Integration

Design for HCM-Taleo

**Invensense Confidential**

|  |  |
| --- | --- |
| Version: 0.4  Filename: Invensense HR Integration Taleo-HCM Design  Date: Thursday, May 11, 2017  Author: Sumitha Vishnu |  |

Table of contents

1 Document Control 3

1.1 Revision History 3

2 Overview 4

2.1 Background 4

2.2 Purpose 4

2.3 Scope 4

2.4 Assumptions 5

3 Overview 6

3.1 Integration Approach 6

3.2 Use case summary 6

4 Interface Details 8

4.1 Scheduling 8

4.1.1 Spring Batch Scheduling 8

4.2 Taleo – hcm Interfaces 8

4.2.1 New Employee Creation 8

4.3 Taleo – HCM Interfaces 10

4.3.1 Start Date Change 10

4.3.2 Manager Change 10

4.3.3 Department Change 11

4.3.4 Location Change 12

4.3.5 Address Change 12

4.3.6 Name Change 13

4.3.7 Termination 14

4.4 Database 15

4.4.1 XXINV\_INTEGRATION\_PERSON 15

4.4.2 XXINV\_INTEGRATION\_INTERFACE 15

4.4.3 XXINV\_INTEGRATION\_HCM\_MAP 15

5 Environment Details 16

5.1 Required Software 16

5.2 Environment configuration 16

5.3 Error Handling and Application logging 16

6 Issues, Risks, and Assumptions 17

6.1 Risks 17

6.2 Assumptions 17

7 References 18

# Document Control

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Description** | **Name** |
| 0.1 | 20/10/2016 | First Version of the Integration Design Document | Sumitha Vishnu |
| 0.4 | 01/02/2016 | Updated to the final solution details | Sumitha Vishnu |
|  |  |  |  |
|  |  |  |  |

# Overview

## Background

## Purpose

The purpose of the document is to define the interface specifications for integrating the Taleo and HCM systems. This document describes the details of the approach and the components involved in the integration.

## Scope

The scope of this document includes the following:

**Integration Approach –** Describes the approach used for integration.

**Use case summary –** Lists the use cases considered while implementing the integration solution.

**Interface Details** – Describes specific information about the interface, when it will be executed, data mapping and any specific requirements for the same.

## Assumptions

The following is a table of assumptions when developing the interface specifications.

|  |  |
| --- | --- |
| **Assumption** | **Assumption** |
| 1 | 1. Integration server has access to the Taleo and HCM systems and necessary configurations are done like firewall, additional port opening etc if any. |
| 2 | 1. Frequency of execution of the interface will be preconfigured. |
| 3 | 1. KnownAs/PreferredName is not updated in Taleo after an update in HCM |
| 4. | 1. Projected start date change in HCM is not updated in Taleo. |
| 5. | 1. Contingent workers are not included in the HCM-Taleo update. |

# Overview

## Integration Approach

The integration program runs on an Application server hosted in an environment which has access to the Taleo and HCM systems. The integration covers Taleo to HCM dataflow and HCM to Taleo dataflow. The programs run at regular intervals initiating the data fetch from Taleo/HCM based on the use case.

API based approach is followed in HCM to Taleo integration for both Taleo and HCM systems. File based approach is followed for HCM system in the Taleo to HCM integration.

TALEO:

REST based GET and PUT APIs are used to get and update data in Taleo system based on the use cases.

The GET REST API for Employee is invoked to fetch the employee data and when there is an update to be performed on the Taleo employee data, the REST PUT API is invoked. Both operations sends and receives data in the JSON format.

Invocation of the Taleo APIs includes the below steps

1. Call the dispatcher service to fetch the API URLs. The company code is used as input. This returns the URL for the Taleo APIs.
2. Login to use the APIs using the url returned by the dispatcher service. An authorization token is received as output.
3. The authorization token is required to invoke any REST API to perform search, get or update operations.
4. Once all the required operations are performed, logout of the system. Once logged out, the same authorization token cannot be used.

HCM:

The Atom Feeds are used to retrieve the updates performed at HCM for the various entities involved like employee data update, employee assignment update, employee termination etc. The atom feeds are queried with date in the input and all the updates done in the system after the specified date is returned. It gives the changed attribute along with the old and new values.

File based loader is used to create the Pending worker in HCM for which a set of files are generated with the data in Taleo and passed onto the integration loader service of HCM.

The data mapping and transformation if any is performed at the integration server. Few lookups happen at the integration server using the database and hence new addition of entities like Business Unit, Legal Entity, Legislation code etc requires update at the integration database.

