**Legacy Bridge Replacement**

**Low Level Architecture**

**Version:** 0.4

**Status:** Draft

**Date:** 05/24/2017

**Prepared by:** John Tutton

**File name:** Legacy Bridge Replacement V 0\_4 Low Level Architecture.docx

|  |
| --- |
| **Working Document** |

**Version history**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Status** | **Author** | **Revisions** |
| 0.1 | 5/15/2017 | Draft | John Tutton | Initial document |
| 0.2 | 5/17/2017 | Draft | John Tutton | Update document to reflect meeting w/Ed |
| 0.3 | 05/24/2017 | Draft | John Tutton | Update document to reflect meeting w/Ed |
| 0.4 | 05/26/2017 | Draft | John Tutton | Update document to reflect meeting w/Ed |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Document Review Record**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Version | Status | Author | Reviewed by |
|  |  |  |  |  |
|  |  |  |  |  |

**Document Approval Record**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Version | Status | Role | Name |
|  |  |  |  |  |
|  |  |  |  |  |

**Document Distribution Record**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Status | Distributed To |
|  |  |  |  |
|  |  |  |  |

**Table of contents**

[1. Introduction 5](#_Toc482104879)

[1.1 Purpose of the document 5](#_Toc482104880)

[1.2 Intended Audience 5](#_Toc482104881)

[2. Project Overview 5](#_Toc482104882)

[2.1 Scope 5](#_Toc482104883)

[2.1.1 In Scope 5](#_Toc482104884)

[2.1.2 Out of Scope 5](#_Toc482104885)

[2.2 Related documents 6](#_Toc482104886)

[2.3 Assumptions 6](#_Toc482104887)

[2.4 Dependencies 6](#_Toc482104888)

[2.5 Constraints 6](#_Toc482104889)

[ERD Current State Overview 7](#_Toc482104890)

[2.6 Functional capabilities 7](#_Toc482104891)

[2.7 Architectural overview 7](#_Toc482104892)

[2.7.1 High-level integration overview 7](#_Toc482104893)

[2.7.2 Current Interface overview 8](#_Toc482104894)

[3. Legacy Bridge Proposed State Overview 9](#_Toc482104895)

[3.1 High-level logical overview of new Legacy Bridge 9](#_Toc482104896)

[3.1.1 Proposed architecture highlights 10](#_Toc482104897)

[3.2 Legacy Bridge Logical System overview 11](#_Toc482104898)

[3.2.1 High Level Logical highlights 12](#_Toc482104899)

[3.3 Legacy Bridge Workflows 13](#_Toc482104900)

[3.3.1 Legacy Bridge High-Level workflow 13](#_Toc482104901)

[3.3.2 High-level Overview for Event Monitor 14](#_Toc482104902)

[3.3.3 High-level Overview of the Data Importer/Data Transformation Process 16](#_Toc482104903)

[3.4 High-level Overview of the Export Service 18](#_Toc482104904)

[Appendix A – Business service classifications 21](#_Toc482104905)

[Definitions, Acronyms and Abbreviations 33](#_Toc482104906)

**Table of Figures**

[Figure 1: Overview of current HRMS and AS/400 landscape 7](#_Toc482104907)

[Figure 2: High-level overview of new Legacy Bridge 9](#_Toc482104908)

[Figure 3: High-level logical overview of the Legacy Bridge 11](#_Toc482104909)

[Figure 4: Legacy Bridge High-Level Workflow 13](#_Toc482104910)

[Figure 5: High-Level overview of the Event Monitor workflow 15](#_Toc482104911)

[Figure 6: High-Level overview of the Data Importer workflow 17](#_Toc482104912)

[Figure 7: High-Level overview of the Export Service 19](#_Toc482104913)

[Table 1: Definitions, Acronyms and Abbreviations 33](#_Toc482104914)

[Table 2: Service Definitions 34](#_Toc482104915)

# Introduction

## Purpose of the document

This document contains the high-level architecture for the implementation of a replacement bridge between the HRMS and the AS/400.

## Intended Audience

The audience for this document is everyone in Enterprise Holdings Inc. who manages, is responsible for, develops, or uses the existing PeopleSoft legacy bridge; including any third parties engaged to work on behalf of Enterprise whose remit includes management and/or associated services relating to the legacy bridge.

# Project Overview

The intention of this project is to provide infrastructure and services to decouple the current HRMS (PeopleSoft) and the legacy HR system (AS/400).

## Scope

### In Scope

The following items are in scope for this document:

* High-level system architecture for moving the legacy bridge off the current HRMS system and into a new Java Spring Boot application.
* High-level integration requirements suitable for business and development partners to allow initial high-level estimates to be produced.

### Out of Scope

The following sections in the requirements document have not been reviewed:

* Any changes to PeopleSoft not related to the legacy bridge move.
* Any changes to the AS/400 platform.

