



PUBLIC

Employee Central: Job Information Entity Configuration Considerations

Document Details

Name	Objective	Audience
Employee Central: Job Information Entity Configuration Considerations	The focus of the document is to clarify concepts related to Job Information entity to address high support ticket volumes in these areas and also to provide recommendations for its configuration.	SAP SuccessFactors Customers: IT and HR professionals. SAP SuccessFactors Implementation Partners: Consultants, solution architects and project managers

Change Log

Version	Date	Description
1.0	26.10.2021	Initial version
1.1	25.04.2022	Updated cross entity rule section with screenshots

Supported Releases

Product	Release - From	Release-Valid till
SAP SuccessFactors Employee Central	2111	

Contribution

Role	Name	Organization
Author / Owner	SAP SuccessFactors Product Management	SAP SE
Author	Rupsa Banerjee	SAP SDC
Author	Tanya Page	SAP UK
Author	Nathan Wilkinson	SAP UK

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.....	4
2	ABSTRACT	4
3	INTRODUCTION.....	4
4	BUSINESS REQUIREMENT.....	4
5	SOLUTION OVERVIEW AND CONCEPTS	5
6	DETAILED SOLUTION	6
6.1	EVENT AND EVENT REASONS.....	6
6.1.1	<i>Configuring Event Reason Derivation</i>	<i>7</i>
6.1.2	<i>Custom Events.....</i>	<i>10</i>
6.1.3	<i>Country Specific Event Reasons.....</i>	<i>11</i>
6.1.4	<i>Recommendations for configuring Event Reasons Derivation rules</i>	<i>12</i>
6.1.5	<i>Behavior with Import</i>	<i>13</i>
6.1.6	<i>Troubleshooting Tips for Event Reason Derivation</i>	<i>14</i>
6.2	CALCULATION BEHIND FTE	14
6.2.1	<i>Rule to calculate FTE</i>	<i>15</i>
6.2.2	<i>FTE as a calculation basis.....</i>	<i>19</i>
6.2.3	<i>FTE Rule Functions</i>	<i>20</i>
6.3	CROSS ENTITY RULES	22
6.4	REFRESH COMPENSATION REQUIRED FEATURE	27
6.5	FORWARD PROPAGATION	29
6.6	JOB INFORMATION HISTORY.....	30
6.7	TIMEIN CALCULATIONS	30
6.8	SYNCHRONIZATION OF DATA FROM POSITION OBJECT WHEN USING TIME-OFF.....	31
6.9	ORDER OF THE FIELDS ON JOB INFORMATION.....	32
6.10	JOB INFO HISTORY EDIT VS TAKE ACTION.....	34
6.11	RESTRICTION OF USING 'ONVIEW' RULES 'ONCHANGE'	35
6.12	TERMINATION	35
6.13	POSITION MANAGEMENT "RIGHT TO RETURN" FEATURE AND IMPACTS ON JOB INFORMATION	35
7	REFERENCES.....	37

1 TERMINOLOGY

The following table explains some abbreviations used in this document.

Abbreviation	Description
EC	Employee Central
ERP	SAP Enterprise Resource Planning often referred in the document pertains to SAP HCM on premise system
MDF	Meta Data Framework
RBP	Role Based Permissions
UI	User Interface
MBCUI	Manage Business Configuration UI
SWH	Standard Weekly Hours

2 ABSTRACT

Job information entity in SuccessFactors Employee Central is an important entity which is used to record Job related Information of an employee during the course of employment. It is important to configure this entity and fields within it correctly so that the Job Information stored for each employee is correct, current, and also in sync with other areas and entities that get influenced or influence this entity.

3 INTRODUCTION

Job information entity in SuccessFactors Employee Central is an important entity which is used to record Job related Information of an employee during the course of employment.

It is an effective dated entity and contains information about the Organization Structure, Position, Job and Pay structure, Supervisor, Employment Contract details etc. applicable to each employee.

Job Information entity is related to other entities within Employee Central like Time Management, Position Management, Compensation Information entity etc. and influences them and is influenced by them. This helps to ensure that the data is consistent across different related entities. Hence, it is important to have clear understanding of job information entity.

This document describes certain aspects of Job Information entity which require better understanding and configuration guidance to help consultants during implementation and support of these topics.

4 BUSINESS REQUIREMENT

Job Information is one of the most important entities within Employee Central. Analysis of support tickets in Job Information area has revealed that the ecosystem has a need for better understanding of this entity.


Problem areas identified within Job Information entity are–

- Job Info Event Reasons
- Understanding calculation behind FTE
- Cross Entity Rules
- Refresh Compensation Required Feature
- Understanding of Position sync rules during hiring and Job Information edit

- On view rules
- Order of fields within the entity
- Right to return and impact on Job Information

5 SOLUTION OVERVIEW AND CONCEPTS

Job Information Entity is used to record data of an employee with respect to organization structure, position, supervisor, job and pay structure, contract, time information etc., as shown below.

Employee Status	Job Information	Time Information
Employee Status	 Charles Braun Supervisor	Holiday Calendar Work Schedule
Event	Job Classification	Time Profile
Event	Job Title	
Event Reason	Pay Grade	Time Recording Profile
Position Information	Regular/Temporary	Time Recording Admissibility
Incumbent of Parent Position	Standard Weekly Hours	Time Recording Variant
Position	Working Days Per Week	Default Overtime Compensation Variant
Position Entry Date	FTE	Dynamic Break Configuration
Time In Position	Is Fulltime Employee	Clock In Clock Out Group
Organization Information	Notes	
Company	Employee Class	EEO Information
Business Unit	FLSA Status	EEO Category 1
Division	Employment Type	EEO Category 4
Department	Country	EEO Category 5
Location	Pay Scale Type	EEO Category 6
Timezone	Pay Scale Area	
Cost Center	Pay Scale Group	
	Pay Scale Level	
	Contract End Date	
	Competition Clause	
	Sideline Job Allowed	
	Probationary Period End Date	
	EEO Job Group	
	Contract Type	

Fields in Job Information Entity are part of Succession Data Model and Country Specific Succession Data Model. Standard data model files are available in SAP Help portal and are uploaded in provisioning when an EC implementation starts.

Job information Features:

- **Effective Dated:** Job Information records are effective dated and contain date wise (chronological) changes in Job Information of an employee. This builds up job History of an employee which usually remains unchanged except when a back dated change is made. Job Information entity also supports future dated changes along with multiple changes per day.
- **Edit, Import and Forward Propagation:** Job Information data changes can be made either manually or through imports. These changes can get propagated to later records in certain conditions and hence it is important to understand how this works.

- **Association:** Foundation objects and Generic Objects can be linked with each other through relationships - one to one or one to many. This feature helps to reduce data errors. Associations should always be configured to go from the lower-level object (the child object) to the higher-level object (the parent object). In other words, the association should be placed on the object you expect to be filtered based on the parent value selected. For example - Location (Child) to Legal Entity (Parent) requires that the association be placed on the Location object. Following this, in the UI, after selecting the Legal Entity, a filtered list of Locations will be available, based on the selection made in Legal Entity. Similarly, if for e.g., two job codes belong to one location then the association must be placed on the child object i.e., job codes, such that when you select the parent (Location) you get child (Job Codes) as filter values based on parent selected.

Note that Many-To-One Association is not supported.

The screenshot shows the 'Associations' configuration page. It includes a table with the following columns: Name, Multiplicity, Destination Object, and Type. The 'Name' column contains the text 'cust_'. The 'Multiplicity' column has a dropdown menu open, showing 'One To One' (selected) and 'One To Many'. The 'Destination Object' column has a search input with the text 'Absence Counting Method'. The 'Type' column has a dropdown menu with 'Composite'. A 'Details' link is visible. Below the table is a 'Searchable Fields' section.

- **Event and Event Reasons:** Events are pre-delivered by SAP SuccessFactors. You can't create new events or change existing ones; you can only change their labels. Events can have one or multiple Event Reasons associated with them. Event reasons are defined by a customer. It is recommended to use Business Rules for Event Reason derivation.
- **Support Business Rules:** Business rules can be used in Job information entity to:
 - Default field values
 - Propagate values, e.g., when Position is selected, Business Unit, Division, Department etc. field values can be defaulted by configuring Position to Job Sync rule.
 - Set field characteristics e.g., mandatory/non-mandatory, visible/not visible etc.
 - Raise warning/Information/error message to control input values for fields
 - Calculate certain fields based on user input for linked fields e.g., FTE etc.
 - Trigger cross entity changes, workflows and Intelligent Services Centre Events

Incorrect configuration or placement of business rules within the HRIS element can lead to rules not working correctly. There can be various reasons that prevent desired execution of a rule and in such cases the **Business Rule Execution Log** admin tool is very helpful for troubleshooting.

- **Pre-calculated field on Job information:** Some Job information fields are auto calculated like Time in company. Logic behind their calculation and impact of Centralized services needs to be understood for these fields.

This document aims to clarify product behavior with respect to the above-mentioned features with the help of sample scenarios wherever possible.

6 DETAILED SOLUTION

6.1 Event and Event Reasons

Events are occurrences that span the various stages of an employee's lifecycle from hire to rehire. Technically, events are defined in a picklist. Events are pre-delivered by SAP SuccessFactors.

Event reasons are foundation objects. They are used to define more specifically the reason why an event has occurred. For example, the event 'Hire' could have event reasons 'New Hire' or 'Acquired Employee'.

6.1.1 Configuring Event Reason Derivation

When a record is added in a Job information of an employee, an event reason has to be selected for every event that has occurred. Event reason derivation rules are configured to derive reasons based on the changes entered in employee data. Multiple event reasons can be associated with an event. Event reasons are mandatory in the system.

Every event reason must have an event and appropriate employee status assigned to it except when it is a data change event, in which case the employee status remains blank. Getting employee status wrong can lead to issues.

Event Reason: New Hire (HIRNEW)
Effective as of 01/01/1970
Blue indicates that the item changed on this date
Event Reason ID HIRNEW
Event Reason Name New Hire ⓘ
Description New Hire ⓘ
Status Active
Event Hire
Employee Status Active
Follow-Up Activity in Position No Selection
Payroll Event
Display in Internal Job History Portlet No

Users can create event reasons by navigating to “Manage Organization, Pay and Job Structures” screen or by mass uploading data using a CSV file in the Admin Center “Import Foundation Data”.

Visibility of event reasons for a role can be controlled through RBP. Once permissions are provided for event reasons, they become available to a role.

Specify what permissions users in this role should have. ⓘ ★= Access period can be defined at the granting rule level.

Payroll Permissions	Event Reasons	<input type="checkbox"/> View	<input type="checkbox"/> Edit
Payroll Control Center	Acquired employee (HIRACORE)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Recruiting Permissions	Across-The-Board (PAYATB)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Reports Permission	Add Global Assignment (ADD_GA)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Analytics permissions	Additional Job (CONCRJOB)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Employee Data	Adjustment to Minimum (PAYADJ)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
General User Permission	Annual Salary Review (ANSALREV)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SAP System Configuration	Assignment Completion (ASSGCOMP)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Payroll Integration Permission	Away on Global Assignment (AGA)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Continuous Performance Management	Back from Global Assignment (BGA)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Benefits Adjustment (PAYBEN)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Benefits Base Update (PAYBENRT)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Bonus Buyout (PAYBBO)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Business Unit Change (DTACHBU)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	California Family Rights Act (LOACAERA)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Event Reason field is hard coded in jobInfo HRIS element but is not available by default in succession data model. The field should be defined either through Business Configuration UI or through xml upload so that translations can be managed and onChange Business Rules can be triggered. If the field is not configured, only on save rule can be triggered with hardcoded field.

event-reason

Event Reason ⓘ

Job Information (jobInfo_jobFl...

Yes

Det

There are two ways to define events and event reasons:

- Activate the automated event and event reason derivation which means business rules can be used to derive the event reasons automatically. This is done in provisioning:

☒ Enable Business Rules for Workflow Derivation (Do not activate together with "Enable Youcalc rules engine for HRIS.") [Not Ready for Sales/Production] — requires "Employee Central V2 (i.e., Event Reason Derivation)", "Enable Generic Objects", "Effective Dated Data Platform", "Employee Profile data audit" and "Enable the Attachment Manager"

- If event reason derivation is not activated in provisioning, the users, with proper authorization, can select the event-reason manually from the UI every time and make change to the employee data that is linked to an event. However, this is **time-consuming and more error prone** as it depends on user's understanding of event reasons.

Automated event reason derivation ensures 100% data consistency as there is no way to assign a wrong event or event reason to a specific transaction. Event reason derivation helps to simplify the process for users. Thus, **it is recommended to implement event reason derivation using business rules.**

A new rule scenario 'Event Reason Derivation' was introduced in Q4 2019 release under Employee Central Core to create business rule to derive event reasons.

Employee Central Core

☐ Generate Assignment ID External

Use this scenario to create the rule that generates the value for Assignment ID External based on MDF Sequence objects. Create a single rule only based on this scenario. Please follow the detailed documentation in the SAP Help Portal under the search term 'Assignment ID External'.

☐ Generate Employee ID For Hire/Rehire

You can use this scenario to generate an Employee ID from the Metadata Framework Sequence (MDF) and assign it to the User ID field of the Employee Information object during the Hire/Rehire with new employment process. You must first register the rule in the Matching User Configuration object settings.

☐ Rules For Hire/Rehire

You can use this scenario to configure rules on the Employee Information object during the Hire/Rehire with new employment process.

☒ Event Reason Derivation

You can use this scenario to derive the event reason automatically for Job and Compensation Information entities.


These rules are for Job Information and Compensation Information base objects only.

Event Reason Derivation


You can use this scenario to derive the event reason automatically for Job and Compensation Information entities.

Rule Name*

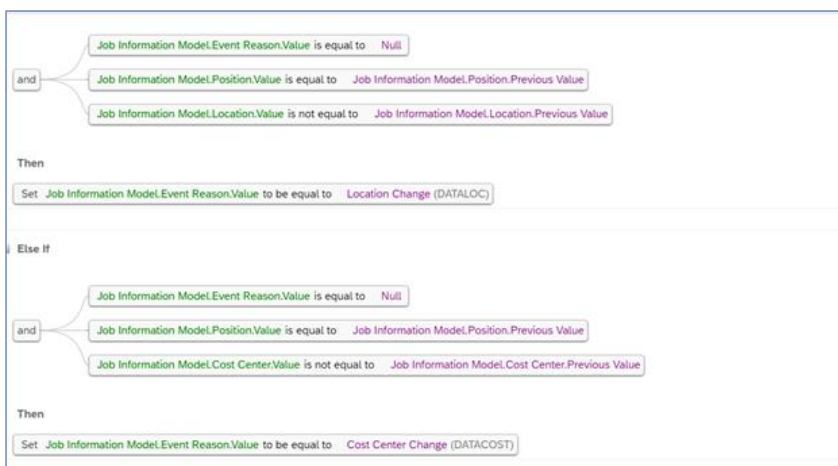
Rule ID*

Start Date* 

Description

Base Object Type* 

The rules 'set' event reasons based on 'If' conditions specified in the rule. A business rule can have many blocks to define conditions and set event reasons. A sample rule to set event reason as "Location Change" / "Cost Center Change" is shown below. When a user edits location/cost center in an employee's record, based on below rule, system automatically identifies the associated event reason and updates the same when record is saved.



Order of event reasons in the rule is important, e.g., if both cost center and location are changing at the same time, system will assign the location change event reason as per above rule because system doesn't evaluate further conditions if one true condition is executed.


Null condition is included in the rule to ensure that the event reason is not overwritten when a user edits a historical record.

Note: There are certain scenarios in which event derivation is not applicable, i.e., hiring, termination, add global assignment, leave of absence, add concurrent employment. One needs to select the reasons manually in these cases. Event Reason Derivation only gets executed for changes in Job Information and compensation information records done through MSS UI and Employee Data Imports (and APIs which runs through the imports coding internally). Not on History UIs.

Depending on the event reason, employee status is updated, if necessary.

Note that event reason is not visible when Event Reason Derivation is **enabled**, and the user is editing the record via 'Job Info Edit' (via pencil icon/MSS).

The event and event reason fields are visible from the history screen as shown below.

Change History	Job Information Changes
Jan 01, 2017	Last updated by  on Monday, September 25, 2017 5:38:08 AM EDT Effective as of: Jan 01, 2017 Employee Status Employee Status Active Event Event Pay Rate Change Event Reason Other Adjustment (PAYOTH)
Other Adjustment	
Sep 01, 2005	
Data Change	

It is recommended to maintain Job and Compensation Information via Actions Menu->" Change Job and Compensation Information" and not via the Job Information Entity itself.

Event Reason Business Rule needs to be assigned as **onSave** rule in the **Manage Business Configuration UI** under Job information or Compensation Information HRIS element depending on the base object type selected. This is saved in the Succession Data Model (SDM) for the corresponding entity.

Trigger Rules			
Base Object	* Event Type	* Rules	Enabled
Job Information	onSave	WF Job Import (WFJobImport)	Yes
Job Information Model	onSave	WF Promotion with and without ...	Yes
Job Information Model	onSave	WF Transfer with and without p...	Yes
Job Information Model	onSave	WF Other Job Info Changes (EC_...	Yes

6.1.2 Custom Events

There is a list of pre-delivered 'Events'. These are stored as picklist. The system will not stop you from creating any new custom events. However, it's imperative that you do not create any new custom event as these will cause issues during workflow and event reason derivation.

Standard Events

- Additional Job
- Assignment
- Assignment Completion
- Job Change
- Completion of Probation
- Data Change
- Demotion
- Furlough
- Hire
- Leave of Absence
- Job Reclassification
- Pay Rate Change
- Position Change
- Probation
- Promotion
- Rehire
- Return from Disability
- Return to Work

- Suspension

In the list of pre-delivered events, you should not repurpose some of the events. For example, Return to Work is only valid for Leave of Absence and Assignment is valid for Benefits only.

6.1.3 Country Specific Event Reasons

Customers having presence in multiple countries/regions often have event reasons that are specific for a country/region. As an employee is always assigned to one legal entity, and thus to one specific country/region, relevant event reasons can be used for that employee (even in history / Add new employee screens).