## Use case summary

Below is the summary of all the use cases for the Taleo-HCM integration.

1. **New Employee in Taleo**

When a candidate has accepted the offer, Candidate will be converted to Employee and is created as a pending worker in HCM

1. **Start Date change in HCM**

When the projected start date changes for a pending worker, HR will update the start date in HCM and data needs to be updated back to Taleo. This is not in the scope of the version 1.0 of the integration program.

1. **Manager change in HCM**

When a Manager changes in HCM, latest changes to be reflected in Taleo. This is not in the scope of the version 1.0 of the integration program.

1. **Department change in HCM**

When a Department changes in HCM, latest changes to be reflected in Taleo. This is not in the scope of the version 1.0 of the integration program.

1. **Location change in HCM**

When employees get transferred to different location, changes related to location name and code to be reflected in Taleo. This is not in the scope of the version 1.0 of the integration program.

1. **Address change in HCM**

When an employee's personal address changes, the changes needs to be updated back to Taleo. This is not in the scope of the version 1.0 of the integration program.

1. **Name change in HCM**

Personal details like first name, middle name and last name changes, these needs to be updated back in Taleo.

1. **Terminations**

When an employee is terminated in HCM, the corresponding employee record should be terminated in Taleo

# Interface Details

## Scheduling

### Spring Batch Scheduling

Spring provides a batch processing framework to configure the application to run at regular intervals. This framework will be used to run the application deployed on the application server to execute the integration program at preconfigured intervals. The intervals at which the program needs to be run can be specified in the property files while creating the deployable component. Any change to the interval will require a redeployment/restart of the application server.

## Taleo – hcm Interfaces

### New Employee Creation

When a candidate accepts the offer in Taleo the status of the requisition is changed to Filled and the candidate is converted as employee in Taleo. This employee has to be created as a pending worker in HCM. The creation of Pending worker in HCM is done using the File based loader approach.

Taleo system is polled at regular intervals and the time of querying the system is maintained in a database table. Every time the program executes, the time in the database table for the creation interface is used as an input to Taleo.

#### Data mapping

This section includes the data fields required for employee creation in HCM and its corresponding field in Taleo.

|  |  |  |
| --- | --- | --- |
| **Taleo** | **HCM** | **Description** |
| EmployeeId |  | EmployeeId is used in creating different ids in the Person and WorkRelationship files |
| FirstName | FIRST\_NAME | First name of the employee |
| LastName | LAST\_NAME | Last name of the employee |
| Email | EMAIL\_ADDRESS | Email address of the employee. If empty in Taleo, it is mapped as firstname.lastname@hcm.com |
| City | TOWN\_OR\_CITY | City from the address fields |
| County | REGION\_1 | County from the address fields |
| Country | COUNTRY | Country from the address fields |
| Address | ADDRESS\_LINE\_1 | AddressLine1 |
| ZipCode | POSTAL\_CODE | Postal code from the address fields |
| State | REGION\_2 | State from the address fields |
| LegalEntity | LEGAL\_ENTITY\_ID | HCM Legal entity is identified from the integration database and is sent to HCM |
| LegislationCode | LEGISLATION\_CODE | HCM Legislation code is identified from the integration database and is sent to HCM |
| Business\_Unit | BUSINESS\_UNIT\_ID | HCM Business unit id is identified from the integration database and is sent to HCM |
| StartDate | PROJECTED\_START\_DATE | Projected Start date |
|  |  |  |

Various Ids required for HCM FBL loader like Person id, Person number, Assignment id, work terms id etc are created using the predefined format using the Taleo employee id. This format is defined in the property file and injected to Spring during deployment.

#### Taleo API Details

The last queried time stored in the database is used in the input as creationDate\_from. All the employees created after that time is returned in the response. The employee ids returned here are used to get all the details of the employee.

|  |  |  |
| --- | --- | --- |
| **API URL** | **Method** | **Data Field** |
| https://<host>:<port>/chNNN/ats/api/v1/object/object/employee/search?fields=status|firstName|employeeId&creationDate\_from=<*Timestampfromdatabase*> | GET | FirstName  EmployeeId  Status |
| https://<host>:<port>/chNNN/ats/api/v1/object/object/employee/*<Employeeid>* | GET | Employee |

#### HCM API details

File based loader (FBL) approach is used to create the pending worker in HCM. In order to use FBL to load a pending worker, two sets of files are required.