## Related documents

Legacy Bridge Replacement High Level Architecture

[LHRS High Level Architecture](https://confluence.ehi.com/display/ES/LHRS+High+Level+Architecture)

EID Generation Low Level Architecture

## Assumptions

The following assumptions are being made:

1. The existing system will continue to support employee, non-employee contingent, and multiple ID synchronization between the current HRMS and the AS/400 system.
2. For the duration of the remaining life of the AS/400 system it will receive updates from the new legacy bridge application.

## Dependencies

The following dependencies have been identified:

* PS Oracle database
* AS400

## Constraints

Development lifecycles for other systems – e.g. HR replacement project, AS400 migration, TempMast.

# ERD Current State Overview

## Functional capabilities

The existing legacy bridge moves data from the PeopleSoft for new/rehire events for employee/non-employee and is hosted inside the AS/400 platform. This EID is moved back to the HRMS system of record (PeopleSoft) before the employee/non-employee record is released to the ERD or other systems.

## Architectural overview

### High-level integration overview

Existing high level integrations are in place between PS and the AS/400 are shown below in Figure 1.



Figure - Overview of current HRMS and AS/400 landscape

### Current Interface overview

* Interface methods
  + AS/400
    - REXEC RPG method execution for New Hire/Rehire
* Live interfaces cover the following types of data:
  + Employee HR data including PII, EID
  + TempMast
    - Contingent employee
    - Multiple ID non-person accounts

# Legacy Bridge Proposed State Overview

## High-level logical overview of new Legacy Bridge

Below is the high-level overview of the replacement system.



Figure : High-level overview of new Legacy Bridge

### Proposed architecture highlights

The proposed system consists of two new Java Spring SQL monitors that watch the Employee and Contingent Employee/Multiple ID accounts. The HRMS system (PeopleSoft) will update the existing trigger tables and the events will be monitored by the external Java application. When events are detected they will be sent to the existing RGP methods on the AS/400 system.

The system consists of two monitors;

* Employee monitor
* Contingent/Non-Person (multiple ID) monitor

As changes happen in these trigger tables the application will read the data from the trigger and the cross-reference tables to determine if there is any work to perform. If they determine that an event that needs to be transferred to the AS/400 system has occurred, they will notify the Data Importer which will pull data directly from the PeopleSoft (PS) Database for the event. The Data Importer (DI) will wrap the data with metadata about the source of the event and pass the payload to the Data Export Engine (DPE). Data from the AS/400 channel will be sent to the AS/400 export service which will call the RPG methods using REXEC calls.

## Legacy Bridge Logical System overview

Below is the proposed systems high-level logical overview



Figure : High-level logical overview of the Legacy Bridge

### High Level Logical highlights

The system is comprised of four logical components;

* + Event Monitor
  + Data Importer
  + Data Transformation
  + Data Exporter

Since the end goal of this framework design is to facilitate the replacement of the current HRMS system, PeopleSoft, with a new cloud version with minimum rework a Black Box System approach has been used. The design of the system focuses on using interface contracts to enforce the standalone nature desired. This allows for each component to be modified/replaced without impacting the rest of the framework.

#### Event Monitor

In the initial phase, each monitor uses SQL queries to look at existing trigger tables looking for events. Since an EID is required to process any record found, after a trigger event has been identified an additional query will be performed against the cross-reference tables associated with the trigger table. If an EID is found, the event can be processed, otherwise the event is skipped until the next monitor cycle. After a configurable number of times an event is skipped an email notice is sent to support and the event is marked as notified. Events that have reached this state will continue to be evaluated each pole cycle but will not generate additional emails.

Each event monitor will update the trigger table associated with the monitor when either the event has been sent to the client or when an error will prevent an event from ever being processed without outside intervention.

#### Data Importer

In the initial phase the data importer will be SQL in nature. The importer takes a notice message from the **Event Monitor** and will retrieve the data from the PeopleSoft DB per the export channel and other information contained in the notice metadata.

#### Data Transformation

The data transformation layer is a sub layer of the Data Importer and converts the incoming data from an HRMS system into the internal data model.

#### Data Exporter

The data export layer consists of a processor that consumes the meta data provided by the event monitor to route the message to the correct exporter. In this phase the exporters are for;

* Employee Events
  + HRZ101A – New Hire/Rehire
  + HRZ102A – Terminate
  + HRZ104A – Job Profile Changes
  + HRZ105A – Demographic Changes
  + HRZ107A – Date Changes
  + HRZ109A – Group Transfers
* Contingent Employee and Multiple EID Events
  + HRZ201A – New Hire/Rehire
  + HRZ202A – Terminate
  + HRZ205A – Demographic Changes

The exporters will be using REXEC connections to the AS/400 and call the correct methods listed above based on the event metadata and event source.

## Legacy Bridge Workflows

This section describes the high-level workflow for the Legacy Bridge.

