

DS.140 Design Specification

Emirates Telecommunication Group Company PJSC

**HCM To EBS PTE Reverse Termination Integration**

Author: Sugavanan

Creation Date: August 25, 202020

Last Updated: August 25, 202020

Document Ref: <Document Reference Number>

Version: DRAFT 1A

1. **Title, Subject, Last Updated Date, Reference Number**, **and** **Version** are marked by a Word Bookmark so that they can be easily reproduced in the header and footer of documents. When you change any of these values, be careful not to accidentally delete the bookmark. **You can make bookmarks visible by selecting the Office Button>Word Options>Advanced> and checking the Show bookmarks option in the Show document content region.**

**Approvals:**

|  |  |
| --- | --- |
| <Approver 1> | Buttar Gurmeet Singh |
| <Approver 2> |  |

1. To add additional approval lines, press [Tab] from the last cell in the table above.

oracle.gif

1. You can delete any elements of this cover page that you do not need for your document.

# Document Control

## Change Record

5

| Date | Author | Version | Change Reference |
| --- | --- | --- | --- |
| 25-Aug-20 | Sugavanan | Draft 1a | No Previous Document |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Reviewers

| Name | Position |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

Contents

1 Document Control ii

1.1 Change Record ii

1.2 Reviewers ii

2 Technical Overview 1

2.1 Integrations 1

2.2 Block Relationship Diagram 1

3 Integration Design 2

3.1 HCM to EBS PTE New Hire Integration 2

4 Data Design 5

4.1 Tables 5

4.2 Validation Logic 5

5 SQL Design 6

5.1 SQL Statements 6

6 Behavior Design 7

6.1 Function (Operation) Design 7

6.2 Business Rule Design 7

7 Database Design 8

7.1 Database Diagram 8

7.2 Desired Table Changes 8

7.3 Tables, Indexes, Sequences 8

7.4 Archiving 8

8 Open and Closed Issues 9

8.1 Open Issues 9

8.2 Closed Issues 9

1. To update the table of contents, put the cursor anywhere in the table and press [F9]. To change the number of levels displayed, select the menu option Insert‑>Index and Tables, make sure the Table of Contents tab is active, and change the Number of Levels to a new value.

# Technical Overview

1. The purpose of this task is to assemble all the information that is required to describe the design of a software component into a complete Design Specification. This task is not a substitute for executing the individual design tasks. This specification work product can serve as a structure for completing the design for each component by providing pointers back into the Design Tasks:   
   - DS.040 Develop Design Architecture Description  
   - DS.080 Design Software Components  
   - DS.090 Design Data  
   - DS.100 Design Behavior  
   - DS.130 Design User Interface

This Design Specification documents the detailed design for HCM to EBS PTE Person Integrations. This specification, the design specifications for the other components that are part of this use-case package (package), along with the Analysis Specification for the package constitute the complete detailed design for this use case package.

## Integrations

1. The intent of this section is to list the building blocks that are required to design the designated component. This includes classes, objects, modules, etc. Reference the Module View of the Architecture Description (RD.130) and appropriate Software Component Design (DS.080) to derive the list of classes and their relationships.

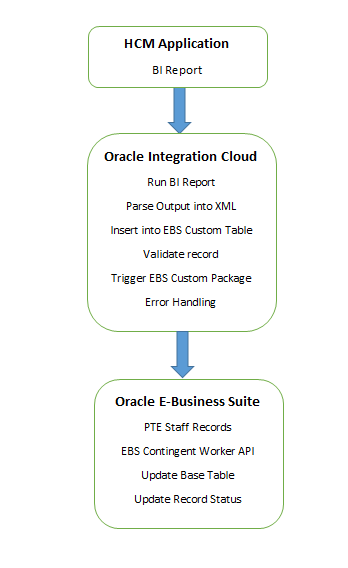
Integration Names

* HCM To EBS PTE Reverse Termination

## Block Relationship Diagram

1. The intent of this section is to graphically depict how the component under consideration interfaces to related components, external systems, and other actors that interact with the use-case package. Reference the Conceptual View and Module View of the Architecture Description (RD.130) and the class diagram prepared in the Software Component Design (DS.080) and Component Behavior Design (DS.100).

The diagram below represents the flow of the data from HCM to EBS for the above mentioned integration.



1. The diagram above is just an example—Double click on it to edit it as an embedded object. Select it and choose Edit->VISIO Object->Open to open it in the full Visio application.  
     
