.CR-106926>

This requirement is for covering the relationship between an “access circuit” asset and the “SO\_BVOIP\_ENABLED” service option.

Implement the following:

Process the following ICORE records into the GDB asset to service option (SO\_BVOIP\_ENABLED) association:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**SERVICE\_OPTION**’, ROLE with FUNCTION\_TYPE ‘**HAS\_SERVICE\_OPTION**’ and FUNCTION\_ROLE NULL |
| CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id +  CUST\_ACCESS\_CKT.seq\_no | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| - | Id\_object\_to [via lookup in GDB SERV\_OPT and SERV\_OPT\_TYPE] | Use SERV\_OPT.id which references SERV\_OPT\_TYPE having serv\_opt\_name = ‘**SO\_BVOIP\_ENABLED**’. |
| CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id +  CUST\_ACCESS\_CKT.seq\_no | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

##### Logic used for creating the association

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SERVICE\_ASGMT.site\_id | CUST\_ACCESS.site\_id <or> CUST\_ACCESS\_CKT.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| SERVICE | SERV\_NAME | in ( ‘VoIP’) |

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that all asset-loading steps have been executed.*

*Note:*

*Ensure that the process implemented by this design item (requirement) is added to and executed in the context of currently deployed processes/jobs that create association data for ICORE.*

#### HLD-254035c- GCP-FLOW-ICORE-303.1 [ICORE Data Flow into (“Access Circuit”) Asset-to-Service-Option: SO\_BVOIP\_ENABLED <INIT>]

This requirement is for doing an initial load for initially building the relationship an “access circuit” asset and the “SO\_BVOIP\_ENABLED” service option.

This initial load needs to have been completed before requirement “HLD-254035c- GCP-FLOW-ICORE-303 [Icore Data Flow into (“Access Circuit”) Asset-to-Service-Option: SO\_BVOIP\_ENABLED]” can be executed for the first time.

The execution of this process requires the maintenance of a change tracking event into GDB.CHANGE\_TRACKING, CHANGE\_SYSTEM and CHANGE\_USER. The change tracking event for the service delivery data flow must use CHANGE\_SYSTEM.name = ’INITIAL\_LOAD’ and CHANGE\_USER.name = ’ INITIAL\_LOAD’. It is sufficient to have one change tracking event for all affected records during a single execution of this process.

NOTE that this requirement is a “one-time” load.

The set of assets for which this initial load applies is as follows  
(all “access circuits” loaded from ICORE):

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| GDB.ASSET\_EXT\_ACCESS\_CIRCUIT.id\_asset | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id\_meta\_table | METADATA.META\_TABLE.id |
| METADATA.META\_SYSTEM.id | METADATA.META\_TABLE.id\_meta\_system |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| METADATA.META\_SYSTEM | NAME | ICORE |

For the initial load the logic from requirement “HLD-254035c- GCP-FLOW-ICORE-303 [Icore Data Flow into (“Access Circuit”) Asset-to-Service-Option: SO\_BVOIP\_ENABLED]” has to be used.

<END 254035c.CR-106926>

#### HLD-254035c- GCP-FLOW-ICORE-320 [ICORE Data Flow into (“Network Connection”) Asset-to-Service-Option: SO\_MANAGED\_INDICATOR]

<BEGIN 254035c.CR-108224>

This requirement is for covering the relationship between an “network connection” asset and the “SO\_MANAGED\_INDICATOR” service option.

*NOTE: For the following we have to do with data based on a PVC record and either a PVC.pvc\_lsite\_id or a PVC.pvc\_rsite\_id. For the Enterprise Key Translation (EKT) this is “abstracted away” by making use of the “NETWORK\_CONNECTION” or the “NETWORK\_CONNECTION\_INET\_VLAN” view.*

Implement the following:

**(1) “Layer-2” and “Layer-3” “Network Connections”**

For “Layer-2” and “Layer-3” “Network Connections” process the following ICORE records into the GDB asset to service option (SO\_MANAGED\_INDICATOR) association:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**SERVICE\_OPTION**’, ROLE with FUNCTION\_TYPE ‘**HAS\_SERVICE\_OPTION**’ and FUNCTION\_ROLE NULL |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| - | Id\_object\_to [via lookup in GDB SERV\_OPT and SERV\_OPT\_TYPE] | Use SERV\_OPT.id which references SERV\_OPT\_TYPE having serv\_opt\_name = ‘**SO\_MANAGED\_INDICATOR**’. |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

**(2) “Internet VLAN” “Network Connections”**

For “Internet VLAN” “Network Connections” process the following ICORE records into the GDB asset to service option (SO\_MANAGED\_INDICATOR) association:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**SERVICE\_OPTION**’, ROLE with FUNCTION\_TYPE ‘**HAS\_SERVICE\_OPTION**’ and FUNCTION\_ROLE NULL |
| NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id; | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| - | Id\_object\_to [via lookup in GDB SERV\_OPT and SERV\_OPT\_TYPE] | Use SERV\_OPT.id which references SERV\_OPT\_TYPE having serv\_opt\_name = ‘**SO\_MANAGED\_INDICATOR**’. |
| NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id; | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

##### Logic used for creating the association

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SERVICE\_ASGMT.site\_id | NETWORK\_CONNECTION.site\_id <or> NETWORK\_CONNECTION\_INET\_VLAN.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| SERVICE | SERV\_NAME | in ( ‘Managed Srvc’) |
| SERVICE\_OPTION | SERV\_OPT | not exists ( ‘DSU’, ‘DSU Enhanced’,  ‘V6 CSU’) |

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that all asset-loading steps have been executed.*

*Note:*

*Ensure that the process implemented by this design item (requirement) is added to and executed in the context of currently deployed processes/jobs that create association data for ICORE.*

#### HLD-254035c- GCP-FLOW-ICORE-320.1 [ICORE Data Flow into (“Network Connection”) Asset-to-Service-Option: SO\_MANAGED\_INDICATOR <INIT>]

This requirement is for doing an initial load for initially building the relationship between an “network connection” asset and the “SO\_MANAGED\_INDICATOR” service option.

This initial load needs to have been completed before requirement “HLD-254035c- GCP-FLOW-ICORE-320 [ICORE Data Flow into (“Network Connection”) Asset-to-Service-Option: SO\_MANAGED\_INDICATOR]” can be executed for the first time.

The execution of this process requires the maintenance of a change tracking event into GDB.CHANGE\_TRACKING, CHANGE\_SYSTEM and CHANGE\_USER. The change tracking event for the service delivery data flow must use CHANGE\_SYSTEM.name = ’INITIAL\_LOAD’ and CHANGE\_USER.name = ’ INITIAL\_LOAD’. It is sufficient to have one change tracking event for all affected records during a single execution of this process.

NOTE that this requirement is a “one-time” load.