### Legacy Bridge Workflow



Figure : Legacy Bridge High-Level Workflow

#### Trigger Tables

* PS\_ZHRT\_INTTRIGGER trigger table fields:
  + SEQ\_NBR – Gets incremented by 1, through the proc GetSequenceNumber
  + OPRID – Oprid of the Employee whose data is being sent
  + EMPLID – PeopleSoft Employee ID of the Employee whose data is being sent
  + EFFDT – Effective date of the changes
  + EFFSEQ – Effective sequence for the changes (if any)
  + PROC\_NAME - Procedure Name field helps to understand what changes are made to the Employee and which library needs to be invoked to send the data to the Legacy System.
  + TASK\_FLAG – Values C complete, P pending, E error, and Z
* ZHRT\_ALTTRIGGER trigger table fields:
  + SEQ\_NBR – PSID index
  + OPRID – EID of employee doing hire/rehire
  + EMPLID – PeopleSoft Employee ID
  + EFFDT – Effective date of the changes
  + EFFSEQ – Effective sequence for the changes (if any)
  + PROC\_NAME
  + TASK\_FLAG – Values C complete, P pending, E error, and Z
  + SEQUENCE

#### PeopleSoft Tables

* PS\_ADDRESSES
* PS\_COUNTRY\_TBL
* PS\_DIVERS\_ETHNIC
* PS\_ETHNIC\_GRP\_TBL
* PS\_HRS\_SOURCE\_I
* PS\_JOB
* PS\_LOCATION\_TBL
* PS\_NAMES
* PS\_PERS\_APPL\_REF
* PS\_PERS\_DATA\_EFFDT
* PS\_PERS\_NID
* PS\_PERSON
* PS\_PERSONAL\_DATA
* PS\_PERSONAL\_PHONE
* PS\_ZHRT\_CMPNY\_CREF
* PS\_ZHRT\_ETHCD\_CREF
* PS\_ZHRT\_JOBCD\_CREF
* PS\_ZHRT\_RFSRC\_CREF
* ZPS\_ZGLT\_PT12P\_CREF

#### Meta Data Properties

* Trigger Source
* Event Type
* Process Type
* Process Name
* Error Type
* Error Message
* Hire Rehire Flag
* Completion Status

### Event Monitor Workflow



Figure : High-Level overview of the Event Monitor workflow

#### Scan Trigger Table for New Events

##### See 3.3.1.1 (Trigger Tables)

##### //Get New Trigger Data from PS\_ZHRT\_INTTRIGGER where TASK\_FLAG = 'P' and EFFDT <= Today or PROC\_NAME='ZHRI101A' or PROC\_NAME='ZHRI106A'

##### //Check-If-Contractor

##### //Insure the new hire and rehire appear before the other changes even if they don't show up first in the sequence; PROC\_NAME in ('ZHRI101A', 'ZHRI106A')

#### Check for EID Process

* Sequence ID <> 0
* Get EID – See [5.1.19 Get-LegID-for-SeqNum](#_Get-LegID-for-SeqNum_-_This)
* EID Not Null and Length > 0
* Import EID

#### Add Event to Job List – See 4. 1 Event Object

#### Add Metadata – See 3.3.1.3 Meta Data Properties and 4.1 Event Object

#### Update Process Flag in Trigger Table

##### PS\_ZHRT\_INTTRIGGER.TASK\_FLAG – Values C complete, P pending, E error

##### ZHRT\_ALTTRIGGER.TASK\_FLAG – Values C complete, P pending, E error

#### Email Error Notification

### Data Importer / Data Transformation Process Workflow



Figure : Data Importer Workflow

#### Get New Record from PS Tables

##### AD-Get-Job-Data

##### AD-Get-Job-Description

##### AD-Get-EmplStatus-Description

##### AD-Get-JobStart-Date

##### AD-Get-Pers-Data-Effdt

##### AD-Get-NameSuffix

##### AD-Get-Country-Code

##### AD-Get-Business-Phone

##### AD-Get-Employee-Fax

##### AD-Get-LegSuperviorID

##### AD-Get-Employment-Data

##### AD-Get-Names

#### Create New Internal HR Record – See 4.3 Person Object

#### Transform Data to Internal Format

##### Action Code (H=Hire, C=Change, T=Termination)

##### Legacy Employee Id

##### Employee Id

##### First Name

##### Last Name

##### Preferred Name

##### Name Suffix Text

##### HR Location

##### Department ID

##### Job Title Code

##### Job Title Description

##### Work Status

##### Employee Status Description

##### Hire Date / Rehire Date

##### Termination Date

##### Current Job Start Date

##### Full/Part Time Flag (F OR P)

##### Employee Class

##### Business Phone Number

##### Fax Number

##### Supervisor ID

##### Leg Supervisor ID

##### Country Code

##### Language Preference

#### Update Metadata – See 3.3.1.3 Meta Data Properties and 4.1 Event Object

#### Route Event to Exporter

#### Check-If-Correct102A - Checks to see if 102A process has PS\_JOB row

#### Check-Effdt-Transaction

#### Build-Active-Dir-Output-File

#### Write-Active-Dir-Output-File

### Export Service Workflow



Figure : Export Service Workflow

# 

#### Read Metadata – See 3.3.1.3 Meta Data Properties and 4.1 Event Object

#### Call REXEC AS/400 Employee Procedure See also 5.0 Breakdown of AS400 Processes

* ZHRI101A – Hire Process
* ZHRI102A – Termination
* ZHRI104A – Job Profile Changes
* ZHRI105A – Demographics Changes
* ZHRI106A – Rehire
* ZHRI107A – Dates – Employment Review, Accomplishment and Contract Data changes
* ZHRI109A – Group Transfers
* ZHRI101D – A row deleted in Hire Process
* ZHRI102D – A row deleted in Term Process
* ZHRI104D – A row deleted in Job Profile
* ZHRI105D – A row deleted in Demographics
* ZHRI106D – A row deleted in Rehire
* ZHRI107D – A row deleted in Employment Review, Accomplishment and Contract Data changes
* ZHRI109D – A row deleted in Group Transfers