   You can also use other tools to draw your diagram and define most of the information in this section. So you may then want to refer to the repository for a specific diagram.

# Integration Design

## HCM To EBS PTE Reverse Termination Integration

HCM To EBS PTE Reverse Termination integration update contingent employees in EBS from HCM.

**Integration Flow:**

1. The scheduled integration runs the PTE staffs reverse termination report in HCM and picks staff record based on current Last Update date by checking against integration date parameter.
2. In loop the integration inserts fetched staff records into EBS custom table with mode as ‘REVTERM’
3. Once insert operation is done, Integration Triggers the Reverse Termination procedure in EBS custom Package XXHCM\_PTE\_UTIL\_PKG
4. This Procedure picks records whose status is null or “ERROR” in custom table and It proceeds for end placement process for those PTE staff s through API.
5. The API reverses the contingent workers termination in EBS. Once reversed, the person staff record is updated back to custom table with respective status and error message.
6. Assigns the current date and time into a variable so as to use it in the next schedule run as parameter

### Reports

|  |  |  |
| --- | --- | --- |
| **Report Name** | **Report Location** | **Description** |
| HCM\_PTE\_REV\_TER\_UPDATE\_REPORT.xdo | /Shared Folders  /Custom  /Human Capital Management  /Employee Integration  /EBS Person Integration | Pulls new Reverse Terminated records from HCM database based on current date |

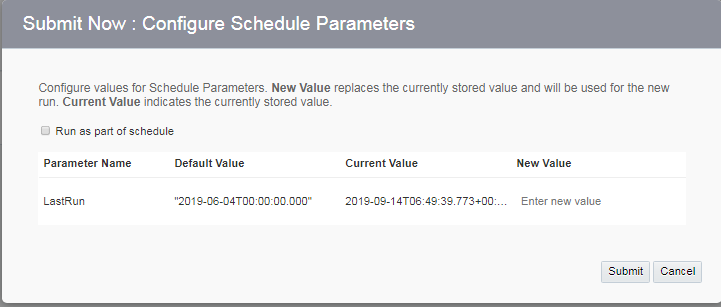
### Web Service Names

| **Service Name** | **Method** | **Comments** |
| --- | --- | --- |
| <https://etcprodjcs-cecws161491.jcs.ocp.oraclecloud.com/OICRawDataToXML/jersey/Service/generateXML> | POST | Converts report output to XML |

**Integration Parameters:**

LastRun date Parameter is automatically passed on to the integration while submitting. This value comes from previous run.

For Adhoc basis run, parameter to be passed in format: **YYYY-MM-DDT00:00:00.000**



### Integration Adapters

| **Adapter Name** | **Type** | **Description** |
| --- | --- | --- |
| BI Publisher Report Service | SOAP | To run HCM reports in OIC |
| OICRawDataToXML | REST | For parsing the data from Reports and converts to XML |
| Oracle Database Adapter | Oracle Database | For connecting to EBS database |
|  |  |  |
|  |  |  |

### EBS Database Packages Involved

* XXHCM\_PTE\_UTIL\_PKG.INSERT\_CWK\_ATOMFEED
* XXHCM\_PTE\_UTIL\_PKG.PTE\_REVERSE\_TERMINATE

### Interface Frequency and Schedule

Immediate (real-time)

On demand

Hourly Day: \_\_\_\_\_\_\_\_\_\_\_ Time: \_\_\_\_\_\_\_\_\_\_\_ Push/Pull: \_\_\_\_\_\_\_\_\_\_\_

Daily Day: \_\_\_\_\_\_\_\_\_\_\_ Time: \_\_\_\_\_\_\_\_\_\_\_ Push/Pull: \_\_\_\_\_\_\_\_\_\_\_

Weekly Day: \_\_\_\_\_\_\_\_\_\_\_ Time: \_\_\_\_\_\_\_\_\_\_\_ Push/Pull: \_\_\_\_\_\_\_\_\_\_\_

Monthly Day: \_\_\_\_\_\_\_\_\_\_\_ Time: \_\_\_\_\_\_\_\_\_\_\_ Push/Pull: \_\_\_\_\_\_\_\_\_\_\_

Quarterly Day: \_\_\_\_\_\_\_\_\_\_\_ Time: \_\_\_\_\_\_\_\_\_\_\_ Push/Pull: \_\_\_\_\_\_\_\_\_\_\_

Annually Day: \_\_\_\_\_\_\_\_\_\_\_ Time: \_\_\_\_\_\_\_\_\_\_\_ Push/Pull: \_\_\_\_\_\_\_\_\_\_\_

Other Day: \_\_\_\_\_\_\_\_\_\_\_ Time: \_\_\_\_\_\_\_\_\_\_\_ Push/Pull: \_\_\_\_\_\_\_\_\_\_\_

1. The intent of this section is to document the report format designs for the reports provided by the component. Reference or include the report specification from the User Interface Design (DS.130) for the component.