The set of assets for which this initial load applies is as follows  
(all “network connections” loaded from ICORE):

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| GDB.ASSET\_EXT\_NETWORK\_CONNECTION.id\_asset | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id\_meta\_table | METADATA.META\_TABLE.id |
| METADATA.META\_SYSTEM.id | METADATA.META\_TABLE.id\_meta\_system |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| METADATA.META\_SYSTEM | NAME | ICORE |

For the initial load the logic from requirement “HLD-254035c- GCP-FLOW-ICORE-320 [ICORE Data Flow into (“Network Connection”) Asset-to-Service-Option: SO\_MANAGED\_INDICATOR]” has to be used.

<END 254035c.CR-108224>

#### HLD-254035c- GCP-FLOW-ICORE-321 [ICORE Data Flow into (“Network Connection”) Asset-to-Service-Option: SO\_MULTICAST\_ENABLED\_INDICATOR]

<BEGIN 254035c.CR-108224>

This requirement is for covering the relationship between an “network connection” asset and the “SO\_MULTICAST\_ENABLED\_INDICATOR” service option.

*NOTE: For the following we have to do with data based on a PVC record and either a PVC.pvc\_lsite\_id or a PVC.pvc\_rsite\_id. For the Enterprise Key Translation (EKT) this is “abstracted away” by making use of the “NETWORK\_CONNECTION” or the “NETWORK\_CONNECTION\_INET\_VLAN” view.*

Implement the following:

*NOTE: This is for “Layer-3” network connections only, as only those ones have “IPFR” data !*

**(1) “Layer-3” “Network Connections”**

For “Layer-3” “Network Connections” process the following ICORE records into the GDB asset to service option (SO\_MULTICAST\_ENABLED\_INDICATOR) association:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**SERVICE\_OPTION**’, ROLE with FUNCTION\_TYPE ‘**HAS\_SERVICE\_OPTION**’ and FUNCTION\_ROLE NULL |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| - | Id\_object\_to [via lookup in GDB SERV\_OPT and SERV\_OPT\_TYPE] | Use SERV\_OPT.id which references SERV\_OPT\_TYPE having serv\_opt\_name = ‘**SO\_MULTICAST\_ENABLED\_INDICATOR**’. |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

##### Logic used for creating the association

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| NETWORK\_CONNECTION.pvc\_id | IPFR.pvc\_id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| IPFR | MULTICAST | in (‘Y’) |

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that all asset-loading steps have been executed.*

*Note:*

*Ensure that the process implemented by this design item (requirement) is added to and executed in the context of currently deployed processes/jobs that create association data for ICORE.*

#### HLD-254035c- GCP-FLOW-ICORE-321.1 [ICORE Data Flow into (“Network Connection”) Asset-to-Service-Option: SO\_MULTICAST\_ENABLED\_INDICATOR <INIT>]

This requirement is for doing an initial load for initially building the relationship between an “network connection” asset and the “SO\_MULTICAST\_ENABLED\_INDICATOR” service option.

This initial load needs to have been completed before requirement “HLD-254035c- GCP-FLOW-ICORE-321 [ICORE Data Flow into (“Network Connection”) Asset-to-Service-Option: SO\_MULTICAST\_ENABLED\_INDICATOR]” can be executed for the first time.

The execution of this process requires the maintenance of a change tracking event into GDB.CHANGE\_TRACKING, CHANGE\_SYSTEM and CHANGE\_USER. The change tracking event for the service delivery data flow must use CHANGE\_SYSTEM.name = ’INITIAL\_LOAD’ and CHANGE\_USER.name = ’ INITIAL\_LOAD’. It is sufficient to have one change tracking event for all affected records during a single execution of this process.

NOTE that this requirement is a “one-time” load.

The set of assets for which this initial load applies is as follows  
(all “network connections” loaded from ICORE):

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| GDB.ASSET\_EXT\_NETWORK\_CONNECTION.id\_asset | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id\_meta\_table | METADATA.META\_TABLE.id |
| METADATA.META\_SYSTEM.id | METADATA.META\_TABLE.id\_meta\_system |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| METADATA.META\_SYSTEM | NAME | ICORE |

For the initial load the logic from requirement “HLD-254035c- GCP-FLOW-ICORE-321 [ICORE Data Flow into (“Network Connection”) Asset-to-Service-Option: SO\_MULTICAST\_ENABLED\_INDICATOR]” has to be used.

<END 254035c.CR-108224>

#### HLD-254035c- GCP-FLOW-ICORE-322 [ICORE Data Flow into (“Network Connection”) Asset-to-Service-Option: SO\_SERVICE\_INTEROPERABILITY]

<BEGIN 254035c.CR-108224>

This requirement is for covering the relationship between an “network connection” asset and the “SO\_SERVICE\_INTEROPERABILITY” service option.

*NOTE: For the following we have to do with data based on a PVC record and either a PVC.pvc\_lsite\_id or a PVC.pvc\_rsite\_id. For the Enterprise Key Translation (EKT) this is “abstracted away” by making use of the “NETWORK\_CONNECTION” or the “NETWORK\_CONNECTION\_INET\_VLAN” view.*

Implement the following:

**(1) “Layer-2” and “Layer-3” “Network Connections”**

For “Layer-2” and “Layer-3” “Network Connections” process the following ICORE records into the GDB asset to service option (SO\_SERVICE\_INTEROPERABILITY) association:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**SERVICE\_OPTION**’, ROLE with FUNCTION\_TYPE ‘**HAS\_SERVICE\_OPTION**’ and FUNCTION\_ROLE NULL |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| - | Id\_object\_to [via lookup in GDB SERV\_OPT and SERV\_OPT\_TYPE] | Use SERV\_OPT.id which references SERV\_OPT\_TYPE having serv\_opt\_name = ‘**SO\_SERVICE\_INTEROPERABILITY**’. |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

**(2) “Internet VLAN” “Network Connections”**

For “Internet VLAN” “Network Connections” process the following ICORE records into the GDB asset to service option (SO\_SERVICE\_INTEROPERABILITY) association:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**SERVICE\_OPTION**’, ROLE with FUNCTION\_TYPE ‘**HAS\_SERVICE\_OPTION**’ and FUNCTION\_ROLE NULL |
| NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id; | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| - | Id\_object\_to [via lookup in GDB SERV\_OPT and SERV\_OPT\_TYPE] | Use SERV\_OPT.id which references SERV\_OPT\_TYPE having serv\_opt\_name = ‘**SO\_SERVICE\_INTEROPERABILITY**’. |
| NETWORK\_CONNECTION\_INET\_VLAN.vpn\_id + NETWORK\_CONNECTION\_INET\_VLAN.pvc\_id + NETWORK\_CONNECTION\_INET\_VLAN.site\_id; | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

##### Logic used for creating the association

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| NETWORK\_CONNECTION.pvc\_id | PVC.pvc\_id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| PVC | PVC\_INTERWORKED | in (‘Y’) |

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that all asset-loading steps have been executed.*

*Note:*

*Ensure that the process implemented by this design item (requirement) is added to and executed in the context of currently deployed processes/jobs that create association data for ICORE.*

#### HLD-254035c- GCP-FLOW-ICORE-322.1 [ICORE Data Flow into (“Network Connection”) Asset-to-Service-Option: SO\_SERVICE\_INTEROPERABILITY <INIT>]

This requirement is for doing an initial load for initially building the relationship between an “network connection” asset and the “SO\_SERVICE\_INTEROPERABILITY” service option.