#### Call REXEC AS/400 TempMast Procedure See also 5.0 Breakdown of AS400 Processes

* ZHRI201A - Contingent Employee and Multiple EID New Hire
* ZHRI202A - Contingent Employee and Multiple EID Terminate
* ZHRI205A - Contingent Employee and Multiple EID Demographic Change
* ZHRI206A - Contingent Employee and Multiple EID Rehire

#### Notify OPS Unsupported Event detected

#### Update Status in Event Monitor - See 3.3.2.4 Update Process Flag in Trigger Table

# Domain Objects

## Event Object (wraps trigger, person, and additional metadata)

### Event Object Properties (Metadata)

* Trigger Source
* Event Type
* Process Type
* Process Name
* Error Type
* Error Message
* Hire Rehire Flag
* As of Today?
* Completion Status

### Event Object Methods

* Hire
* Rehire
* Transfer
* Terminate
* Change Job Profile
* Change Demographic
* Change Date
* Delete

## Trigger Object

### Trigger Object Properties

* Sequence Number - PSID index
* Operator ID – EID of employee doing hire/rehire
* Employee ID – PeopleSoft Employee ID of the Employee whose data is being sent
* Effective Date – Effective date of the changes
* Effective sequence – Effective sequence for the changes (if any)
* Procedure Name - Procedure Name field helps to understand what changes are made to the Employee and which library needs to be invoked to send the data to the Legacy System.
* Task Flag – Completion Status (Values C complete, P pending, E error, and Z)

### Trigger Object Methods

* Read Trigger Record(s)
* Update Status Type

## Person Object

### Person Object Properties (incomplete)

* ZHRF\_LEG\_EMPL\_ID – Legacy Employee ID
* PS\_JOB
  + HR01-Build-Call-Statement
  + LEG SRV YEAR/MON/DAY (EFFDT)
  + LEG UNION FLAG (EMPL\_CLASS)
  + LEG TIMECARD FLAG (FLSA\_STATUS)
  + LEG NID (““)
* PS\_NAMES
  + LEG LAST NAME (LAST\_NAME)
  + LEG FIRST NAME (FIRST\_NAME)
  + LEG MIDDLE INITIAL (MIDDLE\_NAME)
  + LEG NICKNAME (FIRST\_NAME)
  + NAME PREFIX (NAME\_PREFIX)
* PS\_PERS\_DATA\_EFFDT
  + LEG MARITAL STATUS (MAR\_STATUS)
  + LEG GENDER (SEX)
* PS\_PERSON
  + LEG BIRTH YEAR /MON/DAY (BIRTHDATE)
* PS\_ADDRESSES
  + LEG ADDRESS (ADDRESS1)
  + LEG CITY (CITY)
  + LEG STATE (STATE)
  + LEG ZIP (POSTAL)
* PS\_COUNTRY\_TBL
  + PS NID COUNTRY (COUNTRY\_2CHAR)
* PS\_PERSONAL\_PHONE
  + LEG HOME AREA CODE.PHONE/ERROR (PHONE - HOME)
  + LEG WORK AREA CODE.PHONE/ERROR (PHONE - WORK)
* PS\_ZHRT\_JOBCD\_CREF
  + LEG JOB STATUS (ZHRF\_LEGJOBSTSCD)
  + LEG POSITION (ZHRF\_LEGPOSITIONCD)
  + LEG DEPTID (ZHRF\_LEGDEPTCD)
  + LEG SUB DEPTID (‘03’)
* PS\_ZHRT\_ETHCD\_CREF
  + LEG RACE (ZHRF\_LEGETHNICCD
* PS\_ZHRT\_RFSRC\_CREF
  + LEG REF SRC (ZHRF\_LEGRECRUITSRC)
* PS\_HRS\_SOURCE\_I (HRS\_SOURCE\_DESCR)
  + LEG SPECIFIC REF SRC (HRS\_SOURCE\_DESCR)
* PS\_ZHRT\_CMPNY\_CREF
  + LEG GROUP (ZHRF\_LEGGROUP)
* ZPS\_ZGLT\_PT12P\_CREF (ZGLF\_PT2OBR)
  + LEG BRANCH (ZGLF\_PT2OBR)
* PS\_JOB
  + TERM MONTH
  + TERM DAY
  + TERM YEAR
  + REHIRE MONTH
  + REHIRE DAY
  + REHIRE YEAR
  + AUDIT OPRID
  + TERM REASON
* PS\_JOB.ACTION
* PS\_JOB.ACTION\_REASON
  + PS\_ZHRT\_TRMRS\_CREF
    - VOLUNT/INVOL (ZHRF\_LEGTERMCD)
    - TERM CODE (ZHRF\_LEGTERMRSN)
* PS\_JOB
  + WORK STATUS
  + UNION FLAG
  + TIMECARD FLAG
  + EFFDT
  + LEGACY SUBDEPTID (““)
* PS\_JOB.COMPANY
* PS\_JOB. SETID\_JOBCODE
* PS\_JOB. JOBCODE
* PS\_JOB.EMPL\_CLASS
* PS\_JOB.FULL\_PART\_TIME
* PS\_JOB.REG\_TEMP
* PS\_JOB.DEPTID
* ZPS\_ZGLT\_PT12P\_CREF
  + LEGACY BRANCH (ZGLF\_PT2OBR)
* PS\_ZHRT\_JOBCD\_CREF
  + LEGACY DEPTID (ZHRF\_LEGDEPTCD)
  + LEGACY POSITION (ZHRF\_LEGPOSITIONCD)
  + LEGACY JOBSTATUS (ZHRF\_LEGJOBSTSCD)