1. PERSON

The files in this group mainly provides information about the personal details of the employee like firstname, last name, address, email etc. The files in this section are:

* F\_PERSON\_ADDRESS\_VO.dat
* F\_PERSON\_EMAIL\_VO.dat
* F\_PERSON\_LEGISLATIVE\_DATA\_VO.dat
* F\_PERSON\_NAME\_VO.dat
* F\_PERSON\_NATIONAL\_IDENTIFIER\_VO.dat
* F\_PERSON\_PHONE\_VO.dat
* F\_PERSON\_TYPE\_USAGE\_VO.dat
* F\_PERSON\_VO.dat

1. WORKRELATIONSHIP

The files in this group provides information about the assignments of the employee, the start date, the legal entity, legislation code, business unit etc.

* F\_ASSIGNMENT\_SUPERVISOR\_VO.dat
* F\_ASSIGNMENT\_VO.dat
* F\_ASSIGNMENT\_WORK\_MEASURE\_VO.dat
* F\_WORK\_RELATIONSHIP\_VO.dat
* F\_WORK\_TERMS\_VO.dat

These folders are zipped into a folder and loaded to the HCM using the web content server. The web content server creates a content Id and using this content id the loader integration service is invoked at HCM. This service imports and loads the data in the HCM.

After the successful import and load of data, the jobs “**Synchronize Person Records**” and “**Update Person Search Keywords**” need to be run. These processes are prescheduled to run at regular intervals at HCM.

In case of contingent worker conversion, where the employee already exists in HCM the person files are not created again and only workrelationship files are loaded. The preloaded Person id from the integration database is used for this purpose.

## Taleo – HCM Interfaces

### Start Date Change

This use case is out of scope of the integration program

When the start date of an employee is updated in the HCM system, the same has to be updated in the Taleo system.

An employee update application that gets executed as per the schedule gets the update from the HCM system as atom feed. The person id in the atom feed is used to look up Taleo employee id in the integration layer database. Using this Taleo employee id, PUT operation is performed on the employee object in Taleo with the JSON formatted message as input.

This use case is out of

#### Data mapping

This section includes the data fields required for integrating the change of start date in HCM to Taleo

|  |  |  |
| --- | --- | --- |
| HCM | Taleo | Description |
| EffectiveStartDate | startDate | Effective start date in HCM is updated as startdate in Taleo. |

#### HCM API details

The Atom feeds are used to fetch any changes to the employee data.

|  |  |  |
| --- | --- | --- |
| **API URL** | **Method** | **Data Field** |
| https://<host>:<port>/hcmCoreApi/atomservlet/employee/empupdate?updated-min=*<Timestampfromdatabase>* | GET | Atom Feed |

#### Taleo API Details

|  |  |  |
| --- | --- | --- |
| **API URL** | **Method** | **Data Field** |
| https://<host>:<port>/chNNN/ats/api/v1/object/employee/*<EmployeeId>* | PUT | Employee |

### Manager Change

This use case is out of scope of the integration program

When the manager of the employee is updated in HCM, the same is populated in Taleo.

An employee assignment update application that gets executed as per the schedule gets the update from the HCM system as atom feed for employee assignment. The person id and the manager id in the atom feed is used to look up Taleo employee id of the person and the changed manager in the integration layer database. Using this Taleo employee id, PUT operation is performed on the employee object in Taleo with the JSON formatted message as input.

#### Data mapping

This section includes the data fields required for integrating the manager change in HCM to Taleo

|  |  |  |
| --- | --- | --- |
| HCM | Taleo | Description |
| ManagerId | manager | ManagerId is the person Id and manager is the person number |

#### HCM API details

The Atom feeds are used to fetch any changes to the employee data.

|  |  |  |
| --- | --- | --- |
| **API URL** | **Method** | **Data Field** |
| https://<host>:<port>/hcmCoreApi/atomservlet/employee/empassignment?updated-min=*<Timestampfromdatabase>* | GET | Atom Feed |

#### Taleo API Details

|  |  |  |
| --- | --- | --- |
| **API URL** | **Method** | **Data Field** |
| https://<host>:<port>/chNNN/ats/api/v1/object/employee/*<EmployeeId>* | PUT | Employee |

### Department Change

This use case is out of scope of the integration program

When the department of an employee changes in the HCM system, the same needs to be updated in the Taleo system.

An employee assignment update application that gets executed as per the schedule gets the update from the HCM system as atom feed. The person id in the atom feed is used to look up Taleo employee id in the integration layer database. Using this Taleo employee id, PUT operation is performed on the employee object in Taleo with the JSON formatted message as input.

#### Data mapping

This section includes the data fields required for integrating the department change in HCM to Taleo

|  |  |  |
| --- | --- | --- |
| HCM | Taleo | Description |
| OrganizationId | department | Department details |

#### HCM API details

The Atom feeds are used to fetch any changes to the employee data.

|  |  |  |
| --- | --- | --- |
| **API URL** | **Method** | **Data Field** |
| https://<host>:<port>/hcmCoreApi/atomservlet/employee/empassignment?updated-min=*<Timestampfromdatabase>* | GET | AtomFeed |

#### Taleo API Details

|  |  |  |
| --- | --- | --- |
| **API URL** | **Method** | **Data Field** |
| https://<host>:<port>/chNNN/ats/api/v1/object/employee/*<EmployeeId>* | PUT | Employee |

### Location Change

This use case is out of scope of the integration program

When the location of an employee changes in HCM, the same needs to be updated in the Taleo system

An employee update application that gets executed as per the schedule gets the update from the HCM system as atom feed. The person id in the atom feed is used to look up Taleo employee id in the integration layer database. Using this Taleo employee id, PUT operation is performed on the employee object in Taleo with the JSON formatted message as input.

#### Data mapping

This section includes the data fields required for integrating the location change in HCM to Taleo.

|  |  |  |
| --- | --- | --- |
| HCM | Taleo | Description |
| LocationId | location | Location details |

#### HCM API details

The Atom feeds are used to fetch any changes to the employee data.

|  |  |  |
| --- | --- | --- |
| **API URL** | **Method** | **Data Field** |
| https://<host>:<port>/hcmCoreApi/atomservlet/employee/empupdate?updated-min=*<Timestampfromdatabase>* | GET | AtomFeed |

#### Taleo API Details

|  |  |  |
| --- | --- | --- |
| **API URL** | **Method** | **Data Field** |
| https://<host>:<port>/chNNN/ats/api/v1/object/employee/*<EmployeeId>* | PUT | Employee |

### Address Change

This use case is out of scope of the integration program

When the address details of an employee changes in HCM, the same needs to be updated in the Taleo system

An employee update application that gets executed as per the schedule gets the update from the HCM system as atom feed. The person id in the atom feed is used to look up Taleo employee id in the integration layer database. Using this Taleo employee id, PUT operation is performed on the employee object in Taleo with the json formatted message as input.

#### Data mapping

This section includes the data fields required for integrating the address field changes in HCM to Taleo.

|  |  |  |
| --- | --- | --- |
| HCM | Taleo | Description |
| Address1 | Address | Address fields |
| TownOrCity | City | City Name |
| State | State | State name |
| PostalCode | zipCode | Postal code |
| Region1 | County | County |
| Country | Country | Country |

#### HCM API details

The Atom feeds are used to fetch any changes to the employee data.

|  |  |  |
| --- | --- | --- |
| **API URL** | **Method** | **Data Field** |
| https://<host>:<port>/hcmCoreApi/atomservlet/employee/empupdate?updated-min=*<Timestampfromdatabase>* | GET | Atom Feed |

#### Taleo API Details

|  |  |  |
| --- | --- | --- |
| **API URL** | **Method** | **Data Field** |
| https://<host>:<port>/chNNN/ats/api/v1/object/employee/*<EmployeeId>* | PUT | Employee |

### Name Change

When the name details of an employee changes in HCM, the same needs to be updated in the Taleo system

An employee update application that gets executed as per the schedule gets the update from the HCM system as atom feed. The person id in the atom feed is used to look up Taleo employee id in the integration layer database. Using this Taleo employee id, PUT operation is performed on the employee object in Taleo with the JSON formatted message as input.

#### Data mapping

This section includes the data fields required for integrating the name changes in HCM to Taleo.

|  |  |  |
| --- | --- | --- |
| HCM | Taleo | Description |
| First Name | firstName | First Name |
| MiddleNames | middleInitial | Middle name |
| LastName | lastName | Last name |

#### HCM API details

The Atom feeds are used to fetch any changes to the employee data.

|  |  |  |
| --- | --- | --- |
| **API URL** | **Method** | **Data Field** |
| https://<host>:<port>/hcmCoreApi/atomservlet/employee/empupdate?updated-min=*<Timestampfromdatabase>* | GET | AtomFeed |

#### Taleo API Details

|  |  |  |
| --- | --- | --- |
| **API URL** | **Method** | **Data Field** |
| https://<host>:<port>/chNNN/ats/api/v1/object/employee/*<EmployeeId>* | PUT | Employee |

### Termination

When an employee is terminated in the HCM system, the same needs to be updated in Taleo and the status of the employee changes to Terminated.

An employee terminate application that gets executed as per the schedule gets the update from the HCM system as atom feed. The person id in the atom feed is used to look up Taleo employee id in the integration layer database. Using this Taleo employee id, PUT operation is performed on the employee object in Taleo with the JSON formatted message as input.

#### Data mapping

This section includes the data fields required for termination of an employee in Taleo when the same is done in HCM

|  |  |  |
| --- | --- | --- |
| HCM | Taleo | Description |
| ActualTerminationDate | Status | When there is a termination date in HCM, the status is changed as Terminated |

#### HCM API details

The Atom feeds are used to fetch any changes to the employee data.

|  |  |  |
| --- | --- | --- |
| **API URL** | **Method** | **Data Field** |
| https://<host>:<port>/hcmCoreApi/atomservlet/employee/termination?updated-min=*<Timestampfromdatabase>* | GET | AtomFeed |

#### Taleo API Details

|  |  |  |
| --- | --- | --- |
| **API URL** | **Method** | **Data Field** |
| https://<host>:<port>/chNNN/ats/api/v1/object/employee/*<EmployeeId>* | PUT | Employee |

## Database

There are two tables used as part of the integration layer

### XXINV\_INTEGRATION\_PERSON

This table stores the mapping of the person Id and person number from the HCM system to the Taleo employee id.

|  |  |
| --- | --- |
| **Column Name** | **Description** |
| PERSON\_ID | HCM Person ID |
| PERSON\_NUMBER | HCM Person Number |
| TBE\_EMP\_ID | Taleo Employee\_Id |

### XXINV\_INTEGRATION\_INTERFACE

This table stores the interface code and the last executed timestamp.

|  |  |
| --- | --- |
| **Column Name** | **Description** |
| INTERFACE\_ID | Interface Code (ex.empUpdate,empTerminate …) |
| UPDATE\_DATE | Last Updated Date |

### XXINV\_INTEGRATION\_HCM\_MAP

|  |  |
| --- | --- |
| **Column Name** | **Description** |
| HCMGUID | Constant GUID for HCM |
| HCMKEY | Key value received from Taleo |
| SRC\_DATA | Description of the key from Taleo |
| DESCRIPTION | General description |

# Environment Details

## Required Software

Following are the software required for the integration server.

|  |  |  |
| --- | --- | --- |
| **Software** | **Deployment** | **Use** |
| Apache Tomcat 8.0 |  | Local |
| JDK 1.8 |  | Local |
| Ant |  | Local |
| Oracle |  |  |

## Environment configuration

|  |  |  |
| --- | --- | --- |
| **System** | **Stage** | **Server** |
| TALEO | Test – Taleo Business edition 16B2 | https://chp.tbe.taleo.net/chp04/ats/login.jsp |
| HCM | Test – HCM 11 | <https://hbbc-test.hcm.us2.oraclecloud.com/> |
| Web Content server | Test | <https://hbbc-TEST.fs.us2.oraclecloud.com:443/cs/idcplg> |
| HCM Loader service | Test | https://hbbc-test.hcm.us2.oraclecloud.com:443//hcmCommonBatchLoader/LoaderIntegrationService?wsdl |

## Error Handling and Application logging

* Error handling in integration program will store the appropriate error messages.
* Logging will be done in custom log files for the application using log4j.
* The last run timestamp will not be updated in the database in case of any error for specific use cases and the same time stamp would be picked up to run again.

# Issues, Risks, and Assumptions

## Risks

|  |  |  |
| --- | --- | --- |
| **No.** | **Description** | **Ref** |
| 1. 1 | . |  |
|  |  |  |
|  |  |  |

## Assumptions

|  |  |
| --- | --- |
| **No.** | **Description** |
| 1 | Details of all the employees like employee id, person number and person Id are available in the integration database |
| 2 | Standard entities like Business Unit, State, Legal entity and Legislation code has its mapping in the integration database. |
| 3 | Jobs like Synchronize Person Records and Update Person Search Keywords are scheduled to run at regular intervals in HCM |
|  |  |

# References

|  |  |
| --- | --- |
| **No.** | **Description** |
| 1. 1 | Business Requirements Mapping Template.xlsx |
|  |  |
|  |  |