To configure country specific event reasons, the following steps can be followed:

- From the “Configure Object Definitions” page, choose Object Definition “Country/Region”.
- Select Take Action-> Make Correction.
- Under Associations, in the Destination Object field, select the wrapper generic object that you want to use. Under Multiplicity column maintain "One to Many".
- Save the object.

Name	Multiplicity	Destination Object	Type	(13) More
cust_FOWEventReason	One To Many	Event Reason Wrapper	Composite	Details

For each country/region, assign the event reason relevant for that country/region.

- Navigate to “Manage Data”.
- In the Search field, select the Country/Region object.
- In the field next to Country/Region, select the corresponding country/region, for example, United States (USA).
- Select Take Action->Make Correction.
- Add Event Reasons relevant to the country/region and save.

Country/Region: United States (USA)

Country/Region Code (3 char)
USA

Country/Region
United States

Status
Active

Effective as of
01/01/2018
01/01/2015

Country/Region Code (2 char)
US

Country/Region Code (numeric-3)
840

Currency
USD (USD)

Data Retention Enabled
Yes

FOWEventReason
Event Reason
Acquired employee (HIRACORE)
Add Global Assignment (ADD_GA)
Additional Job (CONCRJOB)
Correction-Pay Rate (PAYCHANGE)
Change Bonus Amount (PAYBON)
Comp Frequency Change (PAYFRO)
Data Change (DATACHG)
Cost Center Change (DATACOST)

Country specific event reasons may be required by:

- Customers operating in multiple countries/regions and having country/region-specific event reasons
- Customers having several legal entities in different countries having different event reasons

6.1.4 Recommendations for configuring Event Reasons Derivation rules

Following are some recommendations for configuring Event Reasons Derivation rules:

- Each business rule consumes processing time when saving. This means that the lesser the onSave rules the better will be the system performance. Hence, best practice of "Grouping the event reason derivation logics in a single rule using Else IF" should be followed. Event Reason derivation logic typically spans a series of logical decisions / conditions which are recommended to be combined into a single rule for better system performance and hence overall user experience.

As a leading practice, event reasons should be designed to be logically simple and fewer in numbers. Thus, they should be more generic instead of being very granular and specific.

In cases where the requirements are more complex and hence, larger number of event reasons are needed, or where changes are quite frequent and hence maintainability becomes important, it may be justifiable to split the logic across multiple rules to support easier maintenance. Although this may come at some cost to performance. Hence customers and partners should consider performance impacts and maintainability while deciding on required event reasons.

Please refer to IDP [Employee Central: Optimizing Business Rules for Specific Scenarios](#) for further details.

- If event reason derivation is enabled, system tries to get the event reasons first on the Job Information entity. If system can't get any event reason on Job Information object, then it checks the Compensation Information entity. System triggers an error if it is neither set on the Job Information nor on the Compensation Information entity. Therefore, it implies that it is important to define event reasons for all possible scenarios in the derivation rule(s) and, to set a condition in the end of the rule for 'Catch All' situation.

If there is no 'Catch All' condition in the rule, then the error message would populate "Unable to determine the event reason for the proposed changes". This message is not a defect but a configuration issue which means that based on the clauses you configured for the event derivation rule, it was not possible to set the event reason.

Below is an example of a Catch All business rule block which should be included in the Event Reason Derivation Rule. It should have below shown IF condition and should not be configured for "always true" scenario.

The screenshot shows a configuration window for an 'Else If' rule block. It contains two main sections: 'If' and 'Then'. The 'If' section has a condition: 'Job Information Model.Event Reason.Value is equal to Null'. The 'Then' section has an action: 'Set Job Information Model.Event Reason.Value to be equal to Data Change (DATACHNG)'. Both sections have a small edit icon to the right of the text.

It is important to check if Event Reason is NULL before setting the Catch All Event Reason. This means that if the system is not able to determine event reason based on If conditions specified above this block, it would set 'Data change' as the event reason. Otherwise, the system can overwrite Hire, Termination, Rehire events or Events specified manually.

- Business rule blocks are processed in the sequential order in which they are written. Hence it is important to set conditions for the more specific scenario in the beginning of the rule and then set conditions for more general scenarios. This way system validates most specific scenarios first. If it finds a condition which is true, then it sets associated event reason else it continues to more generic event reasons. The assumption is that if both a specific and a general scenario is "True", the preference is to use the Event Reason assigned to the more specific scenario.

For instance: If field A and B and C are updated then use the event reason X else if field A and B are updated then use the event reason Y else use catch all event reason. If only A and B are checked in the first rule block and A, B and C in the next rule block, the rule will never work for condition C as first block will always be true and system will set the event reason based on the first true condition it encounters. This is because, if a general scenario is checked first, then the more specific scenarios might never get checked. Please note that in the example provided above, the rule starts with validating bigger set of conditions and then more general conditions.

Basic Information

Start Date: 01/01/1900
Rule Type: onSave (onSave)
Description:

Parameters

Name	Object
Context	System Context
Job Information Model	Job Information Model

Variables

If

and

- Job Information Model.Legal Entity.Value is not equal to Job Information Model.Legal Entity.Previous Value
- Job Information Model.Business Unit.Value is not equal to Job Information Model.Business Unit.Previous Value
- Job Information Model.Division.Value is not equal to Job Information Model.Division.Previous Value
- Job Information Model.Department.Value is not equal to Job Information Model.Department.Previous Value
- Job Information Model.Event Reason.Value is equal to Null

Then

Set Job Information Model.Event Reason.Value to be equal to Event Reason X (X)

Else If

and

- Job Information Model.Legal Entity.Value is not equal to Job Information Model.Legal Entity.Previous Value
- Job Information Model.Business Unit.Value is not equal to Job Information Model.Business Unit.Previous Value
- Job Information Model.Division.Value is not equal to Job Information Model.Division.Previous Value
- Job Information Model.Event Reason.Value is equal to Null

Then

Set Job Information Model.Event Reason.Value to be equal to Event Reason Y (Y)

6.1.5 Behavior with Import

To have the event reason derived using business rules during imports, you must enable the “Enable Business Rules for selected entities” permission for imports. We also recommend activating “Enable the Execution of Rules Against NO_OVERWRITE” setting.

Specify what permissions users in this role should have. Ⓢ Access period can be defined at the granting rule level.

Manage Foundation Objects Types

Enable business rules for selected entities Ⓢ

Ⓢ All Ⓢ Others

Addresses

Biographical Information

Compensation Information

Email Information

Employment Details

employmentTerminationInfo

Global Information

Job Information

National ID Information

One-Time Payments

Personal Information

Done Cancel

Specify what permissions users in this role should have. Ⓢ Access period can be defined at the granting rule level.

Manage Apprentices

Manage Time

Manage Time Off

Manage Advances

Manage Benefits

Manage Document Generation

Manage Mass Changes

Employee Central API

Employee Central Import Settings

Manage Income Tax Declarations

Enable execution of rules against NO_OVERWRITE Ⓢ

Ⓢ All Ⓢ Others

Addresses

Biographical Information

Compensation Information

Email Information

Employment Details

Global Information

Job Information

Job Relationships

National ID Information

One-Time Payments

Personal Information

Done Cancel

If the event reason value is provided in the import template, then this value provided in the template is considered rather than the event reason derivation by onSave business rules. The data is validated by the system to ensure that an event reason is given. Since it is a required field, it cannot be empty.

If an event reason column is available in the template but is empty, then the event reason is derived using the onSave business rules.

For Insert operation, event reason has to be either provided in the template or be derived via event reason derivation. For Update/Modify operation, If the event reason is removed from the import template or if the `&&NO_OVERWRITE&&` value is provided in the template for the Event Reason field, then the event reason used before the update will be used.

6.1.6 Troubleshooting Tips for Event Reason Derivation

If the Event Reason Derivation using business rules is not working correctly for you, please check points mentioned below to eliminate some of the common issues.

- Users are permissioned for Event Reasons associated with changes they are making
- Event Reason ID is no longer than 10 characters to avoid loss of characters during replication.
- Event Reason effective date may be in the future and hence it may not be available for the date Job Information change is being done
- Rule may not be correct. Use 'Business Rule Execution Log' to check if the rule is returning the correct event reason
- Sequence of onSave rules in Job Information/Compensation Information HRIS element may need repositioning. Other rules placed before or after the event reason derivation rule may be impacting the field values thereby impacting the derivation of event reason.

6.2 Calculation behind FTE

FTE or Full Time Equivalent is an important field within EC. It is an employee's scheduled hours divided by the employer's hours for a full-time workweek. When an employer has a 40-hour workweek, employees who are scheduled to work 40 hours per week are 1.0 FTEs. Employees scheduled to work 20 hours per week are 0.5 FTEs.

SAP recommends having FTE as a read-only and calculated field. If the standard-hours field is enabled in the configuration, system will always calculate the FTE based on Job Information Standard Hours vs Object Standard Hours. As per EC implementation guide, system derives the standard hours value used to calculate FTE from the following (in this order):

- `jobInfo.standard-hours / jobInfo.Position.standardHours` (Only if Employee Central Position Management is enabled and the standardHours field is visible in the Position object.)
- `jobInfo.standard-hours / jobInfo.job-code.standardHours`
- `jobInfo.standard-hours / jobInfo.location.standardHours`
- `jobInfo.standard-hours / jobInfo.company.standardHours`

This means, that if the job code has a standard hours value, this wins over location, which wins over legal entity. If the system finds no standard hours value, then it will move on to the next object and so on, until it finds a value, e.g., if Employee Central Position Management is not enabled and the standard Hours field is not visible in the Position object, then FTE gets calculated based on the next object (job code, location, legal entity in this order) which has standard hours field. If we chose location to store standard hours, we should ideally remove the Standard Weekly Hours from position and job so as not to have data inconsistencies.

Job Classification: HR Professional (50029122)

- * Start Date 01/01/1990
- * Job Code 50029122
- Job Title HR Professional
- Description HR Professionals
- * Status Active
- Workers'Comp Code
- Parent Job Classification
- Job Level IC - Non-Manager
- Standard Weekly Hours 40

Legal Entity: BestRun (10000)

- * Start Date 01/01/1900
- * Legal Entity ID 10000
- Legal Entity Name BestRun
- Description BestRun for Sandbox
- * Status Active
- Default Pay Group USA Semi-Monthly (US)
- Default Location Boston (10000-0001)
- * Standard Weekly Hours 40
- * Currency USD (USD)
- * Country United States (USA)

Location: Amsterdam (2500-0001)

Effective as of 01/01/1900

Blue indicates that the item changed on this date

- Code 2500-0001
- Name Amsterdam
- Description Amsterdam
- Status Active
- Location Group Europe (Europe)
- Timezone Europe/Amsterdam (GMT+02:00)
- Standard Weekly Hours 40
- Geo Zone Nordics (Nordics)

Object Standard Hours (Legal Entity or Location or Job Classification)

6.2.1 Rule to calculate FTE

Here is a sample rule for calculating FTE based on Standard Hours. The rule is triggered when the standard hour field in Job Information changes.

Following Rule Scenario should be used while creating a business rule to calculate FTE.

Calculate Full-Time Equivalent

You can use this scenario to calculate full-time equivalent for a user using the Job Information Model base object.

IF condition will be “Always True”.

Scenario: Calculate Full-Time Equivalent

Basic Information

Start Date

01/01/1900

Description

Parameters

Name	Object
Context	System Context
Job Information Model	Job Information Model

Variables

If

This rule is always true.
To add an expression please untick the Always True tick box.

THEN condition -

Then

Set Job Information.FTE to be equal to

Calculate FTE based on Standard Hours()

(Optional) Weekly Standard Hours:

Job Information.Standard Weekly Hours

(Optional) Standard Hours of Position:

Job Information.Position.Standard Weekly Hours

(Optional) Standard Hours of Job Code:

Job Information.Job Classification.Standard Weekly Hours

(Optional) Standard Hours of Location:

Job Information.Location.Standard Weekly Hours

(Optional) Standard Hours of Legal Entity:

Job Information.Company.Standard Weekly Hours

The rule checks in the following manner:

If Standard Hours on Position is defined and not zero, then $FTE = \text{Job Info Standard Weekly Hours} / \text{Standard Weekly Hours of Position}$.

If Standard Hours on Position is not defined or zero, then $FTE = \text{Job Info Standard Weekly Hours} / \text{Standard Weekly Hours of Job Code}$.

If Standard Hours on Position and Standard Hours on Job Code are both not defined or zero, then $FTE = \text{Job info Standard Weekly Hours} / \text{Standard Weekly Hours of Location}$.

If Standard Hours on Position, Standard Hours on Job Code and Standard Hours on Location are not defined or zero, then $FTE = \text{Job Info Standard Weekly Hours} / \text{Standard Hours of Legal Entity}$.

If Standard Hours of Position, Standard Hours of Job Code, Standard Hours of Location and Standard Hours of Legal Entity are not defined or zero, then $FTE = 0$. Hence, it is important to define standard hours on at least one of these objects.

Calculation example:- If Standard Weekly Hours defined on the Legal Entity is "40" and Standard Hours defined in Job Information entity is "20". Then FTE will be calculated as $20/40 = 0.50$. So, the FTE is 0.50.

If you do not wish to use the auto-calculation based on standard hours, you can set FTE to any value manually or via rule (except 0 or null, this would trigger the system calculation again).

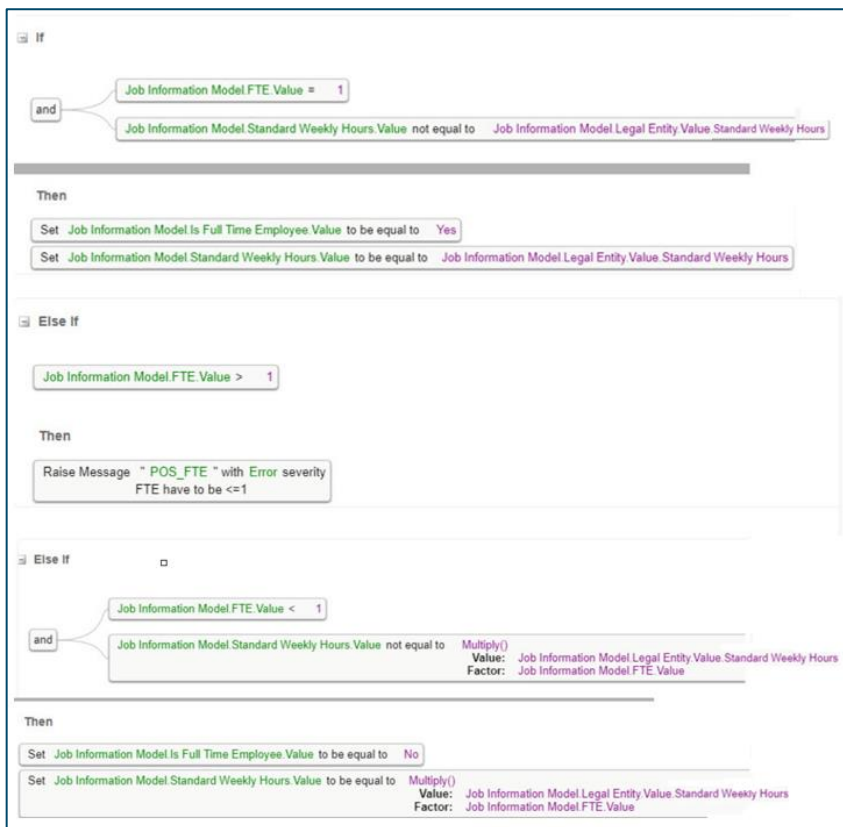
It is necessary for a user to have at least view permission for FTE field. If the FTE field is not visible to the logged-in user for UI transactions, the application will reset the FTE value to null so that it is recalculated.

Let us consider following scenarios.

- Standard Hours are not maintained for any of the four objects – position/job code/location/legal entity. Let's assume, HR Admin role has the permission to maintain FTE while Manager role doesn't have permission to view FTE. HR Admin sets FTE to 1 for an employee. When employee's manager edits Job information record of the employee, system will try to recalculate FTE. As per standard configuration, FTE will be 0 since system tries to recalculate FTE based on Standard Weekly Hours which is not available in any of the four objects. System will set FTE to 0 instead of 1.
- There can be a scenario where a customer's requirement is to calculate Standard Weekly Hours based on FTE. As per standard solution, the UI will always calculate FTE based on Standard Weekly Hours which cannot be disabled if the FTE field is enabled. In such a scenario, Standard Hours will not get calculated after entering FTE or the UI may appear to be loading continuously - it may appear like it is stuck in a loop.

Below steps can be followed to build a custom solution: -

- Standard Hours should be configured in at least one object such as Legal Entity.
- Create a Business rule as shown below using Job Information or Job Information Model as the base object.



- Rule Trigger - Set the rule as an onChange rule, on the "FTE" field in jobInfo HRIS element.

To have similar rule during Hire / Rehire Scenario, create the rule exactly as mentioned above but with the Base Object as "Employee Information" or "Employee Information Model".

- Customers with Time-Off enabled, may have a requirement to set Standard Weekly Hours based on Work Schedule assigned to an employee. In such a scenario, following steps can be followed.

- Create Business Rule as shown below:

CV_Propagate Standard Working Hours and Working days per week in Work Schedule (cv_PropagateDataToWorkSchedule)

Scenario: Basic [Change Scenario](#)

Basic Information		Parameters
Start Date	01/01/1900	Name
Rule Type	onChange (onChange)	Context
Description	Defaults Standard Working Hours and Working days per week.	Job Information Model
Show More		Job Information Model

Variables

If

This rule is always true. To add an expression please uncheck the Always True checkbox.

Then

Set Job Information Model.Standard Weekly Hours.Value to be equal to Job Information Model.Work Schedule.Value.Average Hours Per Week

Set Job Information Model.Working Days Per Week.Value to be equal to Job Information Model.Work Schedule.Value.Average Working Days Per Week

- Assign above rule as onChange rule to Work Schedule field in Job Information.

- Set up rule to calculate FTE and assign it to Standard Weekly Hours field in Job Information as onChange rule.

Calculates FTE. Only Position is used for Full Time Std Hours reference (MC_CalcFTE)

Scenario: Basic [Change Scenario](#)

Basic Information		Parameters	
Start Date	01/01/1900	Name	Object
Rule Type	onChange (onChange)	Context	System Context
Description	Assigned onChange to Std Hours field on Job Info. Calculates FTE. Only Position is used for Full Time Std Hours reference	Job Information	Job Information
Show More			

Variables

If

This rule is always true.
To add an expression please uncheck the Always True checkbox.

Then

Set **Job Information.FTE** to be equal to **Calculate FTE based on Standard Hours()**

(Optional)	Weekly Standard Hours:	Job Information.Standard Weekly Hours
(Optional)	Standard Hours of Position:	Job Information.Position.Standard Weekly Hours
(Optional)	Standard Hours of Job Code:	Null
(Optional)	Standard Hours of Location:	Null
(Optional)	Standard Hours of Legal Entity:	Null

- To calculate FTE if standard method to calculate FTE is not possible

If a customer wants to have a different logic for calculating FTE than the hard coded logic explained above, it is required to disable first and then delete the predelivered Standard Hours field in Job Info. Then create a new custom-double field in JobInfo HRIS element and use this as your field for Standard Hours instead. You will also need a business rule to calculate FTE value.

This is not our recommendation and this solution should only be implemented if there is no way a customer can move to standard.

- FTE to be defaulted to 0 during LOA - If FTE needs to be defaulted to 0 when the employee is on a Leave of Absence, then following configuration is needed. You can make changes directly in XML or via BCUI. BCUI Steps are provided below:

- The pre-delivered standard-hours field needs to be disabled and removed via „Manage Business Configuration“. Navigate to Admin Center > Manage Business Configuration > jobInfo element > first set the "Enabled" status of the field to "No" and click Save. Then once the page refreshes, click Take Action > Make Correction and then click the trash can icon to the right of the standard-hours field and then click Save again.

standard-hours	Standard Weekly Hours	Job Information (jobInfo_jobFieldsList)	No
----------------	-----------------------	---	----

- With the standard-hours field removed entirely from the BCUI (as described above) - add a custom decimal field (custom-double) field to jobInfo and use this as your field for Standard Hours instead.

custom-double1	Standard Weekly Hours	Job Information (jobInfo_jobFieldsList)	Yes	Details
----------------	-----------------------	---	-----	-------------------------

- Create a new Business Rule using the Calculate Full-Time Equivalent (FTE) rule scenario. You can set as an onChange (on the custom <standard-hours> field) and/or an onSave rule (on jobInfo element) so that the FTE can be calculated differently depending on the requirement/scenario.

If

Job Information Model.Event Reason.Value.Event is equal to Leave of Absence (3671)

Then

Set Job Information Model.FTE.Value to be equal to 0

Custom FTE field should never be used as other areas of the application use the "FTE" field in Job Information element (FTE for Position Management, calculation of Compa-Ratio & Range Penetration etc.), so for these other features to work correctly, **the pre-delivered FTE field should be used.**

6.2.2 FTE as a calculation basis

FTE value influences payment and leave eligibility of an employee. It's important to ensure that FTE value is correct within the system at all times.

- Range penetration / compa-ratio calculation: If FTE is not maintained correctly, this could cause erroneous calculations with regards to range penetration / compa-ratio. The result could be an incorrect guideline recommendation in other modules such as compensation.

If the Compa Ratio and Range Penetration calculation is required without the check for an FTE value, follow the steps below:

- Navigate to Admin Center
- Search and select Company System and Logo Settings
- Now enable the flag "Calculate Range Penetration and Compa Ratio even if FTE is null"

Before enabling the switch: -

☐ Calculate Range Penetration or Compa Ratio even if FTE and Standard Weekly Hours are null.

Compa Ratio ?	Error: The key figures could not be calculated as the FTE is initial.
Range Penetration ?	Error: The key figures could not be calculated as the FTE is initial.

After enabling the switch: -

☒ Calculate Range Penetration or Compa Ratio even if FTE and Standard Weekly Hours are null.

Compa Ratio ?	9.1%
Range Penetration ?	-246.83%

- Part-time pay calculation: - A company may have 2 pay components, one for full-time employees and one for part-time employees. Employees have both pay components in their compensation information data. The amount for the part-time pay is calculated using a business rule based on the full-time pay amount and the employee's FTE.

Create Compensation Information Model.Salary Information.Compensation	Populate Compensation Information Model.Salary Information.Compensation with:
Amount	Multiply() Value: Divide() Dividend: Compensation Information Model.Total Allowance.Value Divisor: var_FTE Factor: Compensation Information Model.Employment Details Model.Job Information Model.FTE.Value
Currency	INR
Event Date	Compensation Information Model.Event Date.Value
Frequency	Monthly (MON)
Pay Component	Flexi Pay India (FLEXIPAY_IND)

- Leave Eligibility: - FTE impacts leave eligibility and Time-off calculations in some cases. Employees get leaves based on FTE. Sometimes, it gets recalculated if there is a change in FTE.

Set variables.Amount to be equal to	Lookup(Accrual Lookup By Seniority New)
Select	Accrual Amount Annual Leave
where...	Accrual Lookup By Seniority New. Country is equal to Time Account.User.Job Information.Company.Country Accrual Lookup By Seniority New. Seniority From <= variables.Seniority Accrual Lookup By Seniority New. Seniority To >= variables.Seniority

Create Time Account.Time Account Details	Populate Time Account.Time Account Details with:
Amount posted	Multiply() Value: Time Account.User.Job Information.FTE Factor: variables.Amount
External Code	Generate External Code For Time Off()
Posting Date	variables.Posting Date
Posting Type	Accrual
Posting Unit	Time Account.Time Account Type.Unit

6.2.3 FTE Rule Functions

- Average FTE of a period
 - Average Full-Time Equivalent for Accrual Period
 - Average Full-Time Equivalent for Accruable Period

These functions calculate the average full-time equivalent (FTE) as defined in the job information entity for the accrual period. The function is used for fields that show data in numeric form, to calculate accruals depending on FTE. If there are several effective dated FTE changes during a given year / period, this value calculates an average amount. So, this means that all FTE changes made in that year lead to another accrual result/recalculation.

```

var_AverageFTEAmount = Multiply()
    Value: 76
    Factor: Average full-time equivalent for Accrual Period()
        User: Time Account.User
        Time Account Type: Personal Leave (PL)
        Start Date: Date Plus()
            Base Date: accrualRuleParameters.Accruable Start Date
            (Optional) Number of Months: -12
            (Optional) Number of Days: 0
        End Date: Date Plus()
            Base Date: Date Plus()
            Base Date: accrualRuleParameters.Accruable Start Date
            (Optional) Number of Months: -12
            (Optional) Number of Days: 0
            (Optional) Number of Months: 12
            (Optional) Number of Days: 0
        Accruable Start Date: Date Plus()
            Base Date: accrualRuleParameters.Accruable Start Date
            (Optional) Number of Months: -12
            (Optional) Number of Days: 0
        Accruable End Date: Date Plus()
            Base Date: Date Plus()
            Base Date: accrualRuleParameters.Accruable Start Date
            (Optional) Number of Months: -12
            (Optional) Number of Days: 0
            (Optional) Number of Months: 12
            (Optional) Number of Days: 0

```

- FTE across employments
 - Get Maximum Total FTE for Time Period

This rule function helps to generate error message if FTE across all Employments exceed a limit. However, this rule function is only supported for Concurrent Employment scenarios.

Message definition needs to be created first as the system should throw an error when FTE is exceeded across multiple employments.

Message Definition: FTE_Check (FTE_Check)

* Text The FTE across all Employments is greater than 1 ⓘ

* External Code FTE_Check

* Name FTE_Check

* Status Active

If

Get Maximum Total FTE for Time Period()

(Optional) Person ID for whom maximum FTE is to be calculated: Job Information.Employment Details.Biographical Information.Person ID > 1

(Optional) Start of time period for maximum FTE calculation: Job Information.Event Date

(Optional) End of time period for maximum FTE calculation: Date Plus()

Base Date: Job Information.Event Date

(Optional) Number of Months: 12

(Optional) Number of Days: 0

(Optional) Job Information: Job Information

Then

Raise Message " FTE_Check " with Error severity

The FTE across all Employments is greater than 1

Rule Trigger Points:

- OnChange rule to be assigned to FTE field in jobInfo HRIS element
- OnSave rule to be assigned on the jobInfo HRIS element
- Date on which Maximum FTE is exceeded:
 - Get Date of Maximum Total FTE for Time Period

This rule function can be used to generate error message if FTE across all Employments exceeds a limit.

First step is to create message definition to return the Date on which Maximum FTE is exceeded.

Message Definition: FTE_Check (FTE_Check)

- * Text The FTE across all Employments is greater than 1
- * External Code FTE_Check
- * Name FTE_Check
- * Status Active

Parameters

Data Type	External Code	Name	Status
Date	Date	Date	Active

Next business rule should be created with If condition as shown below:

If

Get Maximum Total FTE for Time Period()

(Optional) Person ID for whom maximum FTE is to be calculated: Job Information.Employment Details.Biographical Information.Person ID

(Optional) Start of time period for maximum FTE calculation: Job Information.Event Date

(Optional) End of time period for maximum FTE calculation: Date Plus()

(Optional) Base Date: Job Information.Event Date

(Optional) Number of Months: 12

(Optional) Number of Days: 0

(Optional) Job Information: Job Information

In THEN -> Raise Message condition you will see a new field called "Parameter", which can be used to further configure the rule to return the "Date" based on the 2nd Rule Function "Get Date of Maximum Total FTE for Time Period ()".