# Data Design

1. The intent of this section is to define the design details of each attribute – format, length, accessibility, visibility, validation rules, mandatory or optional, etc. – that is required for each entity or class within the component. If available, reference or include the class diagram (or other data design model) from the Component Data Design (DS.090) for this component. If not available, use the table below.

## Tables

1. The intent of this section is to identify the table, columns, and source values that are required to support the above data elements. Refer to the Physical Database Design (IM.040), to identify the existing tables where the above attributes are located.

|  |
| --- |
| Table |
|  |
| **XXHCM\_PTE\_INTERFACE** |
|  |
|  |
|  |
|  |

## Validation Logic

1. The intent is to design the rules that are necessary to verify the format, length, relationships etc. of the attributes listed above.

### Table Name

PTE staff reverse termination process follows the unique constraint validation logic to insert into the custom table XXHCM\_PTE\_INTERFACE based on the below listed column value mapping.

| Attribute Name | Value Meaning | Description |
| --- | --- | --- |
| INSERT\_UPDATE\_MODE | ‘REVTERM’ | Default defined value for Reverse Termination Mode |
| PERSON\_NUMBER | PTE staff Number | PTE Staff New Number from HCM system |
| EFFECTIVE\_START\_DATE | Staff Hire Date | Staff Original Join Date |
| WORKER\_TYPE | ‘CWK’ | Staff Person Record Type |
| WORK\_EMAIL | Reverse Termination Action ID | Action Occurrence Unique ID from HCM |
|  |  |  |

# SQL Design

1. The intent of this section is to define the tables, the attributes and the SQL statements that are necessary to create, read, update, and/or delete the attributes for each use case from the database. Include applicable portions of the Software Component Design (DS.080). The focus is on creating the SQL documents for the CRUD (Create, Read, Update, Delete) processes. If the DS.080 is not available, provide the SQL for your module here.

## SQL Statements

|  |
| --- |
| **HCM\_PTE\_REV\_TER\_UPDATE\_DM Report Query** |
| SELECT DISTINCT  PERSONPEO.PERSON\_ID,  PERSONDETAILSPEO.PERSON\_NUMBER,  PERSONNAMEPEO.DISPLAY\_NAME,  PERSONNAMEPEO.FIRST\_NAME,  PERSONNAMEPEO.LAST\_NAME,  ASSIGNMENTPEO.ASSIGNMENT\_NUMBER,  PS.DATE\_START HIRE\_DATE ,  PS.LAST\_UPDATE\_DATE ,  PS.ACTUAL\_TERMINATION\_DATE ,  PERSON\_TYPE.SYSTEM\_PERSON\_TYPE ,  PS.ACTION\_OCCURRENCE\_ID  FROM PER\_PERSONS PERSONPEO,  PER\_ALL\_PEOPLE\_F PERSONDETAILSPEO,  PER\_PERSON\_NAMES\_F\_V PERSONNAMEPEO,  PER\_ALL\_ASSIGNMENTS\_M ASSIGNMENTPEO,  PER\_PERSON\_TYPES\_VL PERSON\_TYPE,  PER\_PERIODS\_OF\_SERVICE PS  WHERE PERSONPEO.PERSON\_ID = PERSONDETAILSPEO.PERSON\_ID  AND PERSONPEO.PERSON\_ID = PERSONNAMEPEO.PERSON\_ID  AND PERSONPEO.PERSON\_ID = ASSIGNMENTPEO.PERSON\_ID  AND PERSONPEO.PERSON\_ID = PS.PERSON\_ID  AND ASSIGNMENTPEO.EFFECTIVE\_LATEST\_CHANGE = 'Y'  AND ASSIGNMENTPEO.ASSIGNMENT\_TYPE IN ('E', 'C')  AND ASSIGNMENTPEO.PERSON\_TYPE\_ID = PERSON\_TYPE.PERSON\_TYPE\_ID(+)  AND SYSDATE BETWEEN PERSONDETAILSPEO.EFFECTIVE\_START\_DATE AND NVL (PERSONDETAILSPEO.EFFECTIVE\_END\_DATE, SYSDATE)  AND SYSDATE BETWEEN PERSONNAMEPEO.EFFECTIVE\_START\_DATE AND NVL (PERSONNAMEPEO.EFFECTIVE\_END\_DATE, SYSDATE)  AND SYSDATE BETWEEN ASSIGNMENTPEO.EFFECTIVE\_START\_DATE AND NVL (ASSIGNMENTPEO.EFFECTIVE\_END\_DATE, SYSDATE)  AND PERSON\_TYPE.SYSTEM\_PERSON\_TYPE = 'CWK'  AND PS.ACTUAL\_TERMINATION\_DATE IS NULL  AND TRUNC( PS.CREATION\_DATE ) <> TRUNC( PS.LAST\_UPDATE\_DATE )  AND ( PS.LAST\_UPDATE\_DATE >= to\_date( :p\_date,'YYYY-MM-DD') )  AND EXISTS ( SELECT 1 FROM PER\_PERIODS\_OF\_SERVICE\_ S  WHERE S.PERSON\_ID = PS.PERSON\_ID  AND S.ACTUAL\_TERMINATION\_DATE IS NOT NULL  AND S.LAST\_UPDATE\_DATE < PS.LAST\_UPDATE\_DATE  AND S.LAST\_UPDATE\_DATE = ( SELECT MAX ( P.LAST\_UPDATE\_DATE ) FROM PER\_PERIODS\_OF\_SERVICE\_ P WHERE P.PERSON\_ID = PS.PERSON\_ID AND P.LAST\_UPDATE\_DATE < PS.LAST\_UPDATE\_DATE )  )  ORDER BY PS.LAST\_UPDATE\_DATE |

