

Integration Strategy

April 2nd, 2021

Document Management

This is a snapshot of an on-line document. Paper copies are valid only on the day they are printed. Refer to the author if you are in any doubt about the currency of this document.

Project ID: Program Equip  
Title: Integration Strategy  
Status: Draft  
   
Release: 1.0  
Document Owner: IBM  
Team: Integration   
Author: Aravind Surabhi

Data Created: April 2nd, 2021  
Created By: Aravind Surabhi

Approval

This document requires the following approval signatures:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Title** | **Signature** | **Date** |
| Dan Trares | Project Executive |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Distribution

This document requires distribution to the following individuals:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Title** | **Signature** | **Date** |
|  |  |  |  |
|  |  |  |  |

Revision History

The following document versions have been created:

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Description** | **Revised By** |
| 1.0 | *20210402* | Initial Draft | A. Surabhi |
| 2.0 | *20210510* | Fixed typos and grammar | S. Alberts |

Reference Documents

The following documents are references in this document:

|  |  |  |
| --- | --- | --- |
| **Document Name** | **Document Description** | **Link/Section** |
| Integration Inventory | List of all the integrations that are in scope with SME assignments | Appendix 10.1 |
| Specification (FS/TS) Template | Template formatted document to be used for creating any new FS/TS | Appendix 10.2 |
| Tenneco Interface Tracker Template | List of Interfaces with Grading Key details, Vendor Involvement and Testability Concerns | Appendix 10.3 |
| Current vs. New SF Landscape Strategy 20210413.pptx | Current SF Module Impact Assessment | Section 3 |

Contents

[1. Introduction 6](#_Toc71562945)

[2. Architecture 7](#_Toc71562946)

[2.1 Integration Landscape 7](#_Toc71562947)

[2.2 Integration Types 7](#_Toc71562948)

[2.3 Development Types 8](#_Toc71562949)

[2.4 Middleware 8](#_Toc71562950)

[2.4.1 API Enablement 8](#_Toc71562951)

[2.5 SFTP Folders 9](#_Toc71562952)

[2.6 Standard Integrations 9](#_Toc71562953)

[2.6.1 SAP Cloud Platform Integration Setup 10](#_Toc71562954)

[2.6.2 Minimum Requirements for SAP ERP HCM for Employee and Organizational Data Replications 10](#_Toc71562955)

[2.6.3 Minimum Requirements for SAP ERP for Cost Center Replications 12](#_Toc71562956)

[2.7 Integration Center 14](#_Toc71562957)

[2.8 Intelligent Services 14](#_Toc71562958)

[3. Current SF Module Impact Assessment 14](#_Toc71562959)

[4. ServiceNow Implementation Dependencies and Integration 15](#_Toc71562960)

[5. Development Process 16](#_Toc71562961)

[5.1 Pre-Development Work 16](#_Toc71562962)

[5.1.1 Vendor Management 17](#_Toc71562963)

[5.1.2 Integration Data Model 17](#_Toc71562964)

[5.1.3 Existing Functional Specifications 18](#_Toc71562965)

[5.2 Start of Development 18](#_Toc71562966)

[5.3 Design 19](#_Toc71562967)

[5.3.1 Change Control 19](#_Toc71562968)

[5.3.2 Functional Speciation (FS) 19](#_Toc71562969)

[5.4 Build / Unit Test 19](#_Toc71562970)

[5.4.1 Technical Specifications (TS) 20](#_Toc71562971)

[5.4.2 Build 20](#_Toc71562972)

[5.4.3 Unit Testing 20](#_Toc71562973)

[5.5 String Testing 21](#_Toc71562974)

[5.6 Formal Testing 21](#_Toc71562975)

[5.6.1 Tester 21](#_Toc71562976)

[5.6.2 Vendor Validation 21](#_Toc71562977)

[6. Critical Design Aspects 22](#_Toc71562978)

[6.1 Design Decisions 22](#_Toc71562979)

[6.1.1 Value Translation 22](#_Toc71562980)

[6.1.2 Notifications 22](#_Toc71562981)

[6.2 Effective Dates 22](#_Toc71562982)

[6.3 Testability 23](#_Toc71562983)

[6.4 Future Dated Rows 23](#_Toc71562984)

[6.5 Sequence of Events 23](#_Toc71562985)

[6.6 Calculated SuccessFactors Fields 24](#_Toc71562986)

[6.7 Server Times 24](#_Toc71562987)

[6.8 First Run 24](#_Toc71562988)

[7. Integration Inventory 24](#_Toc71562989)

[7.1 Inventory 24](#_Toc71562990)

[7.2 Interface Counts 25](#_Toc71562991)

[8. Roles and Responsibilities 25](#_Toc71562992)

[8.1 Development Roles and Responsibility by Phase 25](#_Toc71562993)

[8.2 Segregation of Duties (SOD) 26](#_Toc71562994)

[9. Existing SuccessFactors/SAP HRIS/ERP Integrations - TBD 26](#_Toc71562995)

[10. Appendix 27](#_Toc71562996)

[10.1 Interface Inventory 27](#_Toc71562997)

[10.2 Specification (FS/TS) Template 27](#_Toc71562998)

[10.3 Tenneco Interface Tracker Template 27](#_Toc71562999)

# Introduction

This document outlines the Integration Strategy for Program Equip [Phase 1]. The overall goal of the program is to migrate from current SAP HRIS applications [H1P (MP/RP), H1P (CA/PT), HPC, and VPC] to the new cloud based SuccessFactors application. The business functions within scope are Core HR for Employee Central.

Some integrations currently exist with SAP HRIS application that will need to be replicated. Some integrations will remain in SAP HRIS applications, for these, focus is on breakage and remediation. A few integrations currently existing with SAP HRIS application will no longer be needed due to how SuccessFactors operates. And some integrations are expected to be new of which most of them will be Standard ones.

See section 2.2 for Integration Types and 2.3 for Development Types

The following list contains the countries in scope for Program Equip [Phase 1]:

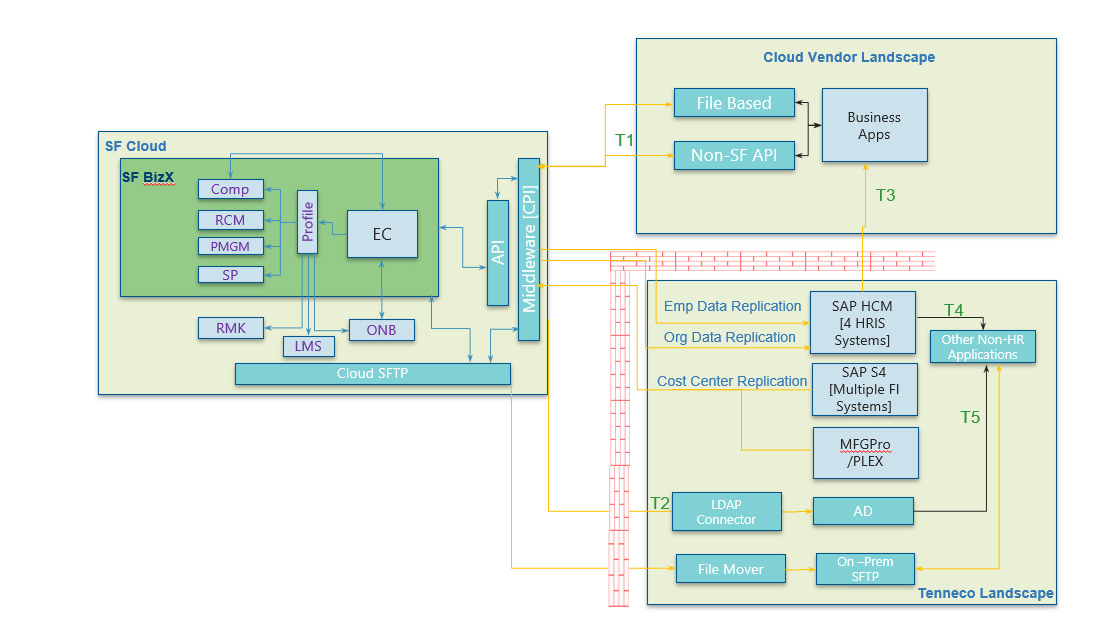


The following list contains the languages in scope for Program Equip [Phase 1]:



# Architecture

## Integration Landscape



## Integration Types

Each integration will fall into one of five categories classified for management of impacts and design/testing coordination.

Type 1 – SAP Cloud and Cloud Vendors

Type 2 – SAP Cloud and SAP On-Premise

Type 3 – Tenneco SAP HCM to Cloud Vendors (note: other than SuccessFactors)

Type 4 – Tenneco SAP HCM to Tenneco Local Applications

Type 5 –Any integration not directly feed by types 1-4 but where HR data flows, one node downstream

Here all the Integration Types have been outlined. For the Inventory Integrations identified in Section 7, they are categorized as T1 or T2 only.

## Development Types

While Integration type classification is driven for better organization of identification of business and application owners, development types are to identify what organization owns the primary implementation of the integration.

D1 - Existing Integration Moving to SF

D2 - Existing Integration Staying in Place (change impact, remediation)

D3 - New SF Integration

D4 - New SAP ECC (non-SF) Integration

D5 - New Other Application Integration (ex: SAP FI, IDAM)

Table below shows primary ownership for implementing integration and development types:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | T1 | T2 | T3 | T4 | T5 |
| D1 | IBM | IBM |  |  |  |
| D2 | Tenneco | Tenneco | Tenneco | Tenneco | Tenneco |
| D3 | IBM | IBM |  |  |  |
| D4 |  | IBM | Tenneco | Tenneco |  |
| D5 |  | Tenneco |  |  | Tenneco |

Examples of classifications:

EC to SAP HRIS integration will have two tracking items.

* EC out – T2D3 (SAP cloud to on-prem and new SF integration)
* SAP HRIS in – T2D4 (SAP cloud to on-prem and new SAP HRIS integration)

Cost Center Integration will have two tracking items:

* EC in – T2D3 (SAP cloud to on-prem and new SF integration)
* SAP FI out – T2D5 (SAP cloud to on-prem and new SAP FI integration)

## Middleware

CPI (Cloud Platform Integration) is the chosen middleware by SAP as the application vendor. SAP provides a license for CPI as part of SuccessFactors License. SAPs best practice is to utilize SAP Cloud Platform Integration tool for all middleware needs.

Primary driver to use the CPI middleware where possible was to use the predefined Standard Integrations scenarios delivered by SAP to connect with On-Prem ERP and it has the best integration with SuccessFactors and different solutions.

### API Enablement

SuccessFactors exposes their API for external middleware connectivity. Note the 4 in api4 represents the data center 4. Number is different for different data centers.

<https://api4.successfactors.com/sfapi/v1/soap>

|  |  |
| --- | --- |
| Endpoint | Information |
| |  |  | | --- | --- | | Service Name: | {urn:server.sfapi.successfactors.com}SFAPIService | | Port Name: | {urn:server.sfapi.successfactors.com}SFAPI | | |  |  | | --- | --- | | Address: | https://api4.successfactors.com:443/sfapi/v1/soap | | WSDL: | <https://api4.successfactors.com:443/sfapi/v1/soap?wsdl> | | Implementation class: | com.successfactors.wsapi\_v1.SFAPI | |
| |  |  | | --- | --- | | Service Name: | {urn:server.sfapi.successfactors.com}SFAPIService12 | | Port Name: | {urn:server.sfapi.successfactors.com}SFAPI12 | | |  |  | | --- | --- | | Address: | https://api4.successfactors.com:443/sfapi/v1/soap12 | | WSDL: | <https://api4.successfactors.com:443/sfapi/v1/soap12?wsdl> | | Implementation class: | com.successfactors.wsapi\_v1.SFAPI | |
| |  |  | | --- | --- | | Service Name: | {urn:server.sfapi.successfactors.com}SFAPIAsyncService | | Port Name: | {urn:server.sfapi.successfactors.com}SFAPIAsync | | |  |  | | --- | --- | | Address: | https://api4.successfactors.com:443/sfapi/v1/async/soap | | WSDL: | <https://api4.successfactors.com:443/sfapi/v1/async/soap?wsdl> | | Implementation class: | com.successfactors.wsapi\_v1.SFAPIAsync | |

## SFTP Folders

SAP provides 2 (two) sets of SFTP Accounts for use with SuccessFactors. One set is dedicated for the production SF environment. All non-production SF environments share the second Account.

SAP issues only a single user ID for each Account. There are no further access controls. As such there should not be any design that allows vendor applications into the SFTP folders. All pulling or drop off of files should be only by the File Mover or middleware that has access.

SAP does not provide long term storage of files in their SFTP folders. SAP has purge logic based on both 14 days and total file space, whatever comes first. All designs should support holding files in the SFTP folders for as minimal time as possible.

For production folders, there will be set up a subfolder for each interface.

For non-production, there will be set up a subfolder for each BizX environment, then a further subfolder for each interface.

## Standard Integrations

Standard Integrations are only provided by SAP and available for SAP Cloud Platform Integration [CPI] or Boomi.

SAP provides many ready to use integrations to many common global HR vendors. SAP worked with these vendors to develop standard integrations, based on common templates based on the vendor’s experience, that are ready to use.

Standard integrations are configurable. However, these integrations in most cases need to be modified as per business requirements. The vendors will be resistant to changing the way the existing integrations work today. Any changes, even to SAP standard integrations, will have a cost to implement in terms of time and resources. In most cases, standard integration serves as a base template to copy and build custom integration.

### SAP Cloud Platform Integration Setup

IBM Integration team will be responsible for the SAP CPI Setup.

For more information about how to set up SAP Cloud Integration see [SAP Cloud Integration](https://help.sap.com/viewer/368c481cd6954bdfa5d0435479fd4eaf/Cloud/en-US).

Based on the Tenneco Architecture and Domains, team will request the Tenneco’s Basis team to install SAP Cloud Connector or use Reverse Proxy if any.

Also see:

https://help.sap.com/viewer/368c481cd6954bdfa5d0435479fd4eaf/Cloud/en-US/14567e1c8618433c9f003e70f0681141.html

### Minimum Requirements for SAP ERP HCM for Employee and Organizational Data Replications

Standard integrations between SuccessFactors Employee Central and SAP HCM will need

On-Premise SAP HCM systems to have support packages and software components in place.

Tenneco’s Basis team to make sure following software components and support packages are in Place for all 4 SAP HRIS applications [H1P (MP/RP), H1P (CA/PT), HPC, and VPC].

| Required Software Component Versions and Support Packages | |
| --- | --- |
| **Component** | **Required Software Component Version** |
| SAP NetWeaver | SAP\_BASIS with one of the following versions:   * 700 with SP27 or a higher SP * 701 with SP12 or a higher SP * 702 with SP08 or a higher SP * 710 with SP15 or a higher SP * 711 with SP10 or a higher SP * 730 with SP03 or a higher SP |
| SAP ERP HCM | The following software component versions are required in your system:   * SAP\_APPL   SAP\_APPL 600 SP15 or a higher version/SP   * SAP\_HR   The required minimum is one of the following versions:   * + SAP\_HR 600 SP59   + SAP\_HR 604 SP25   + SAP\_HR 608 (initial delivery version, no SP required)   The minimum version ensures that the integration works, but you most probably need to implement additional SAP Notes. Exact details of SAP Notes required, depends on the SP level available in your system.  To prevent additional effort caused by investigating the required SAP Notes and by implementing them, we recommend that you use at least one of the following versions:   * + SAP\_HR 600 SPC7 (127)   + SAP\_HR 604 SP93   + SAP\_HR 608 SP21 * EA-HR   The required minimum is one of the following versions:   * + EA-HR 600 SP59   + EA-HR 602 SP38   + EA-HR 603 SP33   + EA-HR 604 SP25   + EA-HR 605 SP02   + EA-HR 606 (initial delivery version, no SP required)   + EA-HR 607 (initial delivery version, no SP required)   + EA-HR 608 (initial delivery version, no SP required)   The minimum version ensures that the integration works, but you most probably need to implement additional SAP Notes. Which SAP Notes are required depends on the exact SP level available in your system.  To prevent additional effort caused by investigating the required SAP Notes and by implementing them, we recommend that you use at least one of the following versions:   * + EA-HR 603 SPA1   + EA-HR 604 SP93   + EA-HR 605 SP64   + EA-HR 606 SP49   + EA-HR 607 SP38   + EA-HR 608 SP21 |
| Integration add-on for SAP ERP HCM and SAP SuccessFactors Employee Central | PA\_SE\_IN 100 SP31  **Note**  Only this newest PA\_SE\_IN support package ensures that all Features described in this document are available in your system.  **Caution**  The PA\_SE\_IN software component can be installed in Unicode systems only. Non-Unicode systems are not supported. |

For more information, see [SFSF EC INTEGRATION 1210Information published on SAP site](http://help.sap.com/disclaimer?site=https://launchpad.support.sap.com/#/softwarecenter/template/products/%20_APP=00200682500000001943&_EVENT=DISPHIER&HEADER=Y&FUNCTIONBAR=N&EVENT=TREE&NE=NAVIGATE&ENR=67837800100900007903&V=INST&TA=ACTUAL&PAGE=SEARCH/SFSF%20EC%20INTEGRATION%201210) in the SAP Software Center. You can access the Software Center from [SAP ONE Support LaunchpadInformation published on SAP site](http://help.sap.com/disclaimer?site=https://launchpad.support.sap.com/) by choosing **Software Downloads**.

### Minimum Requirements for SAP ERP for Cost Center Replications

| Tenneco’s Basis team to make sure following Software components and support packages are in Place for all 21 SAP ERP Systems currently integrated with the 4 SAP HRIS Applications [H1P (MP/RP), H1P (CA/PT), HPC, and VPC].  Required Software Component Versions and Support Packages | |
| --- | --- |
| **For this component...** | **You need this software component version...** |
| SAP NetWeaver | SAP\_BASIS with one of the following versions:   * 700 with SP27 or a higher SP * 701 with SP12 or a higher SP * 702 with SP08 or a higher SP * 710 with SP15 or a higher SP * 711 with SP10 or a higher SP * 730 with SP03 or a higher SP |
| SAP ERP | SAP\_APPL 600 SP15 or a higher version/SP |
| Integration add-on for SAP ERP CO Master Data and SAP SuccessFactors Employee Central | ODTFINCC 600 SP10 or higher |

For more information, see [SFSF EC INTEGRATION 1210Information published on SAP site](http://help.sap.com/disclaimer?site=https://launchpad.support.sap.com/#/softwarecenter/template/products/%20_APP=00200682500000001943&_EVENT=DISPHIER&HEADER=Y&FUNCTIONBAR=N&EVENT=TREE&NE=NAVIGATE&ENR=67837800100900007903&V=INST&TA=ACTUAL&PAGE=SEARCH/SFSF%20EC%20INTEGRATION%201210) in the SAP Software Center. You can access the Software Center from [SAP ONE Support LaunchpadInformation published on SAP site](http://help.sap.com/disclaimer?site=https://launchpad.support.sap.com/) by choosing **Software Downloads**.

| Required SAP Notes | | |
| --- | --- | --- |
| **SAP Note Number** | **Title** | **Description** |
| [1043195 Information published on SAP site](http://help.sap.com/disclaimer?site=https://launchpad.support.sap.com/#/notes/1043195) | **Configuration of Web service runtime** | Gives instructions on how to set up the technical configuration of the Web service runtime environment and how to check it. |
| [1269130 Information published on SAP site](http://help.sap.com/disclaimer?site=https://launchpad.support.sap.com/#/notes/1269130) | **IDoc XML-HTTP SOAP: Problem with SOAP Class** | Gives instructions on how to resolve a SOAP error. |
| [1560878 Information published on SAP site](http://help.sap.com/disclaimer?site=https://launchpad.support.sap.com/#/notes/1560878) | **White list for SOAP Processer & IDoc SOAP** | Gives instructions on how to explicitly allow objects to be processed by SOAP processor. |
| [857321 Information published on SAP site](http://help.sap.com/disclaimer?site=https://launchpad.support.sap.com/#/notes/857321) | **Resending HTTP IDocs Automatically in Status 02** | Gives instructions on how to send IDocs automatically without problems. |
| [1567897 Information published on SAP site](http://help.sap.com/disclaimer?site=https://launchpad.support.sap.com/#/notes/1567897) | **EA/728/SOAP\_MSGID\_PREFIX\_INVALID** | Ensures that IDoc SOAP communication still works if Employee Central sends sequencing information in the control record field ARCKEY. |
| [2255967 Information published on SAP site](http://help.sap.com/disclaimer?site=https://launchpad.support.sap.com/#/notes/2255967) | **Example implementation for company code mapping of cost center** | Provides the CL\_ODTF\_EC\_MAP\_COMP\_CODE\_EXMP class, which you can use as a sample implementation for the **Modify Replication IDOC for Cost Centers** (ODTF\_CO\_REPL\_IDOC\_COST\_CENTERS) Business Add-In (BAdI) to replace the SAP ERP company code with the corresponding Employee Central company code when transferring cost centers. Implement this BAdI if the COMPANY\_CODE\_ID field of the IDoc should contain the Employee Central company code instead of the SAP ERP company code. |

## Integration Center

SuccessFactors option for simple integrations. Designed for business users, not IT specialists.

This tool can create files and transfers them to target locations if mappings and translations are not sophisticated. There are basic abilities for encryption but few for error handling or file archiving. Most pre-project evaluations, by SDLC compliant point of views, consider this tool a fancy reporting tool with some file handling for business owners rather than an integration tool for IT compliance purposes.

While this tool can be utilized for some of the listed integrations, the starting assumption is that they will not be using Integration Center and will be estimated for middleware development. During the project, when details are documented, and evaluation for solution fit with Integration Center will be conducted.

## Intelligent Services

Intelligent Services links with external systems that allow seamless integration with third-party applications besides notifying them as and when these changes originate in HR system on real-time basis. Each **subscriber** is aware of the smallest changes that happen within another application, and automatically responds with intelligence. In summary, An HR transaction no longer begins and ends with a core HR system, but extends across all affected applications to create an end-to-end, complete process.

* Upon Hire event, a record can be created in Active Directory on a real-time basis with username, password enabling new hire to single sign-on across multiple applications. In addition, notification can be sent to manager with welcome email and login credentials.
* Upon Hire event, IT can be notified to make laptop available and third-party benefit vendor can receive request to enroll new employee for eligible benefits
* Termination can trigger downstream applications to perform multiple activities including ending their benefit plans, deactivating their access, running a payroll settlement calculation and so on.

# Current SF Module Impact Assessment

For Program Equip, New SF BizX Environment is the best option to start fresh without disturbing the current SF Environment.

After a couple of meetings, 2 options were derived that will feed the Current SF Environment Employee Profile during the Phase I & II Timeline May-Dec 2022.

Here are the Options outlined:

OPTION 1 [PREFERRED]:

* Send all the Employee Data from SF EC to all 4 SAP HRIS Applications based on Location including the ones without Payroll/Time
* Use the existing 4 Interfaces to feed the Current SF Env. Employee Profile to continue the activities
* During Design Phase we need to make sure all the fields we send from HRIS Systems to Current SF Env, are being accounted for in new SF EC and passed on to 4 HRIS Systems [H1P (MP/RP), H1P (CA/PT), HPC, and VPC].

OPTION 2:

* During the Design phase if we decide to filter and send only the Employees with Payroll/Time to HRIS Applications, then the option is Employee Profile to be updated manually via load

from new to Current SF BizX Env.

* Activities taking place
  + for PMGM Forms in November 2022
    - Manual load is required once or twice in October 2022
  + No Succession
  + Compensation Jan-Mar
  + LMS is ongoing. Connection will be moved to new BizX

**Note:** Need to identify the resource who has expertise to do this activity. This will be an additional staffing effort if IBM needs to bring in the LMS Expert.

* During Design Phase we need to make sure all the fields required by the Employee Profile to support existing SF Environment, are being accounted in new SF EC

Here is the ppt used for Current SF Module Assessment & Plan:



# ServiceNow Implementation Dependencies and Integration

We met with Maurice (Moe) Wilkerson and the following points were discussed w.r.t ServiceNow Integrations with SuccessFactors Employee Central:

* ServiceNow Team is exploring the HR module of ServiceNow. Not sure if that will be the same instance as IT or a separate instance at this point.
* ServiceNow needs the basic employee profile data to be able to identify the requester
* Team will put together requirements for case management that will be used as input to the design.

        Basic Employee Profile Data Includes:  
                \* Departments, Locations, Job Profiles, Worker Profiles and Effective Worker Profiles

During Design Phase, we will look into Use Cases, Employee To-Dos and any additional scenarios from SF EC Perspective.

NOTE: Any Integrations identified will be added as additional work and Timelines will be decided at that time.

# Development Process

While the design and build of SuccessFactors is based on SAP’s iterative cycle methodology, where some requirements are disclosed, the application built, then adjusted over 3 cycles, the interfaces need to follow a closer to traditional SDLC methodology. A primary goal of this strategy is to bridge the agility of the SaaS model with some rigidity of a SDLC model.

Below will outline an escalated methodology that still follows basic SDLC controls. The main goal is to target when the SuccessFactors design is mature enough for the technical interface design. Specifically, Employee Central where the bulk of the data will reside.

Pre-technical specification steps include:

* Vendor Management
* Approved Functional Specifications

Integrations rely heavily on Employee Central design including foundation objects and employee data model (Succession and Country Specific Succession). Any delay in 100% finalization of foundation objects and solid employee data model (almost 90% final) pose a significant risk to timely completion of integration design and subsequent build.

It is very important to have a Statement of Work (SOW) with external integration vendors finalized at the initiation of project to avoid unnecessary delays.

## Pre-Development Work

There are 2 steps that will be critical to starting development on target that can and should start even before the core development teams are on-boarded.

### Vendor Management

For each vendor application, there needs to be a clear identification of the Tenneco resource that have the communication and negotiation responsibility for that vendor. Also, the contact methods need to be documented and understood.

While a larger Communications / Change Management strategy for the project will be a work in progress for some time into the project, a clear leader for vendor management is needed from the start. Often these vendors being outside the project and without a direct stake in the work can be slow to work with.

As stated in section 1.2, the goal is to minimize impacts to the existing integrations. Still there needs to be a clear path to communicate with the vendors to ask clarification questions and get their feedback on the significance of identified impacts.

In addition to the contact information and methods, the following will be needed information about each vendor:

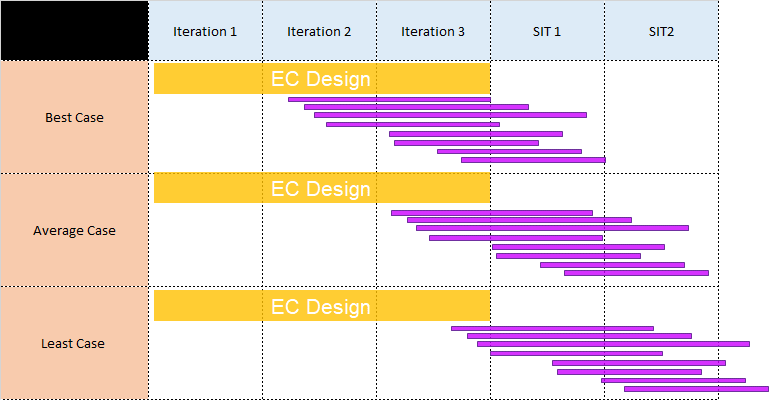
* What is the testing support under existing SLAs? There will be the need for testing, even if the new files are created perfectly like the existing files.
  + What is the lead time for engaging them for testing?
  + How much testing is covered under current costs? Will this be enough or will contract discussions be needed?
  + Do they have a testing environment? If yes, is it currently in a state that is good for testing with the project? If not, how can it get refreshed?
* What are the acceptance criteria for the vendor to allow the new files?

### Integration Data Model

The key to starting interface development on target will be knowing when the new SF data model is mature enough to cover the fields needed for the interfaces. While the data model for SF will be greater than the field needs for the interfaces, for this document purpose the concern is for the interface fields.

To facilitate the start of development the fields needed for the existing interfaces need to be known from the start of the project. With a known list of existing integration fields, the functional design teams can focus early for the inclusion of those fields. The integration team can use the list as a checklist and monitor what interfaces are ready for starting technical build phase.

Employee Data workbook is designed and managed by the EC team which will be used as a reference for the Integration Team. Target for a ‘stable’ Employee Data workbook is mid Iteration 2. At that point most of the business needs will be included. Changing against the known list of fields for integration will start at this point. Based on the results the project will determine the next steps to expand the Employee Data workbook for integration’s remaining needs.



### Existing Functional Specifications

While there is to be some new integration requests, many are existing. The existing integrations will have the same functional expectations and therefore can re-use existing Functional Specifications (FS). It has been identified that not all existing FS exist (can be found), and/or the information is up to date.

Other than resource capacity there is no reason that the collection, updating, and where needed reverse engineering of existing FS should not start even pre-project. With the goal of not delaying the project, all FS should exist and be considered approved by the target date of the mature SF data model.

## Start of Development

Due to the nature of the agile SF methodology, the SF design process is fluid while the integration build would highly desire an early and stable SF data design. The intent will need to be to allow for the agile SF methodology to a point, focus on the data model, and then while not stopping changes, controlling them so impacts are discussed, and the technical teams are in the know.

Start of development will be tagged heavily with the maturity of the data model. The date of maturity needs to be a date set at the start of the project and agreed upon by the functional design teams as they will need to help drive to this goal. The statement is mature data model not a complete data model. There will be a need to achieve a state of maturity and then monitor and control the process for completing and editing the data model.

For development to start on target, the following are critical to have ready in advance:

* Checklist of existing integration data fields
* Mature SF data model
* Change control process for editing the data model, changes are costly once development starts
* Testability – often integrations are designed and built based on the working existing production version. Then often the non-production testing has different architecture where testing fails. Adjusting interfaces just for testability late in the testing phases can result in costly delays. Anomalies for testability need to be identified and accounted for early in the development cycle.

The Integration Team need to be managed so that the pre-work is completed and the developers ready, and then monitor the progress of the data model checklist for targeting the start of individual interfaces. Interfaces can then be assigned when the data model is mature for them.

The key to success at this point in the project is less pre-planning and driving to a date, and more being ready and monitoring for opportunities.

## Design

The integration design phase has two (2) areas, an integration inventory, and functional specifications.

For more details on staffing and resource roles and responsibilities see section 6.

### Change Control

Due to the agile nature of the SF design methodology, changes in the data model and data values will change after development starts. Planning and strategy are needed to evaluate changes and react quickly.

The starting inventory of each field for each existing interface needs to be maintained as a single document throughout the project.

### Functional Speciation (FS)

Also, see 3.1.3 about existing FS.

An approved FS will be needed for each interface to be built. The definition of approved is to be determined by Tenneco. Where available existing and up to date FS can be used. Where not available, or for new integrations, a new FS will need to be created.

Templates for new FS will need to be identified at the start of the project. Review, approval, and change control procedures will need to be defined at the start of the project.

Note: As a document template FS and TS are a single document with creation and approval processes separate by section.

## Build / Unit Test

The build phase has three (3) areas, technical specifications (TS), build, and unit testing.

For more details resource roles and responsibilities see section 8.

Highly recommended that all resources take the SuccessFactors training on the SuccessFactors API’s. Minimally read the published documentations. All integration interaction with SuccessFactors will be via APIs. Selecting the best API to start development is important.

### Technical Specifications (TS)

All integrations in scope as defined by the Integration Inventory will require a TS. Once an individual interface has been selected for development start, based on the data model checklist, the first step is to document the technical aspects needed to solve the requirements as defined by the FS.

Template and approval process for TS to be determined at start of project.

Note: Technical Specifications will not be fully complete before initiating development. However, they will define the integration. Due to the nature of the SF APIs and their usage, plus an often ‘in-flux’ data model, its common that the TS cannot be fully complete prior to development starting with any integration.

### Build

Relationship of the start of build (coding) and the maturity of the TS is a decision to be made before the targeted start of build date. Due to the agile nature of these SaaS projects, it’s common for the TS to be a work in progress along with development. While some aspects, such as field mapping, should be completed prior to build, others such as value mapping and calling API properties will not be clear until development is underway.

Build is a stage where physical activities are being conducted in the applications. Up to this point its paperwork and planning.

### Unit Testing

Unit testing is conducted by the developer with minimal involvement of other resources. This early testing is generally utilizing an SF environment with minimal data. Due to data needs the developer will need to engage with the functional team(s) for support in key entry of data needed for testing cases.

Files created are not intended to be sent to vendor applications and there is no need to coordinate with vendors for this stage.

* Inbound – load files will need to be created manually based on the example version that should be in the TS. Editing of the file by the developers is expected based on the data within the SF environment at the time
* Outbound – files created are held and reviewed locally.

Testing results and any action taken are only documented within the TS. No formal tracking, such as within Quality Center, is expected. This stage is for the confidence level of the developer.

Note: for more information about any testing refer to the project’s Testing Strategy.

## String Testing

String testing is to establish and validate the connectivity with the vendor’s environment. Up to this point no integration has been exchanged with the vendor applications. At this stage the results are only looking to validate connectivity, not the format or contents of the integration package (mostly files but could be other methods). While each vendor has been engaged for design and planning purposes, this is the first stage when some positive validating is expected from the vendor, that being a positive connection.

Note that in many cases this string test is for a testing/quality environment. Additional string testing for production environments will be needed at the go-live phase.

## Formal Testing

There will be cycles of formal testing as defined in the project’s Testing Strategy. How many, entry exit criteria, and results tracking will be defined in the project’s Testing Strategy.

### Tester

For this document the critical difference is the resource participants. The developer must not be the tester for formal testing. As such there will need to be planning to educate the assigned tester on how to execute the assigned integrations.

### Vendor Validation

The requirements for the formality of the vendor sign off will be defined in the project’s Testing Strategy. Regardless of the signoff, at this stage there will be a need for tight coordination with the vendor for integration exchange. By this point the lead time and testing windows with each vendor must be understood. If vendor environment refreshes are needed, they should be completed by this time. This stage has the higher bandwidth need from the Vendor Coordinator.

Experience speaking, this dependency and coordination needed with external project resources is the greatest risk to the project timeline.

In some cases, the vendor will not have a testing/quality environment. In these cases, the file created may not actually be validated by loading. Strategy for how the file will be validated is needed before the build phase as alternate file handing based on the testing phase will need to be included.

# Critical Design Aspects

## Design Decisions

While each interface build is based on their unique requirements, there are two topics that should be standard across all interfaces: value translation and notifications.

### Value Translation

Value mapping is where the SF contents of a field need different values at the receiving side and need a translation in flight. Reverse for inbound integrations. The decision on where to house translation tables should be based on how support is desired. Who should manage the value translation table and where that table should reside for access and controls should be the primary factor. Ideally no coding should be needed for value changes.

Recommendation is that the tables are custom Metadata Framework (MDF) objects in SuccessFactors. The translation logic for new or existing requirements can be maintained by a business owner. The table can be extracted real time via an API call during the interface execution. Knowing the need for these MDF objects in advance is critical as the functional teams will need to know to create these as part of an integration soliton.

No integration translations can be considered a final design until the Data Team has a final data migration design.

### Notifications

Before the build phase starts there needs to be an understood strategy on notifications during integration execution. Notifications can be on technical failures, bad data conditions, or just general status. This strategy will help with consistency for better post go-live support. Additionally, the targets of notifications are different in the testing phase as they would be in a production environment. How to support testing notification needs to be designed and built into the interfaces.

## Effective Dates

While some organizational and employee history will migrate from SAP HCM to SuccessFactors, not all data elements and existing SAP HCM start dates will be migrated. An understanding of the Data Migration Strategy will be needed to successfully design the interface for the first execution and to be aware of vendor impacts.

At the point of go-live, many data elements will be migrated with the values that exist in SAP HCM at that time, except for the effective date. The effective date loaded into SF will be that of the go-live date. The ‘new’ effective date in SF could be an impact to some vendors, even when the file format and all other data are exact replications.

Example, assuming 05/01/2021 go-live date: In SAP HCM an employee has a row of data that was effective as of 10/25/2020. The start date / effective date to be migrated into SF will be 05/01/2021. The effective date of that row is not what was in SAP HCM and could impact vendors when sent on full files where 05/01/2021 date is a new date.

## Testability

The two main areas for understanding are that of vendor testing/quality environments (section 5.6.2) and notifications (section 6.1.2). See the applicable sections for details.

This section reiterates the importance of understanding the needs that will be different for non-production environments than what is needed on the production environment.

## Future Dated Rows

Based on the vendor’s requirements, some integrations will pass data as it is created/changed, others will only want based on the effective date. Each developer will need to understand how the current dated rows are extracted via API vs how rows are extracted based on time stamp. As stated in section 5.4.1, some solution aspects are discovered when working with the new applications. Also, note that SAP is still continually upgrading features with their APIs.

## Sequence of Events

Likely the greatest difference with how SAP HCM operates and that of how SF operates, in context applicable to integration, are the sequence of events with data entry. While both will have an action called ‘Hire’ the data entered under that event can vary with the sequence.

An example could be:

* In SAP HCM there are 30 data points that are entered during the ‘Hire’ event
* In SF there are 20 data points entered during the ‘Hire’ event, then right after a user will enter the remaining data as a ‘Data Change’ event

While at the end of the day/week when the interface is executed, all the data expected has been created, the last event in SF will be ‘Data Change’. Some vendors will not be able to process the data unless the event passed to them is ‘Hire’.

Understanding this difference in sequence of events and what the vendor expects, where event types are critical, as with most payroll applications, is critical to the result of the interface. But based on the nature of the agile methodology, while the data fields and field values can be established early, these sequences of events are often not clear until formal testing is underway.

## Calculated SuccessFactors Fields

SF may show a calculated value on an end user screen but rarely stores the calculated value. Where needed the calculation logic must be repeated in the integration code and attention paid to future SF maintenance of the logic.

## Server Times

Often its observed that the servers where the interfaces are executed are not the same time as the servers where SF data resides. When making decisions based on time stamps, the developer many need to consider differences in the server times.

## First Run

Some integrations will have some unique needs that are only applicable for the first / first few, executions in the SF production environment.

A few examples:

* The execution of the interface includes interaction with previously created files – those preexisting files may need to be manually created.
* The effective dates of the migrated data need to be considered.
* The action type of the concerted data needs to be considered as generally it’s a place holder action to make the migration date and not one expected by vendors.

# Integration Inventory

An approved scope of existing interfaces to replicate and new interfaces to be built.

Type 5si.e. any integration not directly fed by types 1-4 but where HR data flows, one node downstream are only to be added to the inventory for impact analysis. As these are one node down from direct HR integrations, and often owned by non-HR organizations, they are hard to become aware of. Also, there is no intention to make any changes other than remediation as design changes could cause impacts.

For tracking and management purposes, any item in the inventory will represent one side only of an integration. Integration where the project has both sides in the new scope will have two items in the tracker, indication as ID xxxxA and ID xxxxB for linkage.

## Inventory

For real time list refer to Integration Inventory - Section Appendix 10.1. In addition to technical details, the inventory needs to define the TENNECO resources that represent each interface. Some of them have already Identified.

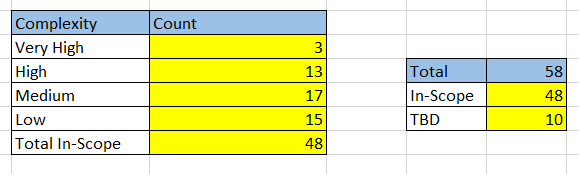
Specific resources:

* Business owner
* IT Owner/Technical SME
* Vendor Manager/Contact

The inventory will be utilized to track the progress of the various stages of the development lifecycle.

## Interface Counts

As of 4/21/2021



Note: Complexity ratings and estimations are based on IBMs benchmarking of SF projects and Biztalk development standards.

# Roles and Responsibilities

The section describes the R&R for the interface’s development phases. Note that pre-development roles and go-live support are not included. The go-live R&R will be defined when the go-live strategy and plan are created.

## Development Roles and Responsibility by Phase

For details of the dates regarding the development phases, see the project timeline and/or project plan.

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Design Phase** | **Build Phase** | **Test Phase** |
| Tenneco Vendor Coordinator | Vendor management strategy and communication plan | Coordinate with vendors for answering questions  Coordinate for upcoming testing phase | Coordinate with vendor to ensure timely support of project testing |
| Tenneco Business Owner | Providing input to FS, review, approval  Contribute to impact analysis | n/a | Preparation and execution of testing, creation of defects, providing test passage |
| Tenneco IT Application Owner | Establishment of policies, infrastructure, and user access management | Quality control | Production migration and move from project to support mode |
| Tenneco Technical Lead | Managing and status reporting for the phase | Managing and status reporting for the phase  Managing resolution of issues | Managing and status reporting for the phase  Managing resolution of issues  Contribute to cut-over planning |
| IBM Integration Team Member | FS creation and review | TS input, review, approval | Support testing for interface execution and file handling |
| IBM SAP HCM Consultant | FS creation and review Coordinate with EC Team | SAP Configuration Mapping & Replication Owner | Support testing for interface execution and file handling |
| IBM Developer | Providing input to FS  Receiver of completed FS | TS creation  CPI / ABAP code development, SAP configuration  Unit testing | Support testing for interface execution and file handling |
| IBM EC Functional | Providing input to FS, Identify design changes | Providing input to TS  Supporting data needs for unit testing Design Impact resolution | Providing input for test preparation and execution  Design Impact resolution |
| IBM Integration Lead | Managing and status reporting for the phase | Managing and status reporting for the phase  Managing resolution of issues  TS approver | Managing and status reporting for the phase  Managing resolution of issues  Contribute to cut-over planning |

## Segregation of Duties (SOD)

By design, SAP’s SuccessFactors is an ‘agile’ and ‘flexible’ application and implementation. As such, the staffing model covers little SODs. Creation of FS/TS inherently has SODs with IBM staffing. Formal testing, as owned by Tenneco inherently has SODs. The build through production deployment does not have any SODs considered. Any required SOD needs to be made aware to IBM or could later impact the staffing model.

# Existing SuccessFactors/SAP HRIS/ERP Integrations - TBD

As there are many modules currently in SF with existing integrations, there could be new integrations, changes to existing, or retiring integration to manage. Similarly for SAP HRIS Applications.

Some of them have been listed in Integration Inventory as TBD Items – Total 10.

Any of the above not known at the time of the integration inventory scoping will be determined by the Change Management process.

# Appendix

## Interface Inventory



## Specification (FS/TS) Template



## Tenneco Interface Tracker Template