It is advisable that the same configuration is used when setting the 4 values to check (Person ID, Start Date, End Date and Job Information - below example), as you used in the IF condition check (above):

Then

Raise Message " FTE_Check " with Error severity

The FTE across all Employments is greater than 1

Date: Get Date of Maximum Total FTE for Time Period()

(Optional) Person ID for whom the date of maximum FTE is to be calculated: Job Information.Employment Details.Biographical Information.Person ID

(Optional) Start of time period for the date of maximum FTE calculation: Job Information.Event Date

(Optional) End of time period for the date of maximum FTE calculation: Date Plus()

(Optional) Base Date: Job Information.Event Date

(Optional) Number of Months: 12

(Optional) Number of Days: 0

(Optional) Job Information: Job Information

6.3 Cross Entity Rules

Cross Entity Rules - Cross-entity rules are triggered from the source entity and changes are executed on the target entity. For example, when changes in Job Information update Compensation Information. A maximum of 5 cross-entity rule scenarios are supported for an HRIS entity.

Cross entity rule is a rule which can update/create/delete a field/row in another entity. Currently it is supported only for specific employment-related entities, which includes:

- Job information
- Compensation Information
- Pay component Recurring
- Pay Component Non-Recurring
- Job Relationships

- Employment Details

Cross Entity Business rules must be created using Rule scenario 'Cross-Entity Rules' available under Employee Central Core.

Employee Central Core

☐ **Generate Assignment ID External**
 Use this scenario to create the rule that generates the value for Assignment ID External based on MDF Sequence objects. Create a single rule only based on this scenario. Please follow the detailed documentation in the SAP Help Portal under the search term 'Assignment ID External'.

☐ **Generate Employee ID For Hire/Rehire**
 You can use this scenario to generate an Employee ID from the Metadata Framework Sequence (MDF) and assign it to the User ID field of the Employee Information object during the Hire/Rehire with new employment process. You must first register the rule in the Matching User Configuration object settings.

☐ **Rules For Hire/Rehire**
 You can use this scenario to configure rules on the Employee Information object during the Hire/Rehire with new employment process.

☐ **Event Reason Derivation**
 You can use this scenario to derive the event reason automatically for Job and Compensation Information entities.

☐ **Generate Alerts**
 You can use this scenario to create rules that generate alerts for Employee Central data, for example, alerts for job information changes.

☐ **Enforce New Employment for Rehire**
 You can use this scenario to configure a rule that validates the business requirements to enforce new employment and returns an error message if the conditions are not met.

☐ **Trigger Workflows**
 You can use this scenario to create a rule that triggers Employee Central workflows to approve data changes.

☐ **Internal Job History**
 You can use this scenario to configure the rule for the Internal Job History Block on the People Profile page.

☐ **Validation for HRIS Elements**
 You can use this scenario to validate HRIS elements and to raise messages.

☐ **Calculate Full-Time Equivalent**
 You can use this scenario to calculate full-time equivalent for a user using the Job Information Model base object.

☐ **Cross-Entity Rules**
 You can use this scenario to configure cross-entity rules that are triggered from the source entity and changes executed on the target entity.

Base Object*

No Selection

Compensation Information Model

Compensation Model

Employment Details Model

Job Information Model

Job Relationships Model

One-Time Payments Model

cross-entity rule.

Continue

The source / target direction is very important. The source element must be the base object of the rule.

Cross entity rules can be used in following scenarios:

- Change in Job Information entity data that then causes change in Compensation Information entity data, for example, transfer to a different location causes addition of a pay component
- Change in Job Information entity data that then updates Job Relationships. For example, change in department causes change in HR Manager
- Changes to Job Information entity data that then changes (create, update, delete) Recurring Pay Components. For example, pay scale level, FTE.

Support for onChange cross entity is introduced for a limited functionality. It works for Job information, Compensation Information and Pay component Recurring when both Job Information and Compensation Information Change checkboxes are selected in MSS UI (Take action) .

▼

Choose what you want to change

☒

Job Information

Change employee's job data, time information, and other information.

☐

Job Relationships

Specify the employer's HR Business Partner, legal advisors, and others besides the primary manager.


☒

Compensation Information

Change the salary, bonus, eligibility for benefits, and other information.

When would you like your changes to take effect? *

MMM dd, yyyy



- Rule using create expression

Please refer to Employee Central implementation guide to check supported Source Elements and corresponding Target elements for creating rules using create expression [Cross-Entity Rules - SAP Help Portal](#). Provided below is a sample rule using create expression.

Scenario: Pay Component and details in Compensation can be defaulted based on Location in Job Information.

Scenario: Cross-Entity Rules
[Change Scenario](#)

Basic Information

Start Date	01/01/1900
Description	
Base Object	Job Information Model

Parameters

Name	Object
Context	System Context
Job Information Model	Job Information Model

Variables

If

and

Job Information Model.Company.Value is equal to BestRun Japan (5000)

Job Information Model.Location.Value is not equal to Job Information Model.Location.Previous Value

or

Job Information Model.Location.Value is equal to Hokkaido (5000-0021)

Job Information Model.Location.Value is equal to Aomori (5000-0022)

Then

Create Job Information Model.Employment Details Model.Compensation

Populate Job Information Model.Employment Details Model.Compensation with:

Event Date	Job Information Model.Event Date.Value
Pay Component	Cold District Allowance(JP) (Cold_District_Allowance_JP)
Amount	30000
Currency	JPY
Frequency	Monthly (Monthly)

Scenario- Based on changes in Pay Scale Level in Job Information; Pay components, amount and other details can be defaulted in Compensation from Pay Scale Level.

Start Date	01/01/1900	Name	Context	Object	System Context
Rule Type	Creates the necessary compensation rule when the criteria for Pay Scale progression are met such as PayScaleLevel L1 --> L2		Job Information Model	Job Information Model	
Description					
Show More					

Variables

If

Job Information Model.Pay Scale Level.Value is not equal to Job Information Model.Pay Scale Level.Previous Value

Job Information Model.Company.Value.Country is not equal to Germany (DEU)

Job Information Model.Pay Scale Level.Value.Pay Component Assignment ≡.Amount not equal to Null
The rule selects one entry from the collection "Pay Component Assignment" where...

Then

Create Job Information Model.Employment Details Model.Compensation ≡

Populate Job Information Model.Employment Details Model.Compensation ≡ with:

Amount	Job Information Model.Pay Scale Level.Value.Pay Component Assignment ≡.Amount The rule selects one entry from the collection "Pay Component Assignment" where...
Currency	Job Information Model.Employment Details Model.Compensation ≡.Currency The rule selects one entry from the collection "Compensation" where...
Event Date	Job Information Model.Event Date.Value
Frequency	Job Information Model.Employment Details Model.Compensation ≡.Frequency The rule selects one entry from the collection "Compensation" where...
Pay Component	Job Information Model.Employment Details Model.Compensation ≡.Pay Component The rule selects one entry from the collection "Compensation" where...

Raise Message " PayScaleLevelPayChg " with Warning severity
An automatic pay change will be created with amount associated to the new Pay Scale Level.

- Rule using set expression

Please refer to Employee Central implementation guide to check supported Source Elements and corresponding Target elements for rules using set expression [Cross-Entity Rules - SAP Help Portal](#). Provided below are sample rules using set expression.

Scenario– Requirement to update Job relationship Name (HR Manager) with update in Job Information Supervisor

Scenario: Basic [Change Scenario](#)

Basic Information	Parameters						
Start Date	01/01/1900						
Rule Type	onSave (onSave)						
Description	Rehire for benefits						
	<table border="1"> <tr> <th>Name</th> <th>Object</th> </tr> <tr> <td>Context</td> <td>System Context</td> </tr> <tr> <td>Job Information Model</td> <td>Job Information Model</td> </tr> </table>	Name	Object	Context	System Context	Job Information Model	Job Information Model
Name	Object						
Context	System Context						
Job Information Model	Job Information Model						

Variables

If

Job Information Model.Supervisor.Value is not equal to Job Information Model.Supervisor.Previous Value

Then

Set Job Information Model.Job Information.Employment Details.Job Relationships ≡.Name to be equal to Job Information Model.Job Information.Supervisor
The rule selects one entry from the collection "Job Relationships" where...
Relationship Type is equal to HR Manager (3782)

Scenario: Requirement to update Pay Group with change in Job Information Job Classification

Scenario: Basic [Change Scenario](#)

Basic Information

Start Date 01/01/1901

Rule Type onSave (onSave)

Description

Parameters

Name	Object
Context	System Context
Job Information Model	Job Information Model

Variables

If

Job Information Model.Job Classification.Value is equal to Consulting/Analyst (50012557)

Then

Set Job Information Model.Employment Details Model.Compensation Information ≡.Pay Group to be equal to India Monthly (N6)
The rule selects one entry from the collection "Compensation Information" where...

- Rule using delete expression

Please refer to Employee Central implementation guide to check supported Source Elements and corresponding Target elements for rules using delete expression [Cross-Entity Rules - SAP Help Portal](#). Provided below are sample rules using delete expression.

Scenario—Change in pay components resulting from change in Employee Class eg, employee's contract changed from regular employee to contractual worker.

Scenario: Cross-Entity Rules [Change Scenario](#)

Basic Information

Start Date 02/01/1900

Description

Base Object Job Information Model

Parameters

Name	Object
Context	System Context
Job Information Model	Job Information Model

Variables

If

and

- Job Information Model.Employee Class.Value is not equal to Job Information Model.Employee Class.Previous Value
- Job Information Model.Employee Class.Value is equal to Contractor (30092)

Then

Delete Job Information Model.Employment Details Model.Compensation ≡
Select Compensation where...
Pay Component is equal to Base Salary Singapore (BASESAL_SG)

Standard Employee Central Configuration guide contains a table ([Link to documentation](#)) that summarizes supported operations between entities for create, modify and delete.

The History UI as well as Imports only support onSave rules for cross-entity rules. Generally, onChange rules work when both entities are displayed on the UI, for example, in Manager Self-Service UIs.

Note: Job Information and Compensation Information as the target element do not support updates of existing records. Cross-entity rules with Job Information or Compensation Information as the target must use the Set command and this always results in the creation of a new record.

IDP document: Employee Central: Managing Indirect Valuation of Pay Components, can be referred to understand other scenarios of cross entity rule ([Link](#)).

- Troubleshooting Tips

If the cross-entity rule is not working:

- Ensure that the source and destination entities are supported
- Ensure that the base object used is correct.
- Cross-Entity business rule scenario is used for creating rules
- Operators - create / Set / Delete, are supported for source and target entities. Full list of supported operators for source and target entities is available in [Employee Central Configuration Guide](#).
- Execute the rule trace and review the Business Rule Execution Log
- Check the rule assignment-
 - Is the rule assigned under correct HRIS element as onSave / onChange rule based on scenario
 - Check the order of the rule. You may need to change the order in which rules are executed to ensure that the rule works as required.

6.4 Refresh Compensation Required Feature

It is possible to set up the system to automatically refresh compensation information when job information fields are updated via Take Action – Change Job and Compensation Information.

The „Refresh Compensation Required“ field is set in Manage Business Configuration → jobInfo HRIS element. You will need to set this to Yes on the fields where the refresh should be enabled:

The screenshot displays the 'Manage Business Configuration' form for the 'jobInfo' HRIS element. The form contains various configuration fields for the 'Standard Weekly Hours' field. The 'Refresh Compensation Required' field at the bottom is highlighted with a red rectangular box, indicating it should be set to 'Yes'.

Label	Standard Weekly Hours
Section	Job Information (jobInfo_jobFi...
Enabled	Yes
Mandatory	Yes
Default Label	Standard Weekly Hours
Data Type	DOUBLE
Type of Reference Object	No Selection
Reference Object	Click or focus to edit
Picklist	No Selection
Parent Field for Picklist	No Selection
Visibility	Edit
Masked	No
Log Read Access	No Selection
Maximum Length	256
Maximum Fraction Digits	2
Show Trailing Zeros	No
Allow Import	No
Refresh Compensation Required	Yes

Let us consider a scenario. You have a cross entity onSave business rule on the pay component recurring entity which recalculates a pay component amount when the Standard Weekly Hours are changed in Job Information entity. When the user Takes Action, they need to remember to select both Job Information and Compensation Information for the on save rule to trigger.

If the user forgets to select Compensation Information, the Standard Weekly Hours are changed, and the FTE is recalculated but the Compensation Info is not updated

Result when Refresh Compensation Required = No

▼ Choose what you want to change

☒ Job Information
Change employee's job data, time information, and other information.

☐ Job Relationships
Specify the employer's HR Business Partner, legal advisors, and others besides the primary manager.

☐ Compensation Information
Change the salary, bonus, eligibility for benefits, and other information.

Do you approve this request?

Change in Standard Hours for Gavin Forsyth

Initiated by EC ADMIN on 28 September 2021

Effective Date 28 September 2021

[View Workflow Participants](#)

Job Information	
Pension Protection (Fixed or Enhanced)	No
Standard Weekly Hours	20 ⁴⁰
FTE	0.5 [±]
Calculation Method Indicator	No
Country	United Kingdom

With this feature set to Yes **on the Standard Weekly Hours field**, the system will force a refresh of the Compensation Info and the on save rule will be triggered.

Result when Refresh Compensation Required = Yes

▼ Choose what you want to change

☒ Job Information
Change employee's job data, time information, and other information.

☐ Job Relationships
Specify the employer's HR Business Partner, legal advisors, and others besides the primary manager.

☐ Compensation Information
Change the salary, bonus, eligibility for benefits, and other information.

Workflow request shows both Job information and Compensation Information changes in approval screen.

Do you approve this request?

Change in Standard Hours for Gavin Forsyth

Initiated by EC ADMIN on 28 September 2021

Effective Date 28 September 2021

[View Workflow Participants](#)

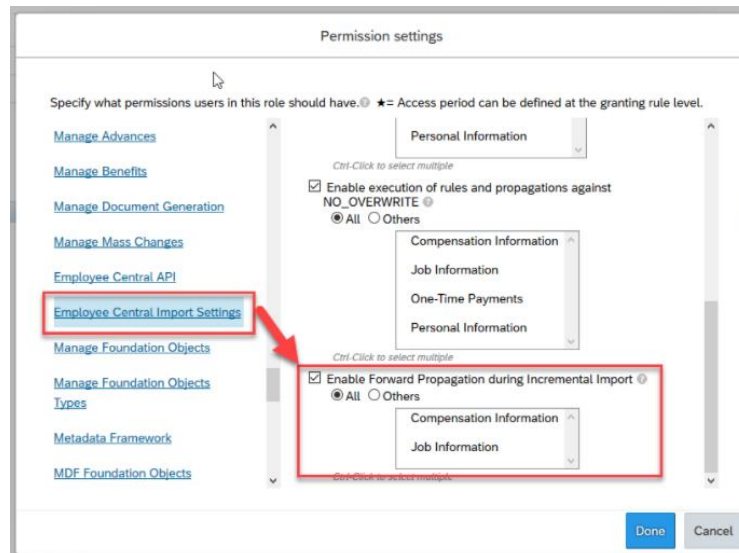
Job Information	
FTE	0.5 [±]
Country	United Kingdom
Standard Weekly Hours	20 ⁴⁰
Pension Protection (Fixed or Enhanced)	No
Calculation Method Indicator	No

Compensation Information	
Base Annual Salary (AnnualizedSalary) ⓘ	150,000 GBP 300,000 GBP
Total Earnings Opportunity (TotalEarningsOpp) ⓘ	150,000 GBP 300,000 GBP

Compensation			
Pay Component	Amount	Currency	Frequency
Base Salary (1002)	150,000 300,000	GBP	Annual (ANN)

6.5 Forward Propagation

Forward Propagation for Job Information is enabled by default in the UI (hard-coded) and cannot be disabled. You must also enable forward propagation for imports.



How does it work?

To understand forward propagation, let's assume that there are 3 fields in Job Information (Field 1, Field 2, and Field 3) and there are 3 records, as shown in table below. If changes are inserted, then forward propagation will happen for only those fields which had same values but not for one which had different values.

Start Date	End Date	Field 1	Field 2	Field 3
01.01.2021	31.03.2021	X1	Y1	Z1
01.05.2021	31.08.2021	X1	Y2	Z1
01.10.2021	31.12.9999	X2	Y3	Z1

Now let's assume that admin inserts another record with start date 01.03.2021.

Then the records in Job Info history will show following values:

Start Date	End Date	Field 1	Field 2	Field 3
01.01.2021	28.02.2021	X1	Y1	Z1
01.03.2021	30.04.2021	X3	Y4	Z2
01.05.2021	31.08.2021	X3	Y2	Z2
01.10.2021	31.12.9999	X2	Y3	Z2

When a new record is inserted, the new value in a field is propagated to future (subsequent) records till system can no longer find the previous value. Also, end date for each record is auto calculated by the system. In the example above, X1 is the existing value in Field 1. When a new record with value X3 is inserted, system replaces X1 in subsequent records till it no longer finds X1 in field 1. Similarly, for field 2 and 3.

It works only when we Insert a new Job Information record through Job History or Take Action (MSS UI) or imports. Forward Propagation happens for all insert transactions (e.g., time off sync, pos 2 job sync, UI transactions, import, transfer direct reports, etc.). Forward Propagation does not occur when updating records through Job History > "Make Correction" nor when a record is Deleted via Job History. Also, not in rehire case.