### Person Object Methods

* Read Person Records

# Controller Objects

## Mediator Pattern

The Mediator Pattern defines an object that encapsulates how a set of objects interact. The Mediator Pattern promotes loose coupling by keeping objects from referring to each other explicitly, and it lets you vary their interaction independently.

## Persistence Layer

## The Persistence Layer will be implemented through Data Access Objects (DAO) and a Java Persistence API (JPA) representing the PeopleSoft database tables that are used by this application (see 3.3.1.2 PeopleSoft Tables).

# Breakdown of AS/400 Processes

### HRI101A – Employee New Hire

#### Input Parameters

* + $Wrk\_Oprid = $AuditOprid
  + $Wrk\_Emplid = $PSEmplid
  + $Wrk\_Effdt = $PSEffdt
  + #PSEffseq to #Wrk\_Effseq
  + $Wrk\_Process\_Name = $WrkProcessOprId
  + $Wrk\_Inf\_ = ‘ ‘
  + $ADAction\_Code = 'H'
  + $ADLegOprid = ‘’
* **Sub-Processes**
  + HR01-Initialize-Fields
  + HR01-Get-Job-Data - retrieve data from the Job table
  + HR01-Get-Personal-Data - retrieve data from the personal data tables
  + HR01-Get-Main-Phone
  + HR01-Get-Office-Phone - retrieve data from the personal phone table
  + HR01-Get-Ethnic-Group - retrieve data from the diversity table
  + HR01-Massage-Data – convert data to form that the RPG program needs it in\*
  + HR01-Build-Call-Statement - builds and execute the RPG call statement with all of the parameters to populate legacy system\*
  + HR01-Get-Group
  + HR01-Get-Branch
  + HR01-Get-Legacy-Ethnic-Code
  + HR01-Get-Position
  + HR01-Get-Referral-Source
  + HR01-Get-Location-Country - Get the country used to get the national id
  + Get-Oprid – get employee’s operator ID, if exists
  + Get-2Char-Country
  + Get-Ethnic-Code
  + Call-System\*
  + Call-Error-Routine\*
  + ZADDOPR.SQC - add the new employee as a PeopleSoft Operator
  + Remove-Non-Letters-Numbers ($WrkOldPhone,$WrkNewPhone) - ZRmvSpcChr.sqc\*
  + Replace-Character($LegLastName,'''','''''',$LegLastName) !Replace all single apostrophe with four apostrophes ZRmvSpcChr.sqc\*
* **Tables Accessed**
  + PS\_ADDRESSES
  + PS\_COUNTRY\_TBL
  + PS\_DIVERS\_ETHNIC
  + PS\_ETHNIC\_GRP\_TBL
  + PS\_HRS\_SOURCE\_I
  + PS\_JOB
  + PS\_LOCATION\_TBL
  + PS\_NAMES
  + PS\_PERS\_APPL\_REF
  + PS\_PERS\_DATA\_EFFDT
  + PS\_PERS\_NID
  + PS\_PERSON
  + PS\_PERSONAL\_DATA
  + PS\_PERSONAL\_PHONE
  + PS\_ZHRT\_CMPNY\_CREF
  + PS\_ZHRT\_ETHCD\_CREF
  + PS\_ZHRT\_JOBCD\_CREF
  + PS\_ZHRT\_RFSRC\_CREF
  + ZPS\_ZGLT\_PT12P\_CREF

### HRI102A – Employee Termination

#### Input Parameters

* + #Wrk\_Sequence to #WrkSeqNbr
  + $PSAuditOperId = $AuditOprid
  + $PSDateIn = $PSEffdt
  + $Wrk\_Emplid = $PSEmplid
  + $ADAction\_Code = 'T'
  + $ADLegOprid = ‘’
* **Sub-Processes**
  + HR02-Initialize-Fields
  + HR02-Get-Action-Reason - determine if a termination was voluntary or involuntary based on Action and Action Reason codes
  + HR02-Get-Reason-Description - gets the description field from the Action Reason table
  + HR02-Process-Data\*
  + HR02-Trim-Parameters\*
  + Get-Oprid – get employee’s operator ID, if exists
  + Call-System\*
  + Call-Error-Routine\*
  + Remove-Non-Letters-Numbers ($PSTermReason,$PSTermReason) - ZRmvSpcChr.sqc\*
  + DTU-Add-Days($PSDateIn, #NumberOfDays, $PSDate) - Add one day to the date using DateMath.sqc\*
* **Tables Accessed**
  + PS\_ACTN\_REASON\_TBL
  + PS\_JOB
  + PS\_ZHRT\_TRMRS\_CREF

### HRI104A – Employee Job Profile Change

#### Input Parameters

* + $PSUserOprid = $AuditOprid
  + $Wrk\_Emplid = $PSEmplid
  + #PSEffseq to #WrkEffseq
  + $ADAction\_Code = 'C'
  + $ADLegOprid = ''
* **Sub-Processes**
  + HR04-Initialize-Fields
  + HR04-Get-Job-Data
  + Get-Oprid
  + HR04-Massage-Data
  + HR04-Call-RPG
  + HR04-Get-Group
  + HR04-Get-Branch
  + HR04-Get-Position
  + Call-System
  + Prepare-Error-Parms
  + Call-Error-Routine
* **Tables Accessed**
  + PS\_JOB - PeopleSoft Job table
  + PS\_ZHRT\_CMPNY\_CREF - Company cross reference file
  + PS\_ZHRT\_JOBCD\_CREF - Job Code Cross Reference Table
  + ZPS\_ZGLT\_PT12P\_CREF - Company cross reference file

### HRI105A – Employee Demographic Change

#### Input Parameters

* + $PSemp = $AuditOprid
  + $Wrk\_Emplid = $PSEmplid
  + $ADAction\_Code = 'C'
  + $ADLegOprid = ''
  + $Wrk\_ADCountryCdBuild = 'Y'
* **Sub-Processes**
* **Tables Accessed**

### HRI106A – Employee Rehire

#### Input Parameters

* + Wrk\_Oprid = $AuditOprid
  + Wrk\_Emplid = $PSEmplid
  + Wrk\_Effdt = $PSEffdt
  + #PSEffseq to #Wrk\_Effseq
  + Wrk\_Process\_Name = $WrkProcess
  + ADAction\_Code = 'R'
* **Sub-Processes**
* **Tables Accessed**

### HRI107A – Employee Date Change

#### Input Parameters

* + Wrk\_Emplid = $PSEmplid
  + ADAction\_Code = ''
  + ADLegOprid = ''
* **Sub-Processes**
* **Tables Accessed**

### HRI109A – Employee Group Transfer

#### Input Parameters

* + PSUserOprid = $AuditOprid
  + Wrk\_Emplid = $PSEmplid
  + #PSEffseq to #WrkEffseq
  + ADAction\_Code = 'C'
  + ADLegOprid = ''
* **Sub-Processes**
* **Tables Accessed**

### HRI101D – A row deleted in Hire Process

#### Input Parameters

* + $ErrorProgramParm = 'HRZ101A'
  + $ErrorMessageParm = 'A row was deleted on the hire process'
  + $WrkCriticalFlag = 'Y'
  + $CompletionStatus = 'C'
* **Sub-Processes**
  + Prepare-Error-Parms
  + Call-Error-Routine
  + Update-Trigger-Row
* **Tables Accessed**
  + PS\_ZHRT\_INTTRIGGER

### HRI102D – A row deleted in Term Process

#### Input Parameters

* + $ErrorProgramParm = 'HRZ102A'
  + $ErrorMessageParm = 'A row was deleted on the termination process'
  + $WrkCriticalFlag = 'Y'
  + $WrkCriticalFlag = 'N'
  + $CompletionStatus = 'C'
* **Sub-Processes**
  + Prepare-Error-Parms
  + Call-Error-Routine
  + Update-Trigger-Row
* **Tables Accessed**
  + PS\_ZHRT\_INTTRIGGER

### HRI104D – A row deleted in Job Profile

#### Input Parameters

* + $ErrorProgramParm = 'HRZ104A'
  + $ErrorMessageParm = 'A row was deleted on the job-profile process'
  + $WrkCriticalFlag = 'Y'
  + $WrkCriticalFlag = 'N'
  + $CompletionStatus = 'C'
* **Sub-Processes**
  + Prepare-Error-Parms
  + Call-Error-Routine
  + Update-Trigger-Row
* **Tables Accessed**
  + PS\_ZHRT\_INTTRIGGER

### HRI105D – A row deleted in Demographics

#### Input Parameters

* + $ErrorProgramParm = 'HRZ105A'
  + $ErrorMessageParm = 'A row was deleted on the demographics process'
  + $WrkCriticalFlag = 'Y'
  + $WrkCriticalFlag = 'N'
  + $CompletionStatus = 'C'
* **Sub-Processes**
  + Prepare-Error-Parms
  + Call-Error-Routine
  + Update-Trigger-Row
* **Tables Accessed**
  + PS\_ZHRT\_INTTRIGGER