This initial load needs to have been completed before requirement “HLD-254035c- GCP-FLOW-ICORE-322 [ICORE Data Flow into (“Network Connection”) Asset-to-Service-Option: SO\_SERVICE\_INTEROPERABILITY]” can be executed for the first time.

The execution of this process requires the maintenance of a change tracking event into GDB.CHANGE\_TRACKING, CHANGE\_SYSTEM and CHANGE\_USER. The change tracking event for the service delivery data flow must use CHANGE\_SYSTEM.name = ’INITIAL\_LOAD’ and CHANGE\_USER.name = ’ INITIAL\_LOAD’. It is sufficient to have one change tracking event for all affected records during a single execution of this process.

NOTE that this requirement is a “one-time” load.

The set of assets for which this initial load applies is as follows  
(all “network connections” loaded from ICORE):

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| GDB.ASSET\_EXT\_NETWORK\_CONNECTION.id\_asset | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id\_meta\_table | METADATA.META\_TABLE.id |
| METADATA.META\_SYSTEM.id | METADATA.META\_TABLE.id\_meta\_system |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| METADATA.META\_SYSTEM | NAME | ICORE |

For the initial load the logic from requirement “HLD-254035c- GCP-FLOW-ICORE-322 [ICORE Data Flow into (“Network Connection”) Asset-to-Service-Option: SO\_SERVICE\_INTEROPERABILITY]” has to be used.

<END 254035c.CR-108224>

#### HLD-254035c- GCP-FLOW-ICORE-323 [ICORE Data Flow into (“Network Connection”) Asset-to-Service-Option: SO\_MD5\_INDICATOR]

<BEGIN 254035c.CR-108224>

This requirement is for covering the relationship between an “network connection” asset and the “SO\_MD5\_INDICATOR” service option.

*NOTE: For the following we have to do with data based on a PVC record and either a PVC.pvc\_lsite\_id or a PVC.pvc\_rsite\_id. For the Enterprise Key Translation (EKT) this is “abstracted away” by making use of the “NETWORK\_CONNECTION” or the “NETWORK\_CONNECTION\_INET\_VLAN” view.*

Implement the following:

*NOTE: This is for “Layer-3” network connections only, as only those ones have “IPFR” data !*

**(1) “Layer-3” “Network Connections”**

For “Layer-3” “Network Connections” process the following ICORE records into the GDB asset to service option (SO\_MD5\_INDICATOR) association:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**SERVICE\_OPTION**’, ROLE with FUNCTION\_TYPE ‘**HAS\_SERVICE\_OPTION**’ and FUNCTION\_ROLE NULL |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| - | Id\_object\_to [via lookup in GDB SERV\_OPT and SERV\_OPT\_TYPE] | Use SERV\_OPT.id which references SERV\_OPT\_TYPE having serv\_opt\_name = ‘**SO\_MD5\_INDICATOR**’. |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

##### Logic used for creating the association

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| NETWORK\_CONNECTION.pvc\_id | IPFR.pvc\_id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| IPFR | MD5 | in (‘Y’) |

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that all asset-loading steps have been executed.*

*Note:*

*Ensure that the process implemented by this design item (requirement) is added to and executed in the context of currently deployed processes/jobs that create association data for ICORE.*

#### HLD-254035c- GCP-FLOW-ICORE-323.1 [ICORE Data Flow into (“Network Connection”) Asset-to-Service-Option: SO\_MD5\_INDICATOR <INIT>]

This requirement is for doing an initial load for initially building the relationship between an “network connection” asset and the “SO\_MD5\_INDICATOR” service option.

This initial load needs to have been completed before requirement “HLD-254035c- GCP-FLOW-ICORE-323 [ICORE Data Flow into (“Network Connection”) Asset-to-Service-Option: SO\_MD5\_INDICATOR]” can be executed for the first time.

The execution of this process requires the maintenance of a change tracking event into GDB.CHANGE\_TRACKING, CHANGE\_SYSTEM and CHANGE\_USER. The change tracking event for the service delivery data flow must use CHANGE\_SYSTEM.name = ’INITIAL\_LOAD’ and CHANGE\_USER.name = ’ INITIAL\_LOAD’. It is sufficient to have one change tracking event for all affected records during a single execution of this process.

NOTE that this requirement is a “one-time” load.

The set of assets for which this initial load applies is as follows:

The set of assets for which this initial load applies is as follows  
(all “network connections” loaded from ICORE):

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| GDB.ASSET\_EXT\_NETWORK\_CONNECTION.id\_asset | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id\_meta\_table | METADATA.META\_TABLE.id |
| METADATA.META\_SYSTEM.id | METADATA.META\_TABLE.id\_meta\_system |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| METADATA.META\_SYSTEM | NAME | ICORE |

For the initial load the logic from requirement “HLD-254035c- GCP-FLOW-ICORE-323 [ICORE Data Flow into (“Network Connection”) Asset-to-Service-Option: SO\_MD5\_INDICATOR]” has to be used.

<END 254035c.CR-108224>

#### HLD-254035c- GCP-FLOW-ICORE-324 [ICORE Data Flow into (“Network Connection”) Asset-to-Service-Option: SO\_CMTU\_INDICATOR]

<BEGIN 254035c.CR-108224>

This requirement is for covering the relationship between an “network connection” asset and the “SO\_CMTU\_INDICATOR” service option.