1. The intent of this section is to define the design considerations necessary to achieve the data retrieval and storage requirements for performance. Include performance requirements as specified in the Supplemental Requirements (RD.055) for service level requirements (i.e., 1-minute response time, etc.)

# Behavior Design

1. The intent of this section is to define the details of each operation (to include pseudo code) required for each entity, module, or class within the component. Refer to Behavior Design (DS.100) and the Design class diagram, with a focus specifically on the Operations section for the Class. In the event that you do not have a class diagram use the table below.

## Function (Operation) Design

| Function (operation) Name | Arguments | Default Values | Return Values | Pseudo code | Comments |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Business Rule Design

1. The intent of this section is to define the implementation strategy for each business rule within this component. Refer to the DS.110 Business Rules Design to capture the Business Rules for this component.
2. The intent of this section is to define the design considerations necessary to achieve the performance requirements for each interface. Include performance supplemental requirements as specified in the Supplemental Requirements (RD.055) work product for service level requirements include the Software Component Design (DS.080) with focus on the interface performance Design.

NA

# Database Design

1. The intent of this section is to design the physical schema of the database, or changes to the database for this component. Refer to the Logical Database Design (DS.150).

This section summarizes new and/or changing database objects and data required to support <Component Name>. However, the complete database design is documented in the Develop Database Design work product.

## Database Diagram

1. The intent of this section is to graphically depict the ER diagram that represents how the database will be designed for this component. Refer to the Logical Database Design (DS.150) with a focus only on the portion of the diagram needed to support this component.

NA

## Desired Table Changes

1. The intent of this section is to document the design changes necessary for existing or new tables required to support this component. Refer to the Logical Database Design (DS.150).

NA

## Tables, Indexes, Sequences

NA

## Archiving

1. The intent of this section is to document the design changes necessary to provide archiving required to support this component. Refer to the Logical Database Design (DS.150).

NA

# Open and Closed Issues

1. Add open issues that you identify while writing or reviewing this document to the open issues section. As you resolve issues, move them to the closed issues section and keep the issue ID the same. Include an explanation of the resolution.  
     
   When this work product is complete, any open issues should be transferred to the project- or process-level Issue Log (Manage focus area) and managed using a project level Issue Form (Manage focus area). In addition, the open items should remain in the open issues section of this work product, but flagged in the resolution column as being transferred.

## Open Issues

| ID | Issue | Resolution | Responsibility | Target Date | Impact Date |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Closed Issues

| ID | Issue | Resolution | Responsibility | Target Date | Impact Date |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |