SAP SuccessFactors

Implementation Design Principle

PUBLIC

SAP SuccessFactors Employee Data: Integration Best Practices and Considerations

Ybrant Partners

THE BEST RUN SAP

Document Details

Name	Objective	Audience
SAP SuccessFactors Employee Data: Integration Best Practices and Considerations	This document provides recommendation on Integration patterns, SAP SuccessFactors Employee Central APIs, Security aspects while Integrating Employee Central with SAP and Non-SAP Systems	SAP SuccessFactors Customers: IT and HR professionals. SAP SuccessFactors Implementation and Software Partners: Solution Architect, Consultants and Developers

Change Log

Version	Date	Description
1.0	07.04.2022	Initial version

Supported Releases

Product	Release - From	Release-Valid till
SAP SuccessFactors Employee Central	H1 2022	

Contribution

Role	Name	Organization
Author / Owner	SAP SuccessFactors Product Management	SAP SE
Author	Sachin Kumar	Ybrant Partners

Implementation Design Principle (IDP) are documents that complement existing implementation handbooks by addressing real-life implementation challenges as well as frequently asked questions. They are best practices verified by the SAP SuccessFactors product in collaboration with our experienced implementation partners and SAP services. IDPs will give structured guidance to address challenges via product configuration and/or provide workarounds to avoid typical implementation pitfalls. Some of the guidance especially technical solutions may require custom development which may require partner support.

The recommendations in this document are based on the features and functionality available up to SuccessFactors release at the time of writing. Future functionality can impact the recommendations provided by this document. We strive to keep these recommendations up-to-date, however, in case you find that a recent functionality has not yet been considered in the latest version of this document, please send an email to SAPSuccessFactorsIDPDoc@sap.com. In addition, the reader is advised to read and familiarize with essential and additional product-related documentation which includes Implementation Guides, SAP Notes, SAP Knowledge Base Articles, and additional assets as referenced in this document, see chapter 8.

TABLE OF CONTENTS

1 TERMINOLOGY

The table below explains some abbreviations used in this document.

Abbreviation	Description
EC	SAP SuccessFactors Employee Central
SAP BTP	SAP Business Technology Platform (SAP BTP) brings together intelligent enterprise applications with database and data management, analytics, integration, and extension capabilities into one platform for both cloud and hybrid environments, including hundreds of pre-built integrations for SAP and third-party applications.
MDF	Metadata framework (MDF) is a framework in SAP SuccessFactors to create own code free custom entities including persistency, UI and APIs. MDF is also used internally by SuccessFactors. In this document we sometimes distinguish between entities and APIs based on MDF and Employee Central to highlight additional features which are available for the corresponding class of entities
RBP	Role Based Permissions
UI	User Interface
PERNR	SAP Personnel Number
SAP ECP	SAP Employee Central Cloud Payroll
Talent Suite / BizX	SuccessFactors Talent Suite without Employee Central
SOAP	Simple Object Access Protocol (SOAP) is a messaging protocol specification for exchanging structured information in the implementation of web services in computer networks. (SOAP, 2019)
REST	Representational state transfer (REST) is a messaging protocol that defines a set of constraints to be used for creating Web services. Web services that conform to the REST architectural style, called RESTful Web services, provide interoperability between computer systems on the Internet. RESTful Web services allow the requesting systems to access and manipulate textual representations of Web resources by using a uniform and predefined set of stateless operations. (Representational state transfer, 2019)
OData	OData (Open Data Protocol) is an ISO/IEC approved, OASIS standard for a messaging protocol that defines a set of best practices for building and consuming RESTful APIs. OData helps you focus on your business logic while building RESTful APIs without having to worry about the various approaches to define request and response headers, status codes, HTTP methods, URL conventions, media types, payload formats, query options, etc. (OData.org, 2018)
Pagination	There are different paging options possible to avoid that API calls result in timeouts. Those are client sided paging and server sided paging. In the case of client sided paging the client (initiator of the communication) tells the server which page and how many records of the page to return, while in case of server sides paging this is done by the server.
Identifier	Identifier in this document refers to language independent, “Property Name” in OData API Data Dictionary or “Identifier” in the “Manage Business Configuration” Screen in SAP SuccessFactors system.
Person	Reference to an actual human person
Worker	A worker in the context of this document will refer to both Employee and Contingent Worker
Contingent worker	Contingent worker is where workers are provided by a staffing firm/vendor to the company for which work is performed and are employees of the staffing firm.
Employee	An employee is a full-time worker in company. Sometimes an employee is defined as a Full Time Equivalent (FTE).
Expatriate	A person who lives outside their native country.
Employee ID	Unique identifier for an Employee
Portlet	A portlet is a configurable object on the SuccessFactors screen
SDM	SAP SuccessFactors Succession Data Model
LMD	Last Modified Date
Source	SAP SuccessFactors is the system of record for Employee data
Target	SAP or Non-SAP system where the Employee data gets replicated to
CI	Cloud Integration (CI) is a capability in the SAP Integration Suite, CI was formerly known as SAP Cloud Platform Integration (CPI)

2 ABSTRACT

SAP SuccessFactors Employee Central is a Core HR Information System which enables businesses to manage/maintain Organisation, Job & Pay Structure data and Person & Employment data. This Implementation Design Principle describes best practices for extracting data from Employee Central in addition to security considerations when exchanging the data between SAP and Non-SAP Systems. It also explains the choice of Employee Identifier depending on the downstream system and landscape.

3 INTRODUCTION

SAP SuccessFactors Employee Central consists of Foundation objects (Position, Location, Division, Legal Entity, Department, Division, Business Unit) and Worker Data related objects (Person, Personal Information, Employment information, Job Information, Compensation information). The foundation objects form the Organization structure and acts as a framework on which the Employee information is built.

The scope of the document addresses how to extract Employee data from SAP SuccessFactors Employee Central to different systems. The middleware technology in this document is “Cloud Integration”. This document assumes that the reader is building an integration between SAP SuccessFactors Employee Central and a 3rd party system.

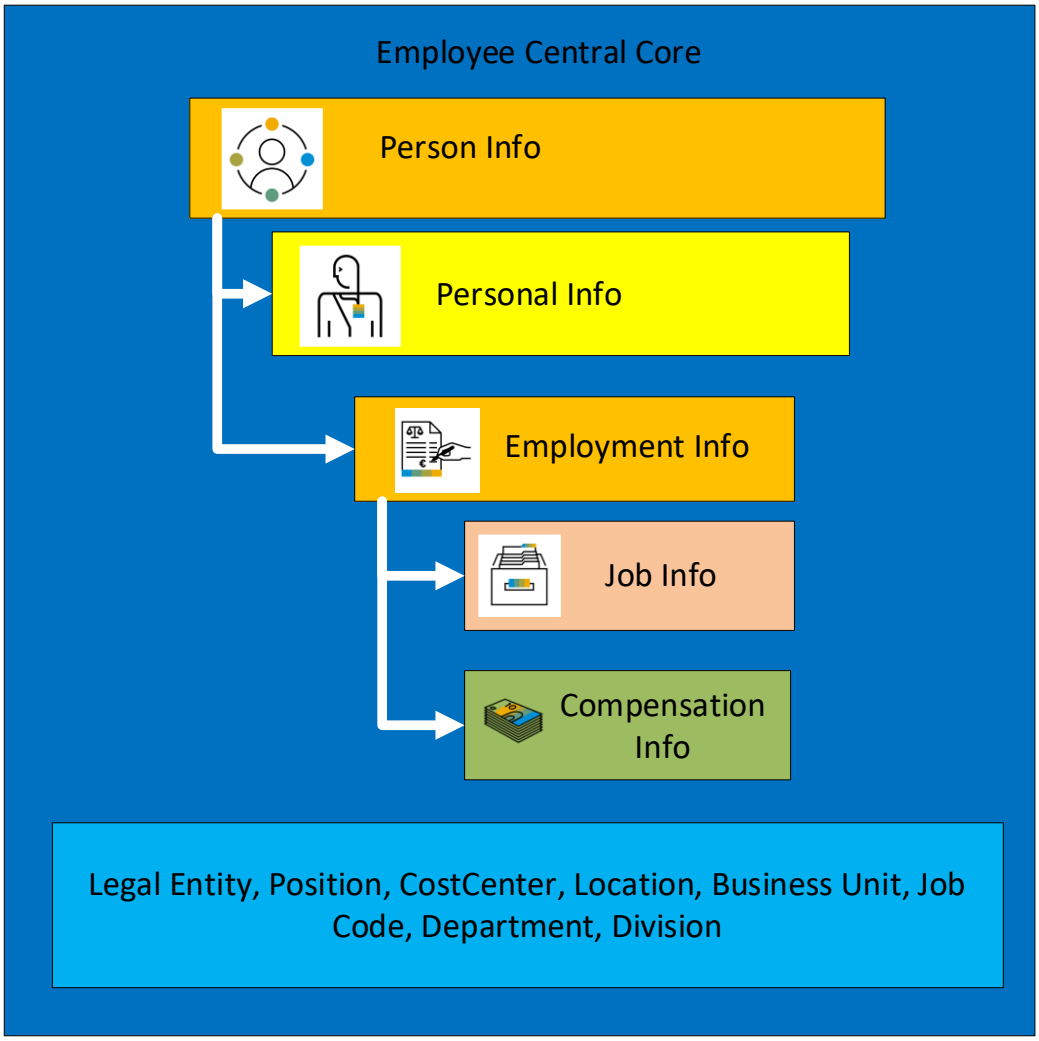


Figure: Data points in SAP SuccessFactors Employee Central Core

4 BUSINESS REQUIREMENT

Target systems which need employee and organization data vary based on:

1. Person, Employment and Compensation information
2. Person and employment data without Compensation information
3. Person Identifier data (Basic data)

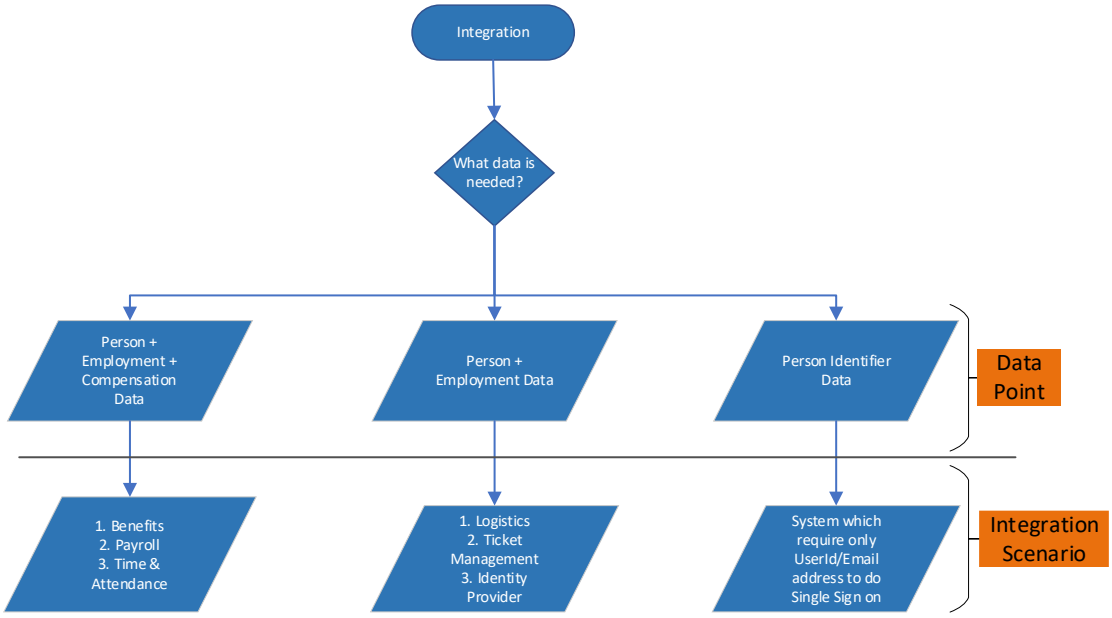


Figure: Employee Data Integrations based on data requirement

4.1 Functional Requirements

Following are the key requirements to be discussed during the design phase of the project.

- Choosing the right Employee Identifier (Refer to section 5.1)
- Replicating Employee data at Employment level (Refer to section 5.2)

4.2 Technical Requirements

4.2.1 Effective-Dated vs Non-Effective-Dated information

SAP SuccessFactors Employee Central (EC) has two kinds of portlets, which can store the Employee Data (Refer to section 5.1.3 for solution overview on this topic in this document)

- Effective dated portlets
- Non-Effective dated portlets

Effective Dated portlets can store history information of the employee data in EC, whereas non-effective dated portlets can only store the snapshot information/single time slice of the Employee Data in EC.

4.2.2 SAP SuccessFactors Integration capability

SAP SuccessFactors offers standard Webservices (APIs) and in-built data extraction tools. SAP SuccessFactors Employee Central supports SOAP and ODataV2 based APIs for data extraction.

- SAP SuccessFactors APIs (in-built)
- SAP SuccessFactors Event Mechanism (in-built)
- SAP SuccessFactors Integration Tools

4.2.3 Data Extraction Summary

Employee data extraction can be classified into:

- Schedule based
- Event based

Note: Employee Data Replication from SAP SuccessFactors Employee Central to target systems cannot just rely on event based integrations. It should be a combination of scheduled and event based mechanisms.

5 SOLUTION OVERVIEW AND CONCEPTS

5.1 Employee Identifier

Employee Identifier in SAP SuccessFactors is used to identify employee and the related employment(s). Deciding the Employee Identifier is a crucial step at the start of the integration requirements gathering based on the target system. SAP SuccessFactors has the following IDs

Target Employee Identifier Requirement	SAP SuccessFactors Employee Identifier	Characteristics	Additional Comments	Field Availability in the SAP SuccessFactors API
Unchangeable key/Identifier at person level	UUID (per-person-uuid)	This Employee Identifier is at person level and remains un-affected, even if the employee undergoes termination and Employee is re-hired with new employment. (System generated and is alphanumeric)	This is available at person level and cannot be used for distinguishing multiple employments in the SAP SuccessFactors system. This is available in an EC configured SAP SuccessFactors system only.	CompoundEmployee - Person Root Segment SFOData.PerPerson
A changeable key/identifier at the person level which can be system generated or created through UI/API	PersonIdExternal	This Employee Identifier is at Person level. Generated via SAP SuccessFactors UI when hiring an employee (Alphanumeric) or can be generated via API "generateNextPersonID"(system generated sequence and numeric). One employee throughout the Hire to Retire lifecycle can have only 1 Person ID External.	This is available at person level and cannot be used for distinguishing multiple employments in the SAP SuccessFactors system.	CompoundEmployee - Person Root Segment SFOData.PerPerson, SFOData.PerPersonal, SFOData.EmpEmployment, SFOData.EmpEmploymentTermination

unchangeable identifier at the employment level	userId	This Employee Identifier is at Basic User information level. This information is available in both EC and Non-EC systems. UserId, once created in the SAP SuccessFactors, cannot be changed. UserId can be alphanumeric. A single Employee in the SAP SuccessFactors can be associated to different UserId (Global Assignment/ Concurrent Employment)	This ID changes with Employments. Hence, choose this ID as the Employee identifier in target system only if the Employee records must be differentiated in the target system at Employment level.	CompoundEmployee - Person Root Segment, Employment segment (SFOData.User, SFOData.EmpEmployment, SFOData.EmpJob) and related navigations
An identifier that can be generated/ changed at the employment level	Assignment ID (assignment_id_external)	This Employee Identifier is at Employment level. Assignment ID can be same as UserId (default) or can be generated via Business Rules by using a sequence number in SAP SuccessFactors which can follow the target system requirements.	There is no UI screen in SAP SuccessFactors to edit the Assignment ID. **Note**: Sample API call to change the Assignment ID in SAP SuccessFactors refer to 2820644 - [1908 Feature] API-12250: Add new attribute assignmentIdExternal in User OData API (sap.com)	CompoundEmployee - Global Assignment Segment, Employment Segment, SecondaryAssignmentsItem segment SFOData.SecondaryAssignmentsItem

Table: Choosing Employee Identifiers in SAP SuccessFactors

Note:

- For more details, please read this section in the SAP Help guide: Differentiating Between Person ID, UUID, User ID, and Assignment ID - SAP Help Portal
- Before considering implementation/adoption of AssignmentID, please read the areas of impact and areas where Assignment ID is not supported: Important Notes About Assignment ID - SAP Help Portal

5.2 Employment types in SAP SuccessFactors

SAP SuccessFactors Employee Central helps you manage full-time employees and Contingent Workforce Management capabilities.

Please refer to the Key Advantages of Using Contingent Workforce Management in Employee Central

SAP SuccessFactors supports multiple employments and contingent employments. Hence, to differentiate these scenarios worker types and employment types are introduced in Employee Central.

We recommend you understand the concepts of Concurrent (multiple) Employment and Contingent Worker in SAP SuccessFactors.

- Events: Events are changes that happen during the different stages of an employee's lifecycle from hire to retire. The event sets the user status. Please read here for more information.

When contingent workforce is enabled in SAP SuccessFactors Employee Central, the following events are applicable:

- SCWK (Start contingent worker)
- ECWK (End contingent worker)

Note: When a contingent worker record is created (Hired in Employee Central) a workorder is assigned to userId/Employment; start record and end record are created.

- Assignment Class: An Assignment class is used to differentiate employments. "assignmentClass" cannot be updated via API and is a picklist type in SAP SuccessFactors. The value of this picklist will depend on the type of employment created in SAP SuccessFactors. The assignment class can take one of these values:
  - ST: Standard
  - GA: Global Assignment
  - PP: Pension Payout
  - BN: Beneficiary
- Employee Status: Indicator if an employee is active or inactive
- Employment Status: Indicator if the employment is still active or obsolete
- Secondary Assignments: Concurrent Employment (a.k.a Multiple Employments) scenario is based on secondary assignments. The object/API in the system is "SecondaryAssignments" - This entity is used to differentiate primary and secondary employments replication. Please read here for more information.
- Work Order: A Work Order is a stipulated contract between the vendor providing the contingent workforce and your company. Please read here for more information.

Supported Employment Types:

- Regular Employment:
  All employees have at least one regular employment contract with the organization to which the employee is hired and goes through his employee life cycle till termination. Note that a special variant of a regular employment is leveraged for contingent workers.

- Global Assignment:
  An employee on a global assignment has two employment assignments, a Home assignment (which can be marked dormant) and an active Host assignment (expatriate record). The Home assignment is activated, and the host assignment is terminated upon completion of the global assignment.

- Concurrent Employment:
  An employee on a concurrent employment has two or more active employments and occupies two positions at the same time. One of them is designated as the Primary employment. There is a valid contract for each of the employments. When a contingent worker works on multiple work orders, each work order is assigned with a new employment

- Contingent Employment:
  Contingent workers are non-permanent who deliver certain services to the company. Contingent workers are also managed in Employee central so that the managers can have a uniform access to employees and contingents. (Please refer implementation guide for more details).

To understand more about the Employment types and how to use multiple employments for specific scenarios like international transfers please refer to the Implementation Design Principle **Employee Central: Managing Employments in SuccessFactors Suite**

Please refer to section 6.1 for understanding the characteristics of Global Assignment and Concurrent Employment.

5.3 Effective Dated and Non-Effective Dated portlets and APIs in SAP SuccessFactors EC

The following table lists the EC portlets and the corresponding OData APIs and CompoundEmployee API segments respectively.

EC Portlet Label	Corresponding OData API	CompoundEmployee API Segment	Effective Dated?
Biographical Information	SFOData.PerPerson	Person	No
Personal Information	SFOData.PerPersonal	Personal_Information	Yes
National ID Information	SFOData.PerNationalId	national_id_card	No
Contact Information	SFOData.PerEmail, SFOData.PerPhone, SFOData.PerEmergencyContacts	Email_Information, Phone_information, emergency_contact_primary	No
Addresses	SFOData.PerAddressDEFLT	Address_Information	Yes
Dependents	SFOData.PerPersonRelationship	dependent_information	Yes
Payment Information	SFOData.PaymentInformationDetailV3, SFOData.PaymentInformationV3, SFOData.PaymentInformationDetailV3 <countrySpecific>	payment_information	Yes
Position Information	SFOData.EmpJob, SFOData.Position	job_information	Yes
Job Information	SFOData.EmpJob	job_information	Yes
Employment Details	SFOData.EmpEmployment	employment_information	No
Job Relationships	SFOData.EmpJobRelationships	job_relation	No
Alternative Cost Distribution	SFOData.EmpCostDistribution, SFOData.EmpCostDistributionItem	alternative_cost_distribution	Yes
Compensation Information	SFOData.EmpCompensation	compensation_information	Yes
Recurring Payments and Deductions	SFOData.EmpPayCompRecurring, SFOData.RecurringDeduction, SFOData.RecurringDeductionItem	paycompensation_recurring, deduction_recurring	Yes
Pay Component Non-Recurring	SFOData.EmpPayCompNonRecurring	paycompensation_non_recurring	No
Recurring Deduction	SFOData.RecurringDeduction, SFOData.RecurringDeductionItem	deduction_recurring	Yes
One Time Deductions	SFOData.OneTimeDeduction	deduction_non_recurring	No

Table: Effective dated vs non-effective dated EC portlets and APIs

(Refer to section 6.2 in detailed overview for understanding when to use User entity and EC entities)

5.4 SAP SuccessFactors APIs

5.4.1 SFAPI (SOAP based, Deprecated as of August-2018)

SOAP based APIs are deprecated in SAP SuccessFactors technology stack, the only exception is CompoundEmployee API (Deprecation of Partner API, SFAPI Adhoc, and SFAPI for Simple Entities - SAP Help Portal)

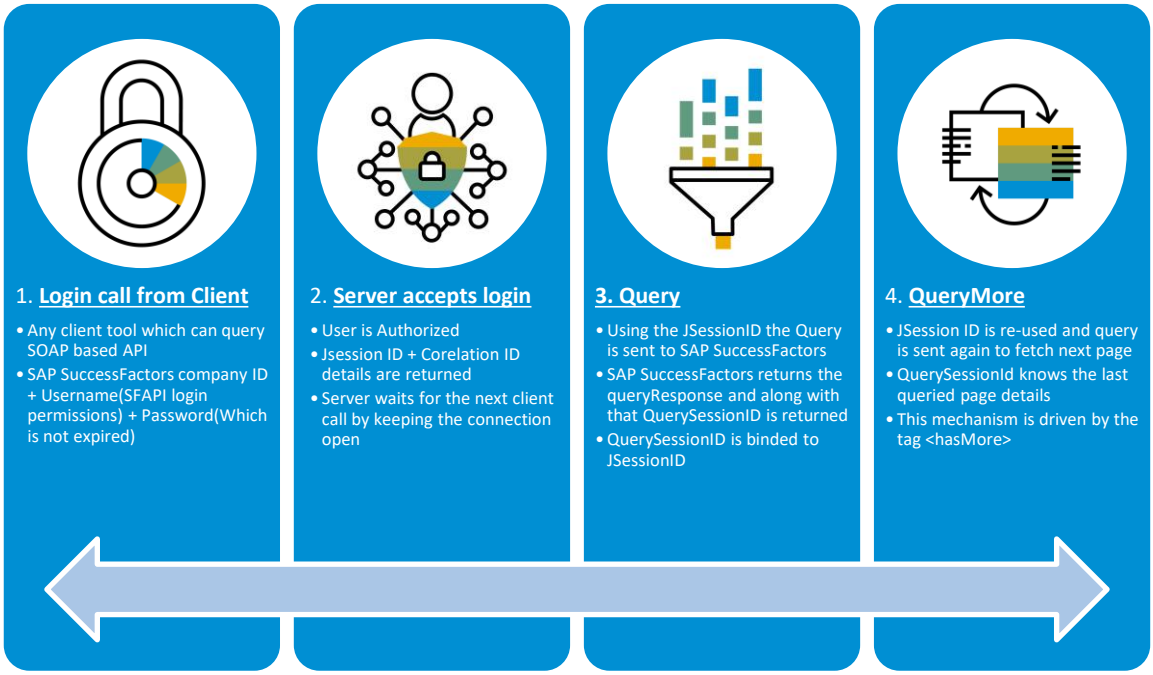


Figure: Login mechanism for SAP SuccessFactors SFAPI

Key Features of CompoundEmployee API:

- CompoundEmployee API operates at segment level (Entity) and not at field level (By default all the fields configured for a respective segment are returned by the API if a value is supplied)
- Please make the Provisioning Settings for CompoundEmployee API to use CompoundEmployee API in SAP SuccessFactors Integrations.
- The CompoundEmployee API selects multiple entities related to an employee and returns data in a hierarchical-structured response XML with Person data as the root node.
- Employees without Employment information are not returned in the response (Example: Technical users)
- The employee data which is approved in the Employee Central is returned by the API. (When the created Employee Data has a pending workflow associated to it, data is not returned)
- CompoundEmployee API supports ascending and descending (default) sorting by start date for **effective-dated entities**.
- Operations supported: Query, QueryMore
- By default, API returns only 200 rows per query (If batch size is not specified)
- Maximum SOAP message size (HTTP content-size) cannot exceed 5MB per batch
- Supports a wide range of query filters and parameters - **Overview of query parameters**

5.4.2 OData API

SAP SuccessFactors recommends OData APIs for integration purposes. OData API supports CRUD (Create, Read, Update, Delete) operations and follows OData V2 specification. The APIs by default return As-Of-Dated information and supports querying history/future dated records.

Note: SOAP APIs are currently deprecated and will not be enhanced.

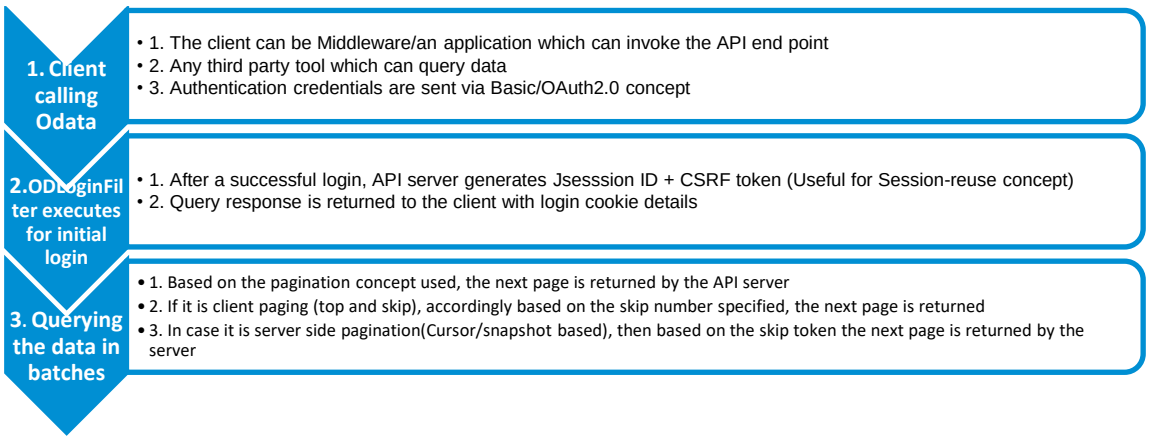


Figure: General Login flow for SAP SuccessFactors OData API

- To know more about SAP SuccessFactors OData APIs capabilities, please refer to the developer guide - SAP SuccessFactors HXM Suite OData API: Developer Guide
- To know more about SAP SuccessFactors OData API functionalities, please refer to the reference guide - SAP SuccessFactors HXM Suite OData API: Reference Guide

Note: Please read through the following Implementation Design Principle document for better understanding of the SAP SuccessFactors API capabilities for building integrations - SAP SuccessFactors Integration: Best Practices using SAP SuccessFactors APIs for Custom Integrations

5.4.3 SAP SuccessFactors Integration Tools

Please refer to the SAP SuccessFactors Integration Tools | SAP Blogs

5.5 Pagination

Please read the Pagination - SAP Help Portal for different types of pagination supported by SAP SuccessFactors APIs

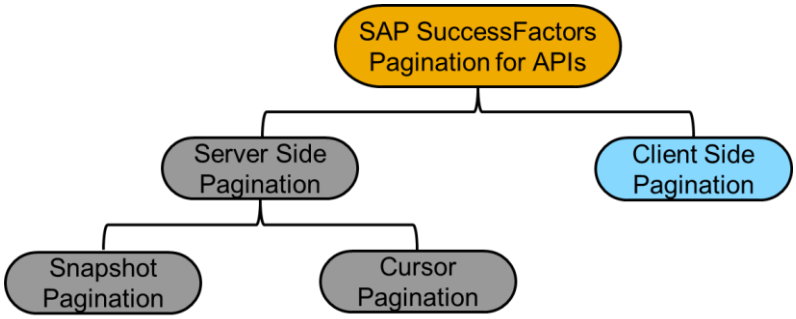


Figure: SAP SuccessFactors supported pagination overview

Client-Side Pagination:
Client-side pagination is supported by all APIs in SAP SuccessFactors. The flexibility with client-side pagination is - you can query any page based on the $top and $skip parameters. Including the “orderby” clause is mandatory - SAP KBA 2272937.

Cursor Pagination:
Cursor based pagination is only supported in Employee Central EmpJob entity, User entity, MDF Generic Objects, and most EC Foundation Objects. Hence, this is not a recommended approach for Employee Data replication. Advantages and disadvantages of cursor-based pagination Click here.

Snapshot Pagination:

In general, the following entities support snapshot pagination:

1. Entities based on succession data model elements.
2. All FO (Foundation Object) entities except FODynamicRole.
3. All EC2MDF entities.
4. Workflow related entities including EmpWfRequest, WfRequest, WfRequestComments, WfRequestParticipator and WfRequestStep.

If the integration has re-try logic and uses Snapshot pagination, then a retry will always start from the 1st page.

Advantages and disadvantages of snapshot-based pagination Click here

Note:

- How Do I Choose? - SAP Help Portal
- CompoundEmployee by default works on Snapshot-Based pagination. No support for cursor/Client-side paging.
- To avoid missing of records/duplicates when doing full loads using OData APIs, it is recommended to use Snapshot Paging – SAP KBA 2723468
- Batch Operation in OData V2 Adapter in SAP Cloud Platform Integration | SAP Blogs
- Batch Request with multiple operations on multiple entity sets using Cloud Integration OData Adapter | SAP Blogs
- Making SAP SuccessFactors and Cloud Integration more reliable and performant | SAP Blogs

5.6 Security Considerations

- **File based integration:** File based Integration through Cloud Integration/SAP SuccessFactors Integration center is **not recommended** by SAP if the target or 3rd party system has web-services enabled for data ingestion.

- **API based integration:** SAP SuccessFactors recommends using a secure authentication inbound mechanism - OAuth2.0 (Basic authentication for SAP SuccessFactors APIs is deprecated)

5.7 Data Extraction Design Approach

5.7.1 Schedule Based

- SAP SuccessFactors Integration Center
  SAP SuccessFactors Integration Center provides a leaner and faster approach for building integrations which do not require complex message transformations from source to target.

  Note: SOAP APIs are not supported in SAP SuccessFactors Integration Center.

- Cloud Integration
  Cloud Integration (SAP Integration Suite) can be used to extract data from SAP SuccessFactors and offers flexible approach for choosing the APIs (supports both SOAP and OData based), support for complex process orchestration and message transformations.

5.7.2 Event Based

Event based integrations must be evaluated based on scenarios and use cases.

5.8 Error/Exception handling and Monitoring

SAP SuccessFactors has inbuilt Execution Manager dashboard (XM) for raising the Error/Exception to business users. The data in the dashboard can be updated real-time via APIs (EMEvent - SAP Help Portal). Monitoring capabilities of XM include Integration Center related jobs/Middleware Integration.

Error handling can be used for validation/functional errors (missing mandatory segments, missing required fields etc.,).

Exception handling can be used for catching abrupt run-time failures in Cloud Integration process due to heap memory/Apache camel framework issues.

6 DETAILED SOLUTION

6.1 Characteristics of Global Assignment, Concurrent Employment and API Considerations

Parameters	Global Assignment	Concurrent Employment
Employment Information	When an Employee in SAP SuccessFactors is on Global Assignment, home assignment (Current Employment) is inactive and host assignment (New Employment) is active	When an Employee in SAP SuccessFactors is on Concurrent Employment, all the employment assignments are active
Full Time Employee (FTE)	FTE is always 1.0, as the employee works for one employment assignment at a given point of time	Summation of all the employments together is 1.0, as the employee splits the work in multiple employments at a given point of time
Assignment Types in Employee Central	Short-term assignment - An employee who is on Global Assignment for a tenure of less than a year Long-term assignment - An employee whose tenure on Global Assignment is more than an year Note: The criteria for assignment the short and long term assignments varies from customer to customer.	Primary Employment - First Employment created in SAP SuccessFactors when the employee is hired Secondary Employment(s) - Following employments after the first employment Note: The primary employment can be selected/changed after adding secondary employments
Contingent Worker	Global Assignments are only for Regular Employment scenario	When a contingent worker is assigned with multiple employments, there must be different WorkOrders for each employment assignment (Contracts)
API Considerations	CompoundEmployee - Details of Global Assignment for the employee is obtained using the segment "global_assignment_information" which is part of "Employment_Information" segment which contains host Assignment Note: When data is extracted via CompoundEmployee, employee status field will be active. Employee records for global assignment needs to be differentiated on the assignment class and not on the employee status. OData API - Global Assignment information specifically at Employment assignment level is obtained through SFOData.EmpGlobalAssignment	CompoundEmployee - Details of Concurrent Employment for the employee is obtained using the segment "SecondaryAssignments" which is a seperate segment. "SecondaryAssignmentsItem" is an iterating segment in the "SecondaryAssignments", usersSysId field identifies the Employee record. Note: "SecondaryAssignmentPeriod" is part of "Employment_information" segment which has secondary employment(s) details. OData API - Concurrent Employment information specifically at Employment assignment level is obtained through SFOData.SecondaryAssignments, SFOData.SecondaryAssignmentsItem
Assignment Class	Used for differentiating the regular employment (ST – Standard) and Global Assignment (GA)	Assignment class is ST (Standard) for all the employment assignments in Concurrent Employment scenario

Position Data	Employee can work for the same position/different positions, when assigned to Global Assignment, depends on the scenario of the employment assignment	Employee will be working for multiple positions as per the Employment assignment (Contracts)
Employment/ Context Switcher	Applicable for Global Assignment	Applicable for Concurrent Employment

Table: Characteristics of Global Assignment, Concurrent Employment and API considerations

For detailed information refer to the following guide: Managing the Employment Lifecycle (from Hiring to Termination) in Employee Central

Note:

- A global assignment and concurrent employment cannot be applied to the same employee at the same time. This means that, if an employee has a concurrent employment, this employee cannot have a global assignment at the same time, and vice versa.
- There is a field called “isPrimary” under “employment_information” node. This field is deprecated (at least as of Q2 2017) and not related to the concurrent employment feature. This should NOT be used to derive the primary or secondary Employment.
- Tips for Using Concurrent Employment in SuccessFactors Employee Central

6.2 When to consider SFOData.User vs EC API

SAP SuccessFactors APIs are designed, based on the following target system data needs:

- Needs only non-effective dated data as-of-today (Target system cannot understand time slices)
- Needs the full history and future effective dated data (Complete replica of EC Employee Data)
- Needs only partial history and supports partial update of history based on change records (For Example: No future dated records)
    - CompoundEmployee is the **only API** in SAP SuccessFactors, which can output the changed field value and the previous field value in the API response, OData APIs do not have that capability

Note: For the last two options listed above, according to the data needs, choose between EC OData APIs and CompoundEmployee by referring to the IDP SuccessFactors Integrations - Best Practices using SAP SuccessFactors APIs for Custom Integrations (Section 6.1 – Using the right APIs)

Design approach when target system cannot understand time slices using OData Entities:

Criteria	Approach 1 - User Entity	Approach 2 - EC Entities	Mitigations
Approach	User entity, side mark foundation data is having descriptions combined with external codes. Better to join in real codes via navigation to EmpJob	Use EC entities, along with Last Modified Date (Future dated change, will affect LMD, but the data would not be queried because the as-of-date is not present in the range).	
Query Example	{{SFSF_ODataV2_API_domain_URL}}User?$select=firstName,lastName,department,dateOfBirth,hireDate&$filter=lastModifiedDateTime+gt+datetimeoffset'2021-09-27T09:50:11Z'	{{SFSF_ODataV2_API_domain_URL}}PerPerson?$select=personalInfoNav/firstName,personalInfoNav/lastName,employmentNav/jobInfoNav/department,dateOfBirth,employmentNav/startDate&$filter=lastModifiedDateTime+gt+datetimeoffset'2020-01-29T09:50:11Z'&$format=JSON&$expand=personalI	

		nfoNav,employmentNav/jobInfoNav	
Replicate the data based on the relevant field changes	As User entity is a mix of Organizational data and Employee data, many changes on non-relevant fields (not related to employee level data) will change the last modified date field of the user entity, hence there might be many cases where an employee record might end up in replication wen triggered via last modified queries but not relevant for the target system as the fields in the select statement remain the same.	Like User entity, EC entities too can trigger replications to downstream systems, even though the selected fields in the query are not changed.	**In both cases a comparison of data in the source and the target system can be used to avoid not needed replications**
Will records be replicated, when not effective dated records are changed which are not effective as of today	No, HRIS-Sync will not get triggered in case other effective dated records are changed Note: In a system where Employee Central is enabled, Employee Profile (User) gets the data from different portlets in Employee Central, through HRIS sync job. The future dated changes flow to User entity, when they become effective according to the start date.	Yes	
Multiple Employments	UserId's will be different and the related user accounts will be different.	Entities like SFOData.EmpEmployment will have multiple records stores in the SAP SuccessFactors database, Root objects like SFOData.PerPerson has always one record even though multiple employments are present for the Employee in the SAP SuccessFactors system.	
Performance of query	Since there is only one entity involved and usually filters are at the parent level most of the times, the query is simple and fast compared to EC entities	As per the business needs navigations must be included, and sometimes the queries will have deep navigations which can slow down the performance. Note: If there are too many navigations and the OData navigation level is more than 3 (which is not recommended), and filters are used at navigation	

		entity levels too – Recommended pagination is “Snapshot”	
Realtime data without delay	Since HRIS-Sync is a job which syncs data from EC entities to user entity there might be a delay and hence data might not have the most current state. This is only the case for changes to data triggered by other jobs, e.g. imports on SFOData.EmpJob	Data is always up to date (with a few exceptions e.g. position to job information sync)	
Employee Status (Employee lifecycle)	The API field to be considered is “status”. By default, SFOData.User entity returns active employees in the system. For more details, please refer to this SAP KBA 2736579	The API field is “Employee Status” (Pre-defined picklist “employee-status”) in SFOData.EmpJob	**Picklist fields return optionId as default values**

Table: Target system needs only effective dated data

Special handling for latest effective timeslice in SFOData.EmpJob API:

“effectiveLatestChange” field is only applicable to SFOData.EmpJob API for identifying or extracting the latest changes for a given timeslice when multiple changes per day is made.

Sample Scenario:

Record #	startDate	seqNumber	effectiveLatestChange
1	2022-01-01	3	Y
2	2022-01-01	2	N
3	2022-01-01	1	N
4	2022-02-01	2	Y
5	2022-02-01	1	N

- asOfDate query returns #1 or #4 according to the asOfDate value used.
  - To get record #1, query to be used <server>/odata/v2/EmpJob?$select=<fieldsNeededForIntegration>&asOfDate=2022-01-01
- date range query fromDate=2022-01-01, it returns all the 5 records
  - <server>/odata/v2/EmpJob?$select=<fieldsNeededForIntegration>&fromDate=2022-01-01
- Query only the latest changes in all the timeslices of job info, it returns only #1 and #4
  - <SERVER>/odata/v2/EmpJob?$filter=effectiveLatestChange eq 'Y'&$select=<fieldsNeededForIntegration>&fromDate=2022-01-01

Note: Value for "effectiveLatestChange" field is "Y" for the timeslice which has the max seqNumber among the ones with same start date.

Design approach using CompoundEmployee API (operation modes)

- Full Transmission Mode:
  This is used only when the target system does not have the capability to detect changes.
  Each time CompoundEmployee is queried it returns the entire history information of the employee from the date of Employee Hire. Employee Data sent across from SAP SuccessFactors overwrites the target system data.

  Note:
  - CompoundEmployee API has restriction on ‘last_modified_on’ field - Compound Employee API Change to Last Modified On & Snapshot Date 2H 2021 Production Release
  - Use Full Transmission query mode with ‘last_modified_on’ date to restrict to query only changed data. (Only exception is first time data load between source and target systems)

- Delta Mode/ Effective dated delta transmission:
  This mode is recommended when the target system can detect changes based on “Action Code” and accordingly can act on the data. Effective-dated delta transmission is designed for target systems that work in an effective-dated manner. This means, it is assumed that target system stores the time frame (start and end date) when a data record is effective. The action codes are as follows:

Action Code	Description
INSERT	INSERT is communicated if a new record is created. For example, when an employee is hired, all segments are returned with action code INSERT.
CHANGE	CHANGE is sent if an existing record has been modified. For example, the amount of a bonus has been adjusted.
NO_CHANGE	NO_CHANGE indicates that there were no relevant changes on the exposed attributes of the segment since the last replication. In case a segment is effective dated, several instances of this segment might be in the response and each can have a different action code. In cases where one instance has the action code NO_CHANGE, this means that in the time frame of this instance no differences exist compared to the data status at the last replication.
DELETE	DELETE means that a record has been deleted completely. For example, paycompensation_non_recurring information has been removed.

Table: Action Codes for CompoundEmployee Delta Query Mode

Characteristics of delta mode:

- The action codes are applied to effective-dated segments as well as to non-effective-dated entities. In case of effective-dated segments, the consumer should be aware that the action code doesn’t always resemble the action that has been performed on the UI.

- Period Delta Mode:
  This is used when the target system can detect the changes based on action codes and needs data only within certain interval, period-based delta transmission is intended for target system that are not able to deal with effective-dated objects. Deltas contain retro-active changes and effective-dated objects that are relevant for the given period.

Note:

- For more details and to understand the CompoundEmployee API completely, please refer to the guide - Employee Central Compound Employee API
- Please note that the XSD structure for delta/periodDelta query modes differ from the Full Transmission and snapshot query mode - Generation of consumer interface structure file for Compound Employee API
- Please pay attention to the request body of the CompoundEmployee used in the actual query step, the XSD generation should have the exact selection segments and the result options values passed, mismatch between the actual query and the XSD generation will result in the failure of queryResponse parsing in the middleware/ETL tool.
- XSD structure format With/Without “renderPreviousTags” parameter - Generating CompoundEmployee XSD structure using “renderPreviousTags” resultOptions parameter | SAP Blogs
- **Effective-dated and period delta transmission**
- **Delta Transmission Mode**
- **Supported Fields in where clause**
- **Overview of query parameters**
- SAP **KBA 2215682 SuccessFactors API URLs for different Data Centers**

6.3 File based approach and API based approach security considerations

File based Integration considerations:

- SAP Customer owned or 3rd party provided FTP:

- FTP (Port 21) is not a recommended protocol for sensitive data like Employee Demographic file, consider encrypting (using public key) the file feed (both, during transfer and at rest).
- Signing the file feed by source using private key is recommended to verify the authenticity.
- SAP SuccessFactors Integration center (Integration Center>SFTP Settings) do not support File Transfer Protocol (FTP).
- Cloud Integration supports File Transfer Protocol (FTP) - Cloud Integration – Connecting to FTP(S)-servers using the FTP Adapter | SAP Blogs
- Setting Up Inbound SFTP Connections (Details) - SAP Help Portal
- Cloud Integration - SFTP Adapter – SAP Help Portal

- SAP SuccessFactors provided SFTP/SAP Customer owned or 3rd party SFTP:
  - Beginning August 1, 2014 SAP SuccessFactors provided SFTP server will support only SFTP for Integration and HTTPS for the accessing via Web-browser client (KBA 2088013)
  - SFTP (Port 22) protocol is considered secure (compared to FTP) for sensitive data like Employee Demographic file (Developer/Organization user can still choose to encrypt/sign the file based on the security policy followed/defined in their organization) - Using SAP SuccessFactors Integration Center with file Decryption and Encryption (PGP) – Outbound and Inbound samples
  - Consider using asymmetric key pair for authentication (SSH keys) – Recommended method for authentication
  - Cloud Integration - How to encrypt/decrypt XML payload with AES256-CBC and RSA Algorithm in Cloud Integration | SAP Blogs , Encryption using AES GCM iaik sampler
  - Apply Security - Apply Message Level Security - Sign and Encrypt | IFlow | SAP API Business Hub – Cloud Integration Exemplar Content
  - Cloud Integration supports SFTP in the middle of the process flow too – Use Poll-Enrich with SFTP Adapter | SAP Blogs
  - SAP SuccessFactors Integration Center provides flexibility to change the SFTP port (SFTP port is by default port 22. This should only be changed if there is an alternative SFTP port due to your security requirements)
    - To add an own SFTP the server IP addresses must be added to allow access from the SuccessFactors data center. This KBA 2659632 explains necessary steps.

API based Integration considerations:

OAuth 2.0:

SAP SuccessFactors supports OAuth2.0 framework for authenticating APIs

Points to be noted:

1. Bearer token is issued always by SAP SuccessFactors OAuth2.0 server for accessing the APIs of SAP SuccessFactors from external systems.
2. The logic of token validation and SAML Assertion expiration checks are mandatory if you are building a new connector to SAP SuccessFactors and authenticating via OAuth 2.0 (The checks are implemented within the tooling of Cloud Integration by default)
3. SAP SuccessFactors OAuth2.0 server implements RFC-7522 specification (grant_type:urn:ietf:params:oauth:grant-type:saml2-bearer) for inbound authentication from external systems.

Note:

- The OAuth2.0 framework is common to SFAPI (CompoundEmployee API) and OData V2/V4 version of SAP SuccessFactors APIs.
- SAP SuccessFactors HXM Suite OData API: Developer Guide (V2) – Please familiarize yourself with the documentation.
- SAP Integration Suite supports OAuth 2.0 in SAP SuccessFactors connector for ODataV2 and SOAP protocols as of Aug-2021
- For Data Integration scenarios such as Employee Data extraction, please use a technical user and use “credential type=OAuth2 SAML Bearer Assertion” in Cloud Integration.
- The keypair (certificate) generated in Cloud Integration binds the SAP SuccessFactors API/Technical user to the certificate. Hence, the SAML Assertion generated will have the specific user context and this governs the data/object access level in SAP SuccessFactors (API user permission in SAP SuccessFactors).

- Ensure to separate the API users in each Integration use case (Individual users per integration or per partner)
    - Give specific/minimum needed Role Based Permissions for the API user
    - Set Role Based Permission, so API user cannot be used to login through the UI of SAP SuccessFactors
    - Set an IP address restriction so it can only be used from the specific integration landscape which the API user was intended to connect. (KBA 2253200)
- Please refer to the SAP SuccessFactors Integration: Migrating SAP SuccessFactors API calls from Basic Authentication to oAuth2.0 for detailed configurations steps.

6.4 Data Extraction Detailed Approach

6.4.1 Delta Load vs. Full Load

Parameters	Full Load	Delta Load
Definition	Full load is also referred to as the Initial run of the integration between Source and Target systems, replicating the data for the first time for the defined scope of the Employee population. It can also be triggered on ad Hoc basis depending on the need of the customer organization **Note**: Full loads triggered daily is not recommended by SAP SuccessFactors	Delta load/Delta run for the integration is based on the concept of changed only records. It is commonly implemented in SAP SuccessFactors using the concept of "Last_Modified_On" queries **Note**: It is recommended by SAP SuccessFactors to use the delta loads, as this reduces the workload/resource consumption on source and target systems
When to use?	Full loads are recommended to be run only during the initial cut-over phase/First time migration scenarios **Note**: Apart from these scenarios a full load is recommended only when there is an emergency/un-avoidable situation. This is to safeguard server resources. E.g., There was a mass change in the organization affecting every employee record.	Delta load is used to keep the Source and Target systems in Sync, by replicating "Changed only" records from Source. This results in the optimal usage of the system resources (source/target/middleware) and is the recommended mode of mechanism.
CompoundEmployee Mode of operation	Full Transmission and Snapshot are the only modes which can be used in Full load scenarios, as rest other modes cannot work without last modified on filter	All modes of operation in CompoundEmployee can be used. **Note**: Last modified on date in CompoundEmployee cannot be greater than 3 months.
When to run?	Recommendation from SAP SuccessFactors is to schedule a full load during off business hours	The frequency of the delta runs can be set based on the customer needs. **Note**: We do not recommend to set the frequency below 5 minutes.

Table: Consideration parameters for Full load and Delta load

Note: Handling Large Data with Content Enricher and OData v2 adapter | SAP Blogs

6.4.2 Filter Conditions

To ensure optimum Process Execution and efficient resource management, it is best advised to filter the data being called from SuccessFactors. To align the relevant data, the query must have a filter “Where” condition. This where condition is responsible for restricting data being queried.

CompoundEmployee API has certain available filter that can be used to restrict data being queried. You can find out these filters at https://userapps.support.sap.com/sap/support/knowledge/en/2318180 . Some filters are mandatory for certain execution modes, while some filter conditions are only available for certain query modes.

ODATA APIs have more extensive field level filters and a lot more filterable fields are available. Various operators and nested AND/OR conditions can be used.

1. eq
2. in
3. ge/gt
4. le/lt
5. contains

While developing the process, it adds value to check the OData Data dictionary to check if a particular field allows being filtered upon. In case the field is not eligible for being filtered based on, then it is advised to query the data and filter it within payload at the next step.

For example: https://<SFSF-API-Domain>/odata/v2/EmpEmployment?$filter=personNav/personalInfoNav/lastModifiedDateTime ge datetimeoffset ‘xxx’ or personNav/lastModifiedDateTime ge datetimeoffset‘xxx’ or jobInfoNav/ lastModifiedDateTime ge datetimeoffset ‘xxx’.
Ensure that all the relevant entities navigations’ lastModifiedDateTime are added in the filter condition.

Extraction Type	Short Description	Target system needs entire history data of the Employee	Target system needs a part of history data of the Employee and not future dated records	Target system needs only as-of-dated records of the Employee
Full Load	Without Last Modified On filed in the filter condition (Usually the first run of the integration)	QueryMode= FullTransmission Mode, without any last modified on filters (Default mode of operations for CompoundEmployee API)	N/A	add effective_end_date= <current_date> in filter condition Effective end date filter
Delta Load	With Last Modified On filed in the filter condition (Subsequent runs after the full load)	QueryMode= FullTransmission Mode, with last modified on filters (Please do not go beyond 3 months in the last modified on date, as ComopoundEmployee will error out)	QueryMode=delta/periodDelta , with fromDate and toDate set to the period start and period end dates and lastModifiedOn filter	1.effective_end_date =<current_date> 2.QueryMode=delta/ periodDelta , with fromDate and toDate set to the current date and period end dates and lastModifiedOn filter **Note**: Link to the SAP Help Guide

Table: CompoundEmployee filters to be considered along with the execution parameters

Extraction Type	Short Description	Target system needs entire history data of the Employee	Target system needs a part of history data of the Employee and not future dated records	Target system needs only as-of-dated records of the Employee
Full Load	Without Last Modified On filed in the filter condition (Usually the first run of the integration)	use fromDate and toDate in the query without last modified on filter	use fromDate to query the history information and set the toDate=<current date>	OData APIs work by default on the as-of-dated information, if not filter condition is passed
Delta Load	With Last Modified On filed in the filter condition (Subsequent runs after the full load)	use fromDate and toDate in the query with last modified on filter	use fromDate to query the history information and set the toDate=<current date> along with last modified on filter	Use last modified on filter

Table: OData API filters to be considered along with the execution parameters

Filter multiple entities or Multiple Calls
In case the filter condition is too long or in case the expand query is not able to go to the level (expanding beyond 5 levels compromises the process efficiency), it is best advised to query data as part of different APIs, with different calls to individual API and then enriching the data as required.
It is always recommended to query filtered data rather than filter queried data.

Note: SAP SuccessFactors API server has a restriction on the character limit for OData GET URI 2576271 - OData GET request client URL size limitation: 2 KB

6.5 Transform Data

The next step in an Integration Process is Data transformation. SuccessFactors, delivers data in XML (and JSON in case of OData), the target system might require data in JSON, XML or CSV. To accommodate the target structure, mapping and various other steps are required.
SAP SuccessFactors EC has a nested structure, Employees' data is maintained under various EC portlets and they have corresponding APIs that are used to query data.

6.5.1 Navigations vs Enricher

In case of querying OData APIs, either the APIs can be expanded while exploring the Navigation and using the *$expand,* keyword. But parallelly this may cause repetitive data being queried from SuccessFactors EC, so in case of a large load of employees' data is queried from SuccessFactors via OData APIs, and same objects are being expanded on, objects which don't have varied values, and are being imported again and again. To avoid querying the same data form SuccessFactors repeatedly, to increase the efficiency of the flow, it is best recommended to use Enricher or Hash map based data caching and enriching mechanism.

In CompoundEmployee API or in OData API, where Navigation for a referenced object isn't feasible, Content Enricher of the object in question is the only possible solution. In such case, the only way is to use Content Enricher as a full load data (still query only the active data records using filters in Content enricher).

Note: Sampler blog content - Handling Large Data with Content Enricher and OData v2 adapter | SAP Blogs

6.6 Confirmation

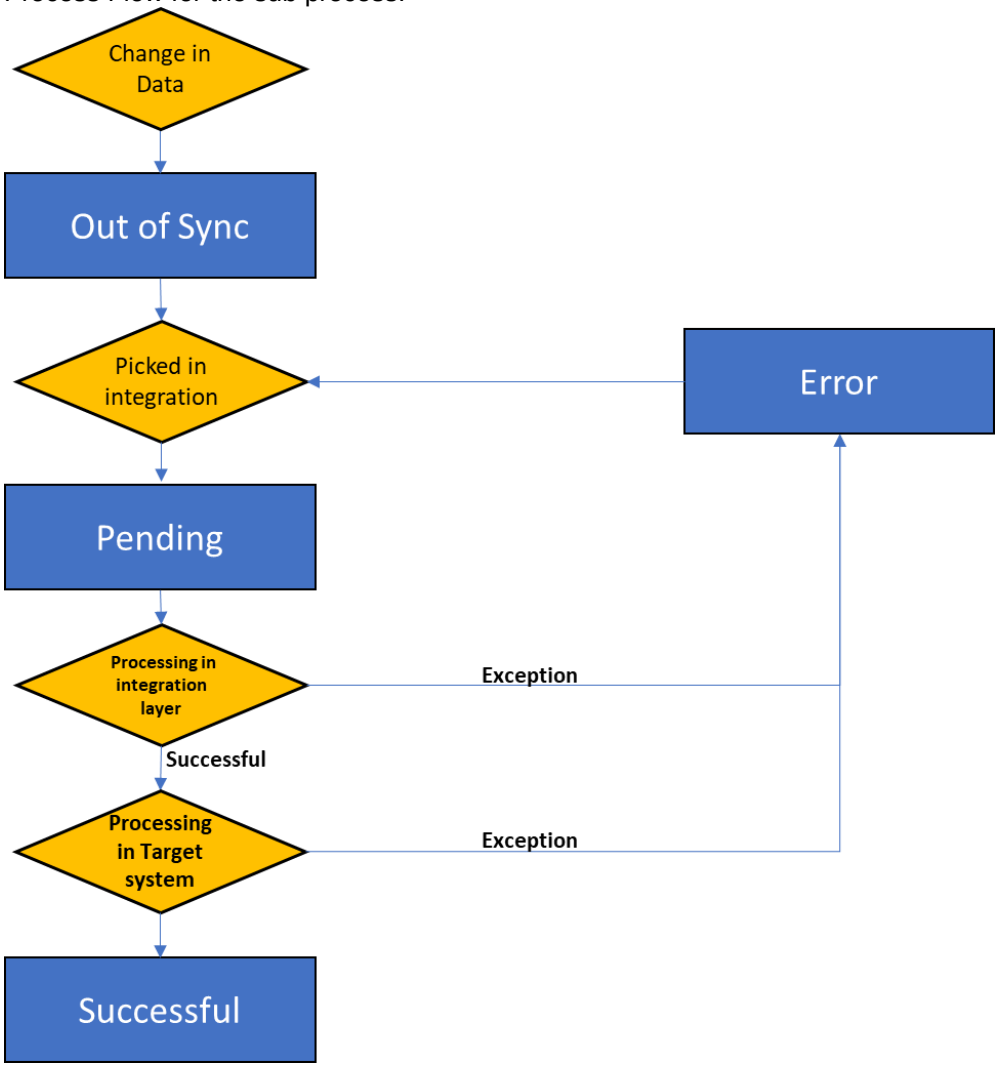
After the data has been sent to the Target system, there is a need for an update in SuccessFactors to mark the record status as successfully replicated. In case the data wasn't replicated due to error, it should be updated as such, in SuccessFactors as well. There are a variety of ways how the Confirmation flow can be configured. The pre-requisite for the Confirmation Procedure to be implemented is, the target system should be able to provide post processing logs. These post processing logs have the information about the outcome of the data upload in the process.

6.6.1 Offline Reports

In case the target system can't send a response back via integration then the data needs to be reported in terms of Excel sheet, with the date of replication of the employee record, status and error message if any. This report should be used for any gaps while the reconciliation. The report can be stored as audit logs and sent it out periodically to the HR or Payroll Administrator via email or downloadable format.

6.6.2 Custom MDF object

In case the target system, is capable of sending the response and output of date update to the integration layer, it is then advised to have a confirmation sub process as part of the end to end integration. Below is the Process Flow for the sub process:



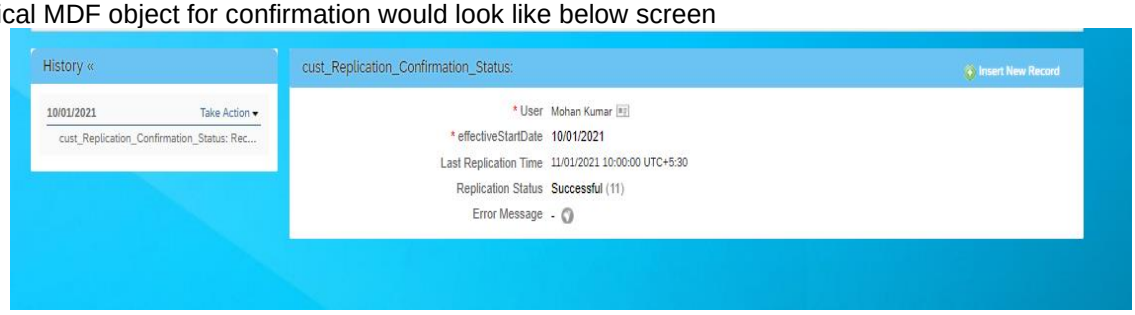
As in flow diagram, the change in data is needed to be picked up, and once picked up in the process, it should be marked as processing/pending. Once a record is marked pending/processing, It is either marked Successful or Error, dependent on post processing in integration and target system. In subsequent executions, it is important, to pickup all records that have been marked as error in the last call.

Ensure that all the entries in the custom MDF marked as Pending are executed and all the records are in either successful or failed status.

The Custom MDF object that keeps the records replication status, should be an effective dated record, that has the value of replication status, last successful replicated on and a flag to manually pick the record if required in the next run. The custom MDF object should be regularly updated via replication process, and in case the record status for any replication process is not successful, then the record needs to be picked up in a schedule call. The employee should be replicated as on the date of erroneous replication, ensuring that the data for the day is replicated as expected.

Note: Be cautious about the amount of records created on the custom MDF object confirmations, recommendation from SAP is to have one MDF record per employee and keep updating the same record,for each time the employee is picked up for replication to send the confirmation back.

A typical MDF object for confirmation would look like below screen



User: Employee for which the replication confirmation record is added
Start Date: The effective date for the employee record sent from SuccessFactors. In case the system only expects current effective data, or the schedule is not frequent (weekly or monthly), the MDF can be set to non effective dated, as the process won't be executing the same employee a lot of times
Last Replication Time: The last time the employee was successfully replicated
Replication Status: The status of the employee record for integration. It is best to keep the field as a picklist with 3 possible values: Pending, Successful, Error.
Error Message: In case the replication fails for the employee this field will keep a record of the error message, the error message can be customised according to the use case in the middleware.
Execute in Next Run(Optional):You can add this field to execute an employee in subsequent run. This flag can be configured to explicitly execute the employee's replication in the next run. A separate query needs to be added in the process to trigger the process to pick employees with execute in Next Run as true along with pending status.

6.6.3 Confirmation Process via Database/third party

In case the Architecture provides support for third party monitoring or Databases which can track the process execution, the confirmation process can be triggered to send data. The Database should have all the relevant columns and should be able to store data at the 3 stages of the replication, Start, Successful end, Error End. The process needs to update the Database at the three steps with each execution. These logs can be used for just audit purposes or if possible to trigger the process again, which however, will require a separate process/subprocess.

6.7 Monitoring

6.7.1 Monitoring using SAP SuccessFactors Execution Manager dashboard

Here are few pointers:

- Execution Manager (XM) is a Business User Friendly tool for HR Admin users.
- Execution Manager (XM) is a Business User Friendly tool for HR Admin users to track the execution step by step. Keep the payload size low (<= 64KB) and do not log several megabytes of data into the tool (Example: Do not log CE API responses in EMEventPayload).
- For each Cloud Integration process run, there should be one EMMonitoredProcess entry in the XM log.
- One EMMonitoredProcess can have multiple EMEvents.
- EMEvennts can have EMEventAttributes and EMEventPayload.
- 15 days is the retention period for the XM logs (default).
- Recommended limit from the SAP SuccessFactors product team for XM logs is less than or equal to 64KB (currently, there are no errors/warning messages but, in the future guardrails will be enforced).
- For internal tool like SAP SuccessFactors Integration center, default logging is done in XM
- XM provides capability to do logging from a 3rd party middleware tool or Cloud Integration via web services for custom built integrations.
- "ProcessType" is an enumerated value and accepts pre-defined value (SFOData.EMMonitoredProcess.ProcessType="INTEGRATION"). Custom built integrations using middleware tool can be monitored via "Middleware Integrations" tab in the UI.

- Integrations which follow, Point to Point Integration pattern through APIs should not use XM for monitoring purposes. (Because the dashboard UI is not designed for this use case, also there is no ProcessType for such scenario).

Following columns are shown in the table in the process view of XM Dashboard

Column Name	Description
Process Instance Name	Name of the process (Provided either by the user or generated by code) which is used to identify the job execution (Instantiation of the process run). This column will be populated as the value of processInstanceName API field (SFOData.EMMonitoredProcess).
Process Identifier	“processDefinitionId” (number or UUID) uniquely identifies that process definition within the system. Example: ID of the Integration Flow in Cloud Integration. The column 'Process Identifier' in the UI shows the values populated using the “processDefinitionId” API field (SFOData.EMMonitoredProcess).
Process Definition Name	Name of the Integration Process. This column is populated using “processDefinitionName“ API field (SFOData.EMMonitoredProcess).
Timestamp	This is the time at which the first event for a particular process is sent to XM
Process State	It shows the current state of each process.
Summary So far	There is an icon for each row under this column, clicking on which shows the 'Summary so Far' for that process.

START should be the first event checkpoint for any process. It can be followed by any number of ERROR, WARNING, INFO event checkpoints. For a long running job, SUMMARY_SO_FAR event checkpoints should be sent at intervals. Once the process is completed, the END event checkpoint should be sent. If the process fails due to any unknown error or exception, the FAILED event checkpoint should be sent.

SFOData.EMEvent.EventType	Short Description	Can be used in Integration?	Comments
Start	used to indicate the start of the Integration run/ Integration process	Yes	Mandatory event type, and this is always set in the Main Integration Process of the Cloud Integration process design. Should be sent only once per process run
INFO	Used to indicate the several stages of the process execution	No	SAP SuccessFactors by default does not allow this EventType to be displayed in the logs, support incident must be requested describing the business use case and SAP SuccessFactors product management will take the decision of enabling it for a customer instance.
SUMMARY_SO_FAR	Used to indicate the summary of the process execution	Yes	Use this EventType, if there is a long running process. This event should be sent at regular intervals to keep the end user informed on the progress.
ERROR/WARNING	Used to indicate the errors/warning state of the Integration process execution	Yes	Use the event type according to the functional design of the integration process to show

			Business errors/warnings to business users.
FAILED	Used to indicate failure of the process Integration. When the process is terminated due to technical failures.	Yes	This is an optional event, use this only when you as a developer specifically know the failure points based on the parameters or process run. Should be sent only once per process run. After this event type, please do not send “END”.
END	Used to indicate the process integration end state (Successful run only)	Yes	This should be used only when the process does not have any failures. Should be sent only once per process run.

Table: Different Event Types possible in SAP SFSF XM

Process State:

SFOData.EMMonitoredProcess.ProcessState values are calculated based on the EMEventType values. “INFO” and “SUMMARY_SO_FAR” events have no effect on the ProcessState.

State Diagram

Derived Process State as shown in UI	Meaning
Completed Successfully	The process has completed successfully without any errors or warnings. That is, no error or warning checkpoints had been sent for that process
Completed With Errors	The process has completed with few errors in it. It might or might not have any warnings it. A process with 'Completed with Errors' process state would include processes which completed with at least one error, and it can have warnings as well.
Completed With Warnings	The process has completed with warnings. Also, this means that the process doesn't have any errors in it. A process with process state 'Completed with Warnings' can only have warnings in it but no errors.
Failed	The process has failed. For example, if scheduler goes down while executing a job/process, the process is said to have 'Failed'
In Progress	The process has started, but hasn't completed yet
Unknown	If the START event checkpoint has not been sent, the state of the process is 'Unknown

Table: Possible values for "ProcessState" in SAP SFSF XM

Note:

- Integration Center jobs monitoring by default is done using the XM dashoard.
- Please refer to the following SAP Blogs for detailed understanding on the integration process development using the XM APIs in Cloud Integration.

CLOUD INTEGRATION and SAP SuccessFactors Monitoring EP01: The one with the Execution Manager Dashboard | SAP Blogs , CLOUD INTEGRATION and SAP SuccessFactors Monitoring EP02: The one where CLOUD INTEGRATION writes logs to Execution Manager | SAP Blogs

6.7.2 Exception Handling

Error Handling is the mechanism, via which the integration feeds the status of process execution. For an efficient and transparent integration, the best practices need to be followed. Exception processes need to be setup at different stages in the Integration.
Exception Handling is handled in the middleware level using Exception Subprocess. There can either be technical or functional exceptions.

6.7.2.1 Technical/Operational Exception Handling

Technical/Operational exceptions might include exceptions that are because of the system inadequacy or bugs. This can be due to either:

- Target/Source system down time
- Password/Authentication expired
- System Failure
- Read/Write Timeout etc.

In cases like system downtime, or Timeout issue, it is best advised to setup a retry mechanism, which ensures, that no data is lost and no execution is missed. Once the execution is competed after the retrial ensure that the same is updated in Execution Manager.
It makes only sense to retry in case the error might get resolved. While this can be the case for a timeout or system downtime it is not the case for a expired authentication or password. For those kind of errors a retry does not make sense, especially not if the retry is within a few minutes or even seconds.

6.7.2.2 Functional/Business Exception Handling

Functional or Business Exceptions are issue raised due to data entry or business scenarios that shouldn't have been possible.

Some examples of such scenarios:

- Missing data in mandatory fields
- Inactive value taken for some field
- Incorrect code assigned for some data
- Events/changes made to terminated/inactive employees effective before the hiring date etc.

All these errors give rise to exceptions on the record level that is part of payload. This may or may not cause the whole process to halt. It is best recommended to weed out the erroneous record, log it and update it in Execution Manager or employ the confirmation process to mark these employee records as failed within SuccessFactors.
If the error reason is assigned to the employee via the confirmation process then the admin can go into the employees' profile and correct the data.
It is advised to replicate the failed process again via an ad hoc run or correcting the data in SuccessFactors and waiting for the system to pick the updated records up and sending it to integration.

6.7.2.3 Exception Handling Mechanisms

Exception Handling can be done by

1. Sending an Email notification to the HR Admin
2. Execution Manager

6.7.2.3.1 Sending Email Notification

To notify the admins for the issue encountered, an Email notification can be configured using SMTP standard connection. The email can be provided in the HTML format according to the organisations standard and can have the logs as an attachment. Ensure that the data sent over via the Email, doesn't have employees' confidential data
An SMTP vendor is a prerequisite for this capability, since Cloud Integration doesn't provide built in email sending capabilities. Once acquired, the SMTP can be called for both types of exceptions: Operational or Business. Along with the errors, the process can also send the corresponding erroneous employee Ids along with the appropriate errors.to the system. For further information on how to trigger mail notification and the best practises to be followed please refer to https://launchpad.support.sap.com/#/notes/0002257038 for available servers and 2589987 - Restriction: Avoid Usage of SAP Internal Mail Service in Cloud Integration Mail Receiver Adapter - SAP ONE Support Launchpad for limitations/workarounds.

Note: Cloud Integration(CPI): Customizing Email using Mail Adapter | SAP Blogs

6.8 Event Based Integrations

Real Time events need to be considered based on the scenario basis.

Sample scenario: An employee (Employee B) is assigned a manager (Employee A) in the SAP SuccessFactors EC system. The newly assigned manager was promoted to the new position for the first time

Integration Scenario:
Source system is SAP SuccessFactors EC
Target system is a 3rd party time system

Business Need: Timesheet approver role needs to be assigned to the manager in real time.

API Used: CompoundEmployee API for replicating data from SAP SuccessFactors EC to 3rd party system

Logic used for assigning the timesheet role: Based on the direct_reports field value, if it is greater than "0" then the supervisor is assigned "Timesheet approver" role

Delta load: As the supervisor changed for the employee, last modified time will be changed and Employee will be picked up in the next replication.

Problem: direct_reports field is stored in the "USERS_SYS_INFO" table and thus employment information is not affected. Hence, the newly promoted manager data will not be having a last modified date with respect to EC tables. Hence, CompoundEmployee API will not be able to detect the changes and replicate the manager to target system (Reference - Support of Specific Attributes - SAP Help Portal)

Events relevant to this scenario: Individual Contributor to Manager / First Time Manager

Solution: Thus, event based approach becomes a need more than a solution in this scenario.

Dependency: HRIS Sync (Human Resource Information System (HRIS) Synchronization) should be configured in the SAP SuccessFactors system. This is a pre-requisite for the solution to work

Note:

- For more details on Intelligent Service Center events please refer to the Implementation Guide here
- For details on SAP Enterprise Messaging basics, please refer to the blog here
- Business Rules when attached to the Entity in Manage Business Configuration, should always be saved with "**OnPostSave**" as the trigger type.
- Sample Implementation – Click Here

6.9 Re-try option on failure

SAP SuccessFactors Integration Center:
Please refer to the SAP SuccessFactors Integration Center guide to enable this option " Execute with Exceptions" - Configuring REST based Outbound Integration

SAP SuccessFactors Intelligent services
Event failed to be consumed by target system/ Cloud Integration from SAP SuccessFactors Intelligent services center (the consumption entry point), then it will be redelivered again up to 9 times.

7 ASSUMPTIONS AND EXCLUSIONS

- This document is written with an assumption the reader has a basic understanding of the SAP SuccessFactors Data models (Data Models in SAP SuccessFactors , Data Models in Employee Central and in Employee Central Payroll)
- This document does not explain the nuances of how to Manage and implement the Employment lifecycle in Employee Central as it is already explained the SAP Help guide - Implementing and Managing the Employment Lifecycle (from Hiring to Termination) in Employee Central

8 REFERENCES

- **SAP SuccessFactors Integration: Migrating SAP SuccessFactors API calls from Basic Authentication to oAuth2.0**
- **Employee Central Core Hybrid: Handling Employee Identifiers**
- **SAP SuccessFactors Integration: Best Practices using SAP SuccessFactors APIs for Custom Integrations**
- **SAP SuccessFactors Integration: Integration Center and Cloud Integration**

www.sap.com/**contactsap**

Ybrant Partners

THE BEST RUN SAP®