*NOTE: For the following we have to do with data based on a PVC record and either a PVC.pvc\_lsite\_id or a PVC.pvc\_rsite\_id. For the Enterprise Key Translation (EKT) this is “abstracted away” by making use of the “NETWORK\_CONNECTION” or the “NETWORK\_CONNECTION\_INET\_VLAN” view.*

Implement the following:

*NOTE: This is for “Layer-3” network connections only, as only those ones have “IPFR” data !*

**(1) “Layer-3” “Network Connections”**

For “Layer-3” “Network Connections” process the following ICORE records into the GDB asset to service option (SO\_CMTU\_INDICATOR) association:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**SERVICE\_OPTION**’, ROLE with FUNCTION\_TYPE ‘**HAS\_SERVICE\_OPTION**’ and FUNCTION\_ROLE NULL |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| - | Id\_object\_to [via lookup in GDB SERV\_OPT and SERV\_OPT\_TYPE] | Use SERV\_OPT.id which references SERV\_OPT\_TYPE having serv\_opt\_name = ‘**SO\_CMTU\_INDICATOR**’. |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

##### Logic used for creating the association

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| NETWORK\_CONNECTION.pvc\_id | IPFR.pvc\_id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| IPFR | PMTU | is not null |

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that all asset-loading steps have been executed.*

*Note:*

*Ensure that the process implemented by this design item (requirement) is added to and executed in the context of currently deployed processes/jobs that create association data for ICORE.*

#### HLD-254035c- GCP-FLOW-ICORE-324.1 [ICORE Data Flow into (“Network Connection”) Asset-to-Service-Option: SO\_CMTU\_INDICATOR <INIT>]

This requirement is for doing an initial load for initially building the relationship between an “network connection” asset and the “SO\_CMTU\_INDICATOR” service option.

This initial load needs to have been completed before requirement “HLD-254035c- GCP-FLOW-ICORE-324 [ICORE Data Flow into (“Network Connection”) Asset-to-Service-Option: SO\_CMTU\_INDICATOR]” can be executed for the first time.

The execution of this process requires the maintenance of a change tracking event into GDB.CHANGE\_TRACKING, CHANGE\_SYSTEM and CHANGE\_USER. The change tracking event for the service delivery data flow must use CHANGE\_SYSTEM.name = ’INITIAL\_LOAD’ and CHANGE\_USER.name = ’ INITIAL\_LOAD’. It is sufficient to have one change tracking event for all affected records during a single execution of this process.

NOTE that this requirement is a “one-time” load.

The set of assets for which this initial load applies is as follows  
(all “network connections” loaded from ICORE):

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| GDB.ASSET\_EXT\_NETWORK\_CONNECTION.id\_asset | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id\_meta\_table | METADATA.META\_TABLE.id |
| METADATA.META\_SYSTEM.id | METADATA.META\_TABLE.id\_meta\_system |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| METADATA.META\_SYSTEM | NAME | ICORE |

For the initial load the logic from requirement “HLD-254035c- GCP-FLOW-ICORE-324 [ICORE Data Flow into (“Network Connection”) Asset-to-Service-Option: SO\_CMTU\_INDICATOR]” has to be used.

<END 254035c.CR-108224>

#### HLD-254035c- GCP-FLOW-ICORE-325 [ICORE Data Flow into (“Network Connection”) Asset-to-Service-Option: SO\_BVOIP\_ENABLED]

<BEGIN 254035c.CR-106926>

This requirement is for covering the relationship between an “network connection” asset and the “SO\_BVOIP\_ENABLED” service option.

*NOTE: For the following we have to do with data based on a PVC record and either a PVC.pvc\_lsite\_id or a PVC.pvc\_rsite\_id. For the Enterprise Key Translation (EKT) this is “abstracted away” by making use of the “NETWORK\_CONNECTION” or the “NETWORK\_CONNECTION\_INET\_VLAN” view.*

Implement the following:

*NOTE: This is for “Layer-3” network connections only, as only those ones have “IPFR” data !*

**(1) “Layer-3” “Network Connections”**

For “Layer-3” “Network Connections” process the following ICORE records into the GDB asset to service option (SO\_BVOIP\_ENABLED) association:

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ASSOCIATION column** | **Processing comment** |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | Id\_role\_notation | NULL |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**ASSET**’, OBJECT\_TYPE\_TO ‘**SERVICE\_OPTION**’, ROLE with FUNCTION\_TYPE ‘**HAS\_SERVICE\_OPTION**’ and FUNCTION\_ROLE NULL |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | Id\_object\_what [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  <BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003> |
| - | Id\_object\_to [via lookup in GDB SERV\_OPT and SERV\_OPT\_TYPE] | Use SERV\_OPT.id which references SERV\_OPT\_TYPE having serv\_opt\_name = ‘**SO\_BVOIP\_ENABLED**’. |
| NETWORK\_CONNECTION.pvc\_id + NETWORK\_CONNECTION.site\_id; | Id\_source\_key [via Enterprise Key Translation] | Use SOURCE\_KEY.id which populated the respective ASSET.id  ~~<BEGIN 254035c.UTC-89113.003> Use SOURCE\_KEY\_CORRELATION.id\_source\_key\_master for SOURCE\_PROCESS according to all currently installed “asset unifications” (see: A Note on “Asset Unification”) where SOURCE\_KEY\_CORRELATION.id\_source\_key equals SOURCE\_KEY.id <END 254035c.UTC-89113.003>~~ |
| - | Gdb\_internal\_flags | NULL |

##### Logic used for creating the association

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| NETWORK\_CONNECTION.pvc\_id | IPFR.pvc\_id |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| IPFR | BVOIP | in (‘Y’) |

Do not create GDB.ASSOCIATION records if the referenced objects were not populated into the GDB, i.e. filtering rules prevented records to be created into the GDB objects, those objects are not to be created by this step and association records that would refer to such object records are to be omitted.

*Note:*

*A filtered object for example is the Asset via the SERVICE\_TYPE\_BLOCKED\_SYSTEM filtering.*

*Note:*

*This step requires that all asset-loading steps have been executed.*

*Note:*

*Ensure that the process implemented by this design item (requirement) is added to and executed in the context of currently deployed processes/jobs that create association data for ICORE.*

#### HLD-254035c- GCP-FLOW-ICORE-325.1 [ICORE Data Flow into (“Network Connection”) Asset-to-Service-Option: SO\_BVOIP\_ENABLED <INIT>]

This requirement is for doing an initial load for initially building the relationship between an “network connection” asset and the “SO\_BVOIP\_ENABLED” service option.

This initial load needs to have been completed before requirement “HLD-254035c- GCP-FLOW-ICORE-325 [ICORE Data Flow into (“Network Connection”) Asset-to-Service-Option: SO\_BVOIP\_ENABLED]” can be executed for the first time.

The execution of this process requires the maintenance of a change tracking event into GDB.CHANGE\_TRACKING, CHANGE\_SYSTEM and CHANGE\_USER. The change tracking event for the service delivery data flow must use CHANGE\_SYSTEM.name = ’INITIAL\_LOAD’ and CHANGE\_USER.name = ’ INITIAL\_LOAD’. It is sufficient to have one change tracking event for all affected records during a single execution of this process. (It is also an option to work with the “standard” CHANGE\_USER/CHANGE\_SYSTEM settings.)

NOTE that this requirement is a “one-time” load.

The set of assets for which this initial load applies is as follows  
(all “network connections” loaded from ICORE):

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| GDB.ASSET\_EXT\_NETWORK\_CONNECTION.id\_asset | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id\_meta\_table | METADATA.META\_TABLE.id |
| METADATA.META\_SYSTEM.id | METADATA.META\_TABLE.id\_meta\_system |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| METADATA.META\_SYSTEM | NAME | ICORE |

For the initial load the logic from requirement “HLD-254035c- GCP-FLOW-ICORE-325 [ICORE Data Flow into (“Network Connection”) Asset-to-Service-Option: SO\_BVOIP\_ENABLED]” has to be used.

<END 254035c.CR-106926>

### Service Delivery Data Flow into Golden Database [Icore Data Flow] for Static Initial Load for 259118

<BEGIN 259118>

#### HLD-259118- GCP-FLOW-ICORE-330 [ICORE Data Flow into (“Network Connection”) <STATIC INITIAL LOAD>]

This requirement is for doing an initial load for the PVC\_ID\_IDENTIFIER identifier data of a “NetworkConnection” asset.

This initial load needs to have been completed before the changes for 259118 for the requirements   
“HLD-254035-GCP-FLOW-ICORE-104-A [Icore Data Flow into Asset PVC (Layer-3 “Network Connection”)]”,  
“HLD-254035-GCP-FLOW-ICORE-104-B [Icore Data Flow into Asset PVC (Layer-2 "Network Connection")]” and  
“HLD-254035-GCP-FLOW-ICORE-104-C [Icore Data Flow into Asset PVC (“Internet VLAN” “Network Connection”)]”   
can be executed for the first time.

The execution of this process requires the maintenance of a change tracking event into GDB.CHANGE\_TRACKING, CHANGE\_SYSTEM and CHANGE\_USER. The change tracking event for the service delivery data flow must use CHANGE\_SYSTEM.name = ’INITIAL\_LOAD’ and CHANGE\_USER.name = ’ INITIAL\_LOAD’. It is sufficient to have one change tracking event for all affected records during a single execution of this process. (It is also an option to work with the “standard” CHANGE\_USER/CHANGE\_SYSTEM settings.)

NOTE that this requirement is a “one-time” load.

The set of assets for which this initial load applies is as follows  
(all “network connections” loaded from ICORE):

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| GDB.ASSET\_EXT\_NETWORK\_CONNECTION.id\_asset | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id\_meta\_table | METADATA.META\_TABLE.id |
| METADATA.META\_SYSTEM.id | METADATA.META\_TABLE.id\_meta\_system |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| METADATA.META\_SYSTEM | NAME | ICORE |

For the initial load - for the set of assets as identified above - the logic is as follows:

* Create an IDENTIFIER\_TYPE and IDENTIFIER\_VALUE record (as described e.g under “HLD-254035-GCP-FLOW-ICORE-104-A [Icore Data Flow into Asset PVC (Layer-3 “Network Connection”)]”, tagged by the project number 259118) and use the value of “GDB.asset\_ext\_network\_connection.network\_connection\_id” for “GDB.ASSET\_IDENTIFIER\_VALUE.value”.

<END 259118>

<BEGIN 259118>

#### HLD-259118- GCP-FLOW-ICORE-331 [ICORE Data Flow into (“Customer Network”) <STATIC INITIAL LOAD>]

This requirement is for doing an initial load for the new “ID\_NETWORK\_SUB\_TYPE” column of GDB.ASSET\_EXT\_CUSTOMER\_NETWORK.

This initial load needs to have been completed before the changes for 259118 for the requirements   
“HLD-254035-GCP-FLOW-ICORE-103-A [Icore Data Flow into Asset VPN (Layer-3 “Customer Network”)]”,  
“HLD-254035-GCP-FLOW-ICORE-103-B [Icore Data Flow into Asset PVC (as VPN) (Layer-2 "Customer Network")]” and  
“HLD-254035-GCP-FLOW-ICORE-103-C [Icore Data Flow into Asset VPN (“Internet VLAN” “Customer Network”)]”   
can be executed for the first time.

The execution of this process requires the maintenance of a change tracking event into GDB.CHANGE\_TRACKING, CHANGE\_SYSTEM and CHANGE\_USER. The change tracking event for the service delivery data flow must use CHANGE\_SYSTEM.name = ’INITIAL\_LOAD’ and CHANGE\_USER.name = ’ INITIAL\_LOAD’. It is sufficient to have one change tracking event for all affected records during a single execution of this process.

NOTE that this requirement is a “one-time” load.

The set of assets for which this initial load applies is as follows  
(all “customer networks” loaded from ICORE):

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| GDB.ASSET\_EXT\_NETWORK\_CONNECTION.id\_asset | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id\_meta\_table | METADATA.META\_TABLE.id |
| METADATA.META\_SYSTEM.id | METADATA.META\_TABLE.id\_meta\_system |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| METADATA.META\_SYSTEM | NAME | ICORE |

For the initial load - for the set of assets as identified above - the logic is as follows:

* Use the same logic as is used under   
  “HLD-254035-GCP-FLOW-ICORE-103-A [Icore Data Flow into Asset VPN (Layer-3 “Customer Network”)]”,  
  “HLD-254035-GCP-FLOW-ICORE-103-B [Icore Data Flow into Asset PVC (as VPN) (Layer-2 "Customer Network")]” and  
  “HLD-254035-GCP-FLOW-ICORE-103-C [Icore Data Flow into Asset VPN (“Internet VLAN” “Customer Network”)]”   
  tagged by the project number 259118, to initialize the value of the new column “id\_network\_sub\_type”.

<END 259118>

### Service Delivery Data Flow into Golden Database [Icore Data Flow] for Static Initial Load for 258491

<BEGIN 258491>

#### HLD-258491- GCP-FLOW-ICORE-340 [ICORE Data Flow into Asset (“Customer Network for Network Connections”(Layer-3, MVL)) to Asset] <STATIC INITIAL LOAD>]

This requirement is doing an initial load for the relationship between a layer-3 customer network asset and its associated layer-3 MVL network connections.

This initial load needs to have been completed before the changes for 258491 for the requirements   
“HLD-254035-GCP-FLOW-ICORE-206-A [Icore Data Flow into Asset (“Customer Network for Network Connections”(Layer-3)) to Asset]”   
and  
“HLD-258491-GCP-FLOW-ICORE-206-A [Icore Data Flow into Asset (“Customer Network for Network Connections”(Layer-3, MVL)) to Asset]”   
can be executed for the first time.

The execution of this process requires the maintenance of a change tracking event into GDB.CHANGE\_TRACKING, CHANGE\_SYSTEM and CHANGE\_USER. The change tracking event for the service delivery data flow must use CHANGE\_SYSTEM.name = ’INITIAL\_LOAD’ and CHANGE\_USER.name = ’ INITIAL\_LOAD’. It is sufficient to have one change tracking event for all affected records during a single execution of this process. (It is also an option to work with the “standard” CHANGE\_USER/CHANGE\_SYSTEM settings.)

NOTE that this requirement is a “one-time” load.

The set of assets for which this initial load applies is as follows  
(at the end of the day it boils down to associations between a layer-3 customer network asset and its associated layer-3 MVL network connections where the associated MVL “network connections” have been loaded from ICORE):

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| GDB.ASSET\_EXT\_NETWORK\_CONNECTION.id\_asset | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id | GDB.ASSET.id |
| METADATA.SOURCE\_KEY.id\_meta\_table | METADATA.META\_TABLE.id |
| METADATA.META\_SYSTEM.id | METADATA.META\_TABLE.id\_meta\_system |

Database constraints:

|  |  |  |
| --- | --- | --- |
| **Table name** | **Column name** | **Constraint(s)** |
| METADATA.META\_SYSTEM | NAME | ICORE |

For the initial load - for the set of assets as identified above - the **high-level logic** is as follows:

* **(1)** From the set of ICORE network connections as identified above find the ones that are MVL PVCs (network connections).
* **(2)** For that new set of ICORE MVL PVCs (network connections) find their associated VPNs (customer networks) as associated via the standard ICORE association (via the IPFR table).
* **(3)** For each ICORE MVL PVC (network connection) and its associated VPN (customer network) drop the corresponding association from GDB.
* **(4)** For each ICORE MVL PVC (network connection) create its association(s) with the corresponding MVL VPNs (customer networks) according to the logic as given under requirement “HLD-258491-GCP-FLOW-ICORE-206-A [Icore Data Flow into Asset (“Customer Network for Network Connections”(Layer-3, MVL)) to Asset]”.

The **detailed logic for the steps (1) to (3)** is given below.

(Item (4) above already contains enough details using the reference given !)

**(1)**

Using the “GDB.ASSET\_EXT\_NETWORK\_CONNECTION.id\_asset” (= GDB.ASSET.id) value, do a reverse EKT lookup to find the ICORE PVC ID (= “NETWORK\_CONNECTION.pvc\_id” ).  
For that ICORE PVC ID check whether the corr. PVC\_EXTENSION record (join via “PVC\_ID”) has a value of MVL = ‘Y’. If that is the case then the ICORE PVC ID is an MVL one and needs to be processed in the following steps.

**(2)**

For the MVL ICORE PVC IDs as identified under “(1)” find the ICORE VPN ID using this:

|  |  |  |
| --- | --- | --- |
| NETWORK\_CONNECTION.pvc\_id =  PVC.pvc\_id = IPFR.pvc\_id  <and>  IPFR.vpn\_id = VPN.vpn\_id  (= VPN.vpn\_id) | GDB.ASSET.id [via Enterprise Key Translation for VPN.vpn\_id] | Use SOURCE\_KEY.id which populated the respective ASSET.id |

**(3)**

For the pairs of   
GDB.ASSET.id [NETWORK\_CONNECTION.pvc\_id[MVL]] and   
GDB.ASSET.id [VPN.vpn\_id]

find all records in GDB.ASSOCIATION with

Id\_object\_what = GDB.ASSET.id [NETWORK\_CONNECTION.pvc\_id[MVL]] and

Id\_object\_to = GDB.ASSET.id [VPN.vpn\_id].

Drop each record found from GDB.ASSOCIATION.

<END 258491>

### Service Delivery Data Flow into Golden Database [Icore Data Flow] for Contact (OPT-E-WAN)

<284465c-CR158090>

For project 284465c CR158090, ICORE location contact data needs to be stored into GDB Contact database for OPT-E-WAN service.

#### HLD-284465c-CR158090-GCP-FLOW-ICORE-001 [Load Contact Data into Golden Database Process]

Load the following into the ADDRESS\_NOTATION table – if SITE\_CONTACT.country, cont\_city and cont\_address is not null

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB ADDRESS\_NOTATION column** | **Processing comment** |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘ADDRESS\_NOTATION’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| - | id\_address\_notation\_type | Referencing ADDRESS\_NOTATION\_TYPE ‘INVENTORY\_ADDRESS\_REPRESENTATION’ |
| SITE\_CONTACT.cont\_id | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| SITE\_CONTACT.country | Country\_code | *Try to make sure the country data is converted to 3-character ISO country code data* |
| SITE\_CONTACT.cont\_state | Subdivision |  |
| SITE\_CONTACT.cont\_city | City |  |
| SITE\_CONTACT.cont\_address | Address\_line1 |  |
| - | Address\_line2 | Set to NULL |
| - | Address\_line3 | Set to NULL |
| SITE\_CONTACT.cont\_zip | Postal\_code |  |
| - | Clli |  |

Load the following into the LOCATION\_NOTATION table – if an ADDRESS\_NOTATION record is created above and SITE\_CONTACT.cont\_room is not null

|  |  |  |
| --- | --- | --- |
| **Icore table.column** | **GDB LOCATION\_NOTATION column** | **Processing comment** |
| - | id\_object\_type | Referencing OBJECT\_TYPE ‘LOCATION\_NOTATION’ |
| - | id\_change\_tracking | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| - | is\_read\_only | ‘Y’ |
| ADDRESS\_NOTATION.id | id\_address\_notation |  |
| SITE\_CONTACT.cont\_id | id [via Enterprise Key Translation] | Use SOURCE\_KEY.id into GDB |
| SITE\_CONTACT.cont\_room | Room |  |
| - | Floor |  |
| - | Building |  |
| - | suite |  |

Load the following location contact data into GDB. Do not load the contact if the id\_organization cannot be found via the joins and EKT translation.

|  |  |  |  |
| --- | --- | --- | --- |
| **ICORE SITE\_CONTACT** | **GDB.CONTACT** | **Datatype** | **Comments** |
| - | Id\_object\_type | Number(10) | Referencing OBJECT\_TYPE ‘CONTACT’ |
| - | Id\_change\_tracking | Number(20) | See HLD-254035-GCP-FLOW-ICORE-001 [Icore Data Flow into Golden Database Process] |
| SITE\_CONTACT.cont\_id | Id [via Enterprise Key Translation] | Number(20) | New primary key from Enterprise Key Translation. |
| SITE\_CONTACT.first\_name | First\_name | VarChar2(100) | NULL |
| SITE\_CONTACT.last\_name | Last\_name | VarChar2(100) | NOT NULL |
| CUSTOMER.cust\_id | Id\_organization [via Enterprise Key Translation for CUSTOMER.cust\_id] | Number(20) | SITE\_CONTACT.site\_id = SITE.site\_id and  SITE.cust\_id = CUSTOMER.cust\_id |
| LOCATION\_NOTATION.id (if created above) | Id\_location\_notation | Number(20) | NULL |
| ADDRESS\_NOTATION.id (if created above) | Id\_address\_notation | Number(20) |  |
| - | Middle\_name | VarChar2(100) | NULL |
| - | Preferred\_language | VarChar2(50) | NULL |
| - | Salutation | VarChar2(50) | NULL |
| - | Job\_title | VarChar2(100) | NULL |
| - | Time\_zone | VarChar2(50) | NULL |
| - | Last\_validation\_date | Date | NULL |
| - | Is\_verification\_opted \_out | Char(1) | NULL |
| - | Is\_read\_only | Char(1) | ‘N’ |
| - | Id\_contact\_unified | Number(20) | NULL |

|  |  |  |  |
| --- | --- | --- | --- |
| **ICORE SITE\_CONTACT** | **GDB.PHONE** | **Datatype** | **Comments** |
| - | Id\_object\_type | Number(10) | Referencing OBJECT\_TYPE ‘CONTACT’ |
| - | Id\_change\_tracking | Number(20) | NOT NULL |
| - | Id | Number(10) | New primary key from sequence Phone\_id |
| - | Id\_object | Number(20) | Contact\_id EKT |
| SITE\_CONTACT.cont\_phone | Phone\_number | VarChar2(100) | NOT NULL |
| - | Prompts | VarChar2(100) | NULL |
| SITE\_CONTACT.cont\_ext | Extension | VarChar2(50) | NULL |
| - | Provider | VarChar2(100) | NULL |
| - | Type | ‘Work Phone’ | NOT NULL |
| - | Is\_preferred | Char(1) | NULL |
| - | Capability | VarChar2(100) | NULL |

|  |  |  |  |
| --- | --- | --- | --- |
| **ICORE SITE\_CONTACT** | **GDB.PHONE** | **Datatype** | **Comments** |
| - | Id\_object\_type | Number(10) | Referencing OBJECT\_TYPE ‘CONTACT’ |
| - | Id\_change\_tracking | Number(20) | NOT NULL |
| - | Id | Number(10) | New primary key from sequence Phone\_id |
| - | Id\_object | Number(20) | Contact\_id EKT |
| SITE\_CONTACT.cell\_phone | Phone\_number | VarChar2(100) | NOT NULL |
| - | Prompts | VarChar2(100) | NULL |
| - | Extension | VarChar2(50) | NULL |
| - | Provider | VarChar2(100) | NULL |
| - | Type | ‘Work Cell’ | NOT NULL |
| - | Is\_preferred | Char(1) | NULL |
| - | Capability | VarChar2(100) | NULL |

|  |  |  |  |
| --- | --- | --- | --- |
| **ICORE SITE\_CONTACT** | **GDB.PAGER** | **Datatype** | **Comments** |
| - | Id\_object\_type | Number(10) | Referencing OBJECT\_TYPE ‘CONTACT’ |
| - | Id\_change\_tracking | Number(20) | NOT NULL |
| - | Id | Number(10) | New primary key from sequence Pager\_id |
| - | Id\_object | Number(20) | Contact\_id EKT |
| SITE\_CONTACT.cont\_beeper | Pager\_number | VarChar2(100) | NOT NULL |
| SITE\_CONTACT.cont\_beeper\_pin | Pin | VarChar2(50) | NULL |
| - | Provider | VarChar2(100) | NULL |
| - | Type | ‘Work Pager’ | NOT NULL |
| - | Is\_preferred | Char(1) | NULL |
| - | Capability | VarChar2(100) | NULL |

|  |  |  |  |
| --- | --- | --- | --- |
| **ICORE SITE\_CONTACT** | **GDB.FAX** | **Datatype** | **Comments** |
| - | Id\_object\_type | Number(10) | Referencing OBJECT\_TYPE ‘CONTACT’ |
| - | Id\_change\_tracking | Number(20) | NOT NULL |
| - | Id | Number(10) | New primary key from sequence Fax\_id |
| - | Id\_object | Number(20) | Contact\_id EKT |
| SITE\_CONTACT.cont\_fax | Fax\_number | VarChar2(100) | NOT NULL |
| - | Extension | VarChar2(50) | NULL |
| - | Type | ‘Work Fax’ | NOT NULL |
| - | Is\_preferred | Char(1) | NULL |
| - | Capability | VarChar2(100) | NULL |

|  |  |  |  |
| --- | --- | --- | --- |
| **ICORE SITE\_CONTACT** | **GDB.EMAIL** | **Datatype** | **Comments** |
| - | Id\_object\_type | Number(10) | Referencing OBJECT\_TYPE ‘CONTACT’ |
| - | Id\_change\_tracking | Number(20) | NOT NULL |
| - | Id | Number(10) | New primary key from sequence Email\_id |
| - | Id\_object | Number(20) | Contact\_id EKT |
| SITE\_CONTACT.cont\_email | Email\_address | VarChar2(100) | NOT NULL |
| - | Type | ‘Work Email’ | NOT NULL |
| - | Is\_preferred | Char(1) | NULL |
| - | Capability | VarChar2(100) | NULL |

For all the records above – use the following Database traversals & Constraints:

Database traversals:

|  |  |
| --- | --- |
| **From table.column** | **To table.column** |
| SITE\_CONTACT.site\_id | SERVICE\_ASGMT.site\_id |
| SERVICE\_ASGMT.serv\_opt\_id | SERVICE\_OPTION.serv\_opt\_id |
| SERVICE\_OPTION.serv\_id | SERVICE.serv\_id |
| Trim(SERVICE.serv\_name) | ‘OPT-E-WAN’ |

#### HLD-284465c-CR158090-GCP-FLOW-ICORE -002 [ ASSOCIATION CONTACT -> SITE ]

Implement the following:

Process the following ICORE records into the GDB contact to site association:

|  |  |  |
| --- | --- | --- |
| **ICORE column** | **GDB column ASSOCIATION** | **Processing comment** |
| - | Id | New primary key from sequence ASSOCIATION\_ID |
| - | id\_change\_tracking | See HLD-254035a-GCP-FLOW-IDIS-001 [IDIS Data Flow into Golden Database Process] |
| - | Is\_read\_only | ‘N’ |
| - | Id\_association\_type | Referencing ASSOCIATION\_TYPE.id which represents:  OBJECT\_TYPE\_WHAT ‘**CONTACT**’, OBJECT\_TYPE\_TO ‘SITE’. Use FUNCTION\_TYPE = ‘LOCAL\_SITE’. User FUNCTION\_ROLE = ‘PRIMARY’ if SITE\_CONTACT.cont\_primary = ‘Y’, else ‘ALTERNATE’ |
| SITE\_CONTACT.cont\_id | Id\_object\_what [via Enterprise Key Translation] | Unification master should be used if available |
| SITE.site\_id | Id\_object\_to [via Enterprise Key Translation] | SITE\_CONTACT.site\_id = SITE.site\_id |
|  | Id\_role\_notation | Foreign Key to ROLE\_NOTATION.ID  Referencing ROLE\_NOTATION.Name for Contact\_type, Object\_type\_what: ‘CONTACT’,  Object\_type\_to: ‘SITE’ and ROLE\_NOTATION.NAME = ‘Site Contact’  Add a new ROLE\_NOTATION record if the described ROLE\_NOTATION record is not already existing. |
| SITE\_CONTACT.cont\_id | Id\_source\_key | Use SOURCE\_KEY.id into GDB |
| - | Gdb\_internal\_flags | NULL |

## Appendix: Circuit Id Formats

This appendix section contains data about circuit id formats as found in ECDB and ICORE.

(See also “US Domestic Circuit ID Format” above)

**PA510-2:**

For new ports added to the **provg\_port** table use the below logic to populate **provg\_port.port\_ckt\_clci\_icore\_format**.

1 If **provg\_port.int\_indicator** = ‘Y’,

Then populate **provg\_port.port\_ckt\_clci\_icore\_format** = **provg\_port.port\_ckt\_clci**

1. If **provg\_port.int\_indicator** = ‘N’,

Then populate **provg\_port.port\_ckt\_clci\_icore\_format** using the logic below

The **provg\_port.port\_ckt\_clci** where **provg\_port.int\_indicator** = ‘N’ will be in the format as shown below:

|  |  |  |
| --- | --- | --- |
| **Character Position** | **Name** | **Data Type** |
| 1-2 | Prefix | Char(2) |
| 3-4 | Service Code | Char(2) |
| 5-6 | Modifier | Char(2) |
| 7-12 | Serial Number | Number(6) |
| 13-15 | Suffix | Char(3) |
| 16-19 | Company Code | Char(4) |

The reformatted **port\_ckt\_clci\_icore\_format** is shown below.

|  |  |  |
| --- | --- | --- |
| **Character Position** | **Name** | **Data Type** |
| 1-2 | Prefix | Char(2) |
| 3 | ‘.’ | Char(1) |
| 4-5 | Service Code | Char(2) |
| 6-7 | Modifier | Char(2) |
| 8 | ‘.’ | Char(1) |
| 9-14 | Serial Number | Number(6) |
| 15 | ‘.’ | Char(1) |
| 16-18 | Suffix | Char(3) |
| 19 | ‘.’ | Char(1) |
| 20-23 | Company Code | Char(4) |

Fields must be separated by mandatory delimiters (.). All the blanks (including leading blanks, trailing blanks, and blanks in between) should be removed from the **provg\_port.port\_ckt\_clci** column to convert it to ICORE format.

The table below shows sample circuits in **provg\_port.port\_ckt\_clci** and **provg\_port.port\_ckt\_clci\_icore\_format**.

|  |  |
| --- | --- |
| **PORT\_CKT\_CLCI** | **PORT\_CKT\_CLCI\_ICORE\_format** |
| ‘QWDHEC123456100ATI ‘ | QW.DHEC.12345.100.ATI |
| ‘ DHEC 123 ATI ‘ | DHEC.123..ATI |
| ‘ DHEC 1234100ATI ‘ | .DHEC.1234.100.ATI |

**Note:** ICORE format of **provg\_port.port\_ckt\_clci** is not a fixed length format. The ‘.’ delimiter between Prefix and Service Code/Modifier is optional in ICORE database, if the prefix is blank. Queries to ICORE database should take this into consideration when searching based on the **port\_ckt\_clci\_icore\_format**

## Appendix: Copies of references to other HLDs

This appendix section contains copies of references to other HLDs.

At the moment no such copies are given.

## Appendix: USRP constant value mapping



## Alternative Designs

No alternative designs are being considered at this time.

## Assumptions/Risks

**Assumptions:**

1. Production support team (EMAS, PSO, DBA) will apply the necessary system and database setup for mission crititical systems.
2. Due to changes in identified systems between the Solution Approach, the System Requirements and the HLD it is acceptable to switch from near real-time processing to batch processing for the service delivery to Golden Database flow.
3. Near real-time delete and immediate delete in batch processing potentially removes necessary data in error, if the upstream system executes any type of delete&insert for the same record.  
   It is assumed that GCP shall implement a safety net and delay deletes identified in upstream (service delivery) systems before they are applied to the Golden Database inventory.
4. The Golden DB is supposed to represent the inventory hierarchy as enabled via Service Delivery processes into the databases in scope.
5. The baselined System Requirements document is the only source for requirements, i.e. discussion points from meetings and email chains that are not represented in the SR document are obsolete.

**Risks:**

1. The engagement of CSI for exclusively providing web service access to data contained in the GDB may significantly delay this project.

## Traceability Matrix

**{@TBD}**

|  |  |
| --- | --- |
| Requirement ID | Design Element Identifier |
| FR-1.1 | HLD - 1 |

## Pre-Production Disaster Recovery Planning

All relevant pre-production disaster recovery policies, standards and documentation have been reviewed and no changes are needed. A copy of the current GCP disaster recovery plan (GCP ARM (Application Recovery Manual)) is available at the following P8 link:

[Manual](https://cps.web.att.com/CPSWorkplace/getContent?id=current&vsId=%7BBF905B1E-9C53-4F48-BC36-64193179A0F7%7D&objectStoreName=IT-Enterprise.__.Systems.__.and.__.Software.__.Eng&objectType=document&guestID=servicesguest)

## Other Plans and References

**{@TBD}**

|  |
| --- |
|  |

## 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** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. parse var acc\_ckt one '.' two '.' three '.' four '.' five

   if five = '' then do /\* 3 dot format \*/

   tempa = ‘ ‘||one||right(strip(two,'L','0'),6,' ')||left(three,3)||four   
   /\* *‘DHEC.127819..ATI’ becomes ‘ DHEC127819 ATI’ Note: Two leading blanks* \*/

   end

   else do /\* 4 dot format \*/

   tempa = left(one,2)|| two || right(strip(three,'L','0'),6,' ')||left(four,3)||five /\*22.HXGM.010062..SUV becomes 22HXGM 10062 SUV *\*/*

   **end**

   **Sample: ‘*DHEC.127819..ATI’*** *becomes ‘* ***DHEC127819 ATI’ or***

   ***‘.DHEC.123817.800.ATI’ becomes ‘ DHEC123817800ATI’***

   **22.HXGM.010062..SUV** becomes **22HXGM 10062 SUV**

   **We need to ensure that links are possible regardless of the icore format used.**

   (see “Appendix: Circuit Id Formats”) [↑](#footnote-ref-1)