### HRI106D – A row deleted in Rehire

#### Input Parameters

* + $ErrorProgramParm = 'HRZ106A'
  + $ErrorMessageParm = 'A row was deleted on the re-hire process'
  + $WrkCriticalFlag = 'Y'
  + $WrkCriticalFlag = 'N'
  + $CompletionStatus = 'C'
* **Sub-Processes**
  + Prepare-Error-Parms
  + Call-Error-Routine
  + Update-Trigger-Row
* **Tables Accessed**
  + PS\_ZHRT\_INTTRIGGER

### HRI107D – A row deleted in Employment Review, Accomplishment, Contract Data changes

#### Input Parameters

* + $ErrorProgramParm = 'HRZ107A'
  + $ErrorMessageParm = 'A row was deleted on the dates process'
  + $WrkCriticalFlag = 'Y'
  + $WrkCriticalFlag = 'N'
  + $CompletionStatus = 'C'
* **Sub-Processes**
  + Prepare-Error-Parms
  + Call-Error-Routine
  + Update-Trigger-Row
* **Tables Accessed**
  + PS\_ZHRT\_INTTRIGGER

### HRI109D – A row deleted in Group Transfers

#### Input Parameters

* + $ErrorProgramParm = 'HRZ109A'
  + $ErrorMessageParm = 'A row was deleted on the group transfer process'
  + $WrkCriticalFlag = 'Y'
  + $WrkCriticalFlag = 'N'
  + $CompletionStatus = 'C'
* **Sub-Processes**
  + Prepare-Error-Parms
  + Call-Error-Routine
  + Update-Trigger-Row
* **Tables Accessed**
  + PS\_ZHRT\_INTTRIGGER

### HRI201A – Contingent Employee and Multiple EID New Hire

#### Input Parameters

* + Wrk\_Oprid = $NAuditOprid
  + Wrk\_Emplid = $NPSEmplid
  + Wrk\_Effdt = $NPSEffdt
  + #NPSEffseq to #Wrk\_Effseq
  + Wrk\_indexNum = to\_char(#indexNum)
  + Wrk\_Process\_Name = $NWrkProcess
* **Sub-Processes**
  + HR201-Get-POI-LegPosNo
  + Call-System
  + HR201-INITIALIZE-FIELDS
  + HR201-Get-POI-data
  + MAIN-SQL-POI
  + HR201-Get-EMP-data
  + MAIN-SQL-EMP
  + HR201-Get-Personal-Data
  + HR201-Massage-Data
  + HR201-Build-Call-Statement
  + Get-Oprid
  + HR201-Get-POI-LegPosNo
  + HR201-GET-EMP-POI
  + HR201-Get-primEid-POIdata
  + HR201-Get-Alternate\_Type
* **Tables Accessed**
  + PS\_PERS\_DATA\_EFFDT
  + PS\_NAMES
  + PS\_ZHRT\_PER\_POI\_TR
  + PS\_ZHRR\_MULTPL\_EID
  + PS\_ZHRR\_POI\_EMP\_VW
  + PS\_ZPTT\_XLAT\_TBL

### HRI202A – Contingent Employee and Multiple EID Terminate

#### Input Parameters

* + PSAuditOperId = $NAuditOprid
  + PSDateIn = $NPSEffdt
  + Wrk\_Emplid = $NPSEmplid
  + Wrk\_indexNum = to\_char(#indexNum)
* **Sub-Processes**
  + HR202-Initialize-Fields
  + Get-Oprid
  + HR202-Call-System
  + Call-System
* **Tables Accessed**
  + PS\_ZHRT\_PER\_POI\_TR
  + PS\_ZHRR\_MULTPL\_EID

### HRI205A – Contingent Employee and Multiple EID Demographic Change

#### Input Parameters

* + PSAuditemp = $NAuditOprid
  + Wrk\_Emplid = $NPSEmplid
  + Wrk\_indexNum = to\_char(#indexNum)
  + PSEffdt = $NPSEffdt
* **Sub-Processes**
  + HR205-Initialize-Fields
  + HR205-Get-POI-data
  + MAIN-SQL-POI - ZHRISTDT.SQC
  + HR205-Get-EMP-data - Get Alternate EID Emp data
  + MAIN-SQL-EMP - ZHRISTDT.SQC
  + HR205-Get-Personal-Data
  + HR205-Massage-Data
  + Get-Oprid
  + HR205-Call-RPG-Program
  + HR205-Get-POI-LegPosNo
  + HR205-GET-EMP-POI
  + HR205-Get-primEid-POIdata
  + HR205-Get-Alternate\_Type
  + Replace-Character($PSLName,'''','''''',$PSLName) !Replace all single apostrophe with four apostrophes ZRmvSpcChr.sqc
  + Call-System
* **Tables Accessed**
  + PS\_PERS\_DATA\_EFFDT
  + PS\_NAMES
  + PS\_ZHRR\_POI\_EMP\_VW
  + PS\_ZHRT\_PER\_POI\_TR
  + PS\_ZHRR\_MULTPL\_EID
  + PS\_ZPTT\_XLAT\_TBL

### HRI206A – Contingent Employee and Multiple EID Rehire

#### Input Parameters

* + Wrk\_Oprid = $NAuditOprid
  + Wrk\_Emplid = $NPSEmplid
  + Wrk\_Effdt = $NPSEffdt
  + #NPSEffseq to #Wrk\_Effseq
  + Wrk\_indexNum = to\_char(#indexNum)
  + Wrk\_Process\_Name = $NWrkProcess
* **Sub-Processes**
* **Tables Accessed**

### Get-LegID-for-SeqNum - This routine gets the Legacy ID from Alternate EID Table

#### Input Parameters

* + Wrk\_Emplid
  + #indexNum
* **Tables Accessed**
  + PS\_ZHRR\_MULTPL\_EID
    - ZHRF\_LEG\_EMPL\_ID
* **Process**

begin-select

MULT.ZHRF\_LEG\_EMPL\_ID

Let $PSOprid = Ltrim(Rtrim(&MULT.ZHRF\_LEG\_EMPL\_ID,' '),' ')

if $PSOprid <> ''

Let $Found = 'Y'

end-if

show 'Mult $PSOprid: ' $PSOprid

from PS\_ZHRR\_MULTPL\_EID MULT

where MULT.Emplid = $Wrk\_Emplid

and MULT.Sequence = #indexNum

end-select

# Externally Configurable Parameters

# Appendix A – Business service classifications

## Definitions, Acronyms and Abbreviations

The following abbreviations and acronyms have been used in this document.

Table : Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| **Term** | **Meaning** |
| **PS** | People Soft |
| **ERD** | Employee Reference Data |
| **SQL** | **Structured Query Language**: language used to access data held in a database |
| **TBD** | **To Be Determined** |
| **REST** | **Representational State Transfer**: Interface standard that allows for interchanging data between systems via web services. |
| **JSON** | **JavaScript Object Notation**: a lightweight data-interchange format easily read by humans and processed by computers |
| **ACL** | **Access Control List**: a list that tells a computer system which access rights a user or client has to a particular data object such as allowed access to PII data elements |
| **RESTful** | Interfaces that implement a REST like service |
| **DB** | **Database**: a system used to store large record sets allowing for standard methods to manage CRUD operations. |
| **CRUD** | **Create Retrieve Update Delete**: a basic set of operations done on data sets |
| **ETL** | Method to Extract, Translate, Load data from one system to another |
| **DAO** | A Data Access Object (DAO) is an object that provides an abstract interface to some type of database or other persistence mechanism. By mapping application calls to the persistence layer, the DAO provides some specific data operations without exposing details of the database. In the context of this application, it is the Java representation of the pertinent People Soft database. |
| **JPA** | Java Persistence API (JPA). The is a Java specification for accessing, persisting, and managing data between Java objects / classes and a relational database. This is used to implement the DAO. |
| **Marshal** | Marshaling is the process of transforming the memory representation of an object to a data format suitable for storage or transmission, and it is typically used when data must be moved between different parts of a computer program or from one program to another. |
| **REXEC** | Remote Process Execution |
| **The Mediator Pattern** | Allows loose coupling by encapsulating the way disparate sets of objects interact and communicate with each other. Allows for the actions of each object set to vary independently of one another. |

Table : Service Definitions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Characteristic** | **Aspect** | **Mission Critical** | **Business Critical** | **Business Operational** | **Administrative Services** |
| **Service Hours** | Operational hours | 7 x 24 | 7 x 24 | 5 x 24 | 5 x 8 |
| Maintenance window | 4 hours / month | 8 hours / month | 48 hours / week | 60 hours / week |
| **Service Availability** | Availability | 99.99% | 99.9% | 99% | 99% |
| Backups | No impact to availability | No impact to availability | May impact availability | May impact availability |
| Availability measurement | Required | Required | Required | Required |
| **Reliability** | Unplanned outage | 1 / year | 4 / year | 8 / year | N/A |
| **Customer Support** | Maximum priority of helpdesk call | 0 | 0 | 1 | 2 |
| **Service performance** | Metrics defined in SLA | Required | Required | Required | Not required |
| Backups | No impact to service performance | May impact service performance | May impact service performance | May impact service performance |
| Restore metrics defined in SLA | Required | Required | Required | Not required |
| **Change Management** |  | Required | Required | Required | Not required |
| **IT Service Continuity** | DR Solution | Required with full operational capability | Required with full operational capability | Not required | Not required |
| Availability in DR mode | 99.99 | 99.5% | N/A | N/A |
| Data loss | None | Some, defined in SLA | N/A | N/A |
| Service recovery time | 2 hours | 72 hours | N/A | N/A |
| **Security** | Compliance | Required | Required | Required | Required |
| **Service Reviews** | Frequency | Monthly | Monthly | Six monthly | Not required |