6.6 Job Information History

As mentioned in Employee Central Configuration Guide, only Data Change events should be inserted or created in Job History. Please note that with 2111 release, you can no longer create new records, or edit the event field of records, for the events listed below using the Job Information History UI. Instead, these transactions must be made by selecting appropriate option under "Take Action" (e.g., Add Global Assignment) or by using specific wizards available for such events (e.g., Add New Employee). This ensures that the status of the employee, along with their employment and/or termination details, is updated correctly. A full purge import can also be used to make these changes. Here is the list of affected events:

- Hire (H)
- Termination (26)
- Rehire (R)
- Leave of Absence (10)
- Return to Work (22)
- No Show (NS)
- Add Global Assignment (GA)
- Away on Global Assignment (AGA)
- Obsolete (OGA)
- Back from Global Assignment (BGA)
- End Global Assignment (EGA)
- Start Pension Payout (SPP)
- End Pension Payout (EPP)
- Discard Pension Payout (OPP)
- Surviving Spouse Start (SSS)
- Surviving Spouse End (ESS)
- Work Order End (ECWK)
- Add Higher Duty/Temp Assignment (HD)
- End Higher Duty/Temp Assignment (END_HD)
- Obsolete Higher Duty/Temporary Assignment (OHD)

When you edit a record in the user's Job Information history for an event that isn't listed above, you can't select any of the events listed above. However, you can change the event reason within the same event.

An issue sometimes observed with Job information records is that they don't show up in employee profile, but the records are available in history. This happens when the last Job Information record has an end-date set in the past. The end-date should always be 12/31/9999 for the last Job Information record. If it is a date in the past, this issue will occur.

To resolve this issue:

- Delete the last Job Information record with invalid end-date and recreate it. If you are doing this via Imports, do not enter any end-date. The system will calculate this automatically.
- If the first option does not resolve the issue, you must perform a Job History Full Purge Import.

6.7 TimeIn Calculations

Time in Job/Position/Company/Department/Location/Pay scale level is calculated based on Entry Date for respective fields –Job Entry Date/Position Entry Date/Company Entry Date/Department Entry Date/Location Entry Date/Pay Scale Entry date. These Entry date fields must be configured in succession data model.

Timein calculation is transient and is based on approximate values and hence may not be exactly accurate:

1 month = 30 days

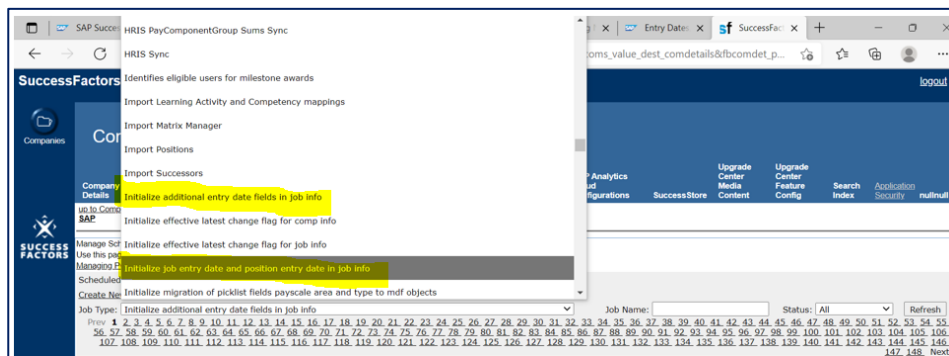
1 year = 360 (12*30)/361/362/363/364/365 days

If number of days is 370 then Timein calculation would yield 1 year (365 days), 0 months,5 days.

Please refer to Employee Central Configuration guide for details on setting up these fields. Please note that once the fields are added to the configuration, you will need to run following jobs to set the Entry Dates which the UI will use to calculate the "Time In" value:

- Initialize job entry date and position entry date in job info
- Initialize additional entry date fields in job info

These jobs should never be set up as recurring jobs.



These transient fields are calculated and filled in the Job Information History UI, Employment Information page and Manager Self-Service (MSS).

If Centralized services for Job information is enabled in the system, the entry dates are always automatically calculated, which means that values set using the UI, imports, or business rules are overwritten by the automatically calculated value.

We recommended that you set the visibility to 'View' for these fields in the Succession Data Model or the Country/Region-Specific Succession Data Model. If one of the base fields changes, all entry dates in the record are recalculated. Users should not have insert and correct permissions for these fields. Only view permission should be granted for these transient fields.

6.8 Synchronization of data from Position Object when using Time-Off

Position to Job sync rule is used to synchronize data for common fields from the position object to incumbent's job information. This rule is assigned differently depending on the requirement.

During the hiring process and job information changes the rule that is used is stored as 'onchange' rule on the position field in job information.

If you make a change to a position via the position organization chart or via import, the rule used to synchronise the data is stored in position management settings.

If you have Time-Off enabled and you have standard weekly hours on the sync rule then it is worth considering creating an additional rule that doesn't have the standard weekly hours in the rule that is triggered from the position. If you do have Standard Weekly Hours in the sync rule and you update the position via the org chart or import, then this could adjust the FTE on the employees record and hence initiate an accrual recalculation.

6.9 Order of the fields on Job information

As you progress through the deployment of Employee Central you will no doubt add in many custom fields to meet the customer's requirements. When custom fields are added they are typically added at the bottom in job information HRIS element.

When you view a job information record, the order of the field reflects how the fields are ordered in "Manage Business Configuration UI". While when you edit a record, they reflect the order in "Configure People Profile". Therefore, it is very important that the order of the fields in edit mode reflect any logic relating to any 'onchange' rules.

Once you have finished all the configuration move the fields in "Configure People Profile" so that you reduce any error relating to cyclical rules.

Lets consider an example. Lets change the order of cost centre field in "Configure People Profile" as shown below, and save.

Navigate to employee profile and view organization Information. Notice that “Cost Center” field is placed after ‘TimeZone field’.

This order is as available in Manage Business Configuration UI.

Identifier	Label	Section	Enabled	(21) More
position	Position	Q Position Information (jobInfo_...	Yes	Details
positionEntryDate	Position Entry Date	Q Position Information (jobInfo_...	Yes	Details
timeInPosition	Time In Position	Q Position Information (jobInfo_...	Yes	Details
company	Company	Q Organization Information (jobl...	Yes	Details
business-unit	Business Unit	Q Organization Information (jobl...	Yes	Details
division	Division	Q Organization Information (jobl...	Yes	Details
department	Department	Q Organization Information (jobl...	Yes	Details
location	Location	Q Organization Information (jobl...	Yes	Details
timezone	Timezone	Q Organization Information (jobl...	Yes	Details
cost-center	Cost Center	Q Organization Information (jobl...	Yes	Details
manager-id	Supervisor	Q Job Information (jobInfo_jobFl...	Yes	Details

Edit the Job Information. Notice that Cost Centre field is placed before the TimeZone field, as available in Configure People Profile.

Position Information

Incumbent of Parent Position: No Selection

Position*: HR Administrator (30...)

Position Entry Date: Mar 17, 2002

Time In Position: 19 Years 8 Months 16 Days

Organization Information

Company*: BestRun Company (1710)

Business Unit*: Corporate (CORP)

Division: Corporate Services (CORP_SVCS)

Department: Global Human Resources (50007726)

Location*: Corporate - US-Philadelphia (1710-2001)

Cost Center: BestRun Corp Services (US10_M1)

Timezone*: US/Eastern (GMT-05:00)

6.10 Job Info History Edit Vs take Action

One can update/change Job information records in following ways:

- By editing the history – This is an option of editing existing record along with adding any missing record.

Job Information [Edit Icon] [Clock Icon]

Effective as of: Jan 1, 2017

Job Information History

Please note that with 2111 release, you can no longer create new records, or edit the event field of records, for the events listed in “Job Information History” section above.

- Take Action/ Edit Job Information - This option allows users with access (manager and admins), to initiate new changes with respect to Job information of an employee

Actions [Header] [As of Today]

Take Action

Change Job and Compensation Info

Job Information [Edit Icon] [Clock Icon]

Effective as of: Jan 1, 2017

Edit Job Information

Employee Status: Active

It is necessary to understand which one to use when, as there are differences in system behaviour based on option selected. Following table summarizes these differences:

Feature	Editing Job Information History	Editing Job information using Take Action/Edit Job Information
Event Reason	Needs to be manually selected	If rule is used, this is derived based on event reason derivation rules
Workflow	Workflow is not supported	Workflow is supported
Role	Usually Admin roles have this access	Usually Manager (MSS) along with Admin roles have this access
Data Inconsistency	More prone to error as event reason selection is manual	Recommended for making changes due to greater data consistency. Of the two, “Take Action” is the recommended approach to make any updates.

6.11 Restriction of using 'onview' rules 'onChange'

Onview rules are useful rules to control visibility of fields when the page loads. The data is not stored in the database and hence can't be reported on. Onview rules can't be added to a field and triggered when the data changes. For example, we have two fields – Date of Birth and Age and Age is populated by an onView rule. As the data is not stored in the database, you cannot use the Age field to trigger any rule.

6.12 Termination

Event reasons associated with Termination event should have Employee Status as either Terminated or Retired.

Event Reason: Voluntary Separation Program (TERVVSP)

Effective as of 01/01/1970

Blue indicates that the item changed on this date

Event Reason ID	TERVVSP
Event Reason Name	Voluntary Separation Program ?
Description	Voluntary Separation Program ?
Status	Active
Event	Termination
Employee Status	Terminated
Follow-Up Activity in Position	No Selection
Payroll Event	
Display in Internal Job History Portlet	No

If this is not maintained correctly, employee will still appear “Active” in the system.

To confirm, navigate to terminated employee's job Information record and check the last record which **must be with event** “Termination”. Employee status is visible on the record and should be “Inactive”. If this is not the case, then employee status linked with the Termination Event Reason needs to be updated as shown in screenshot above.

The system does not stop a user from adding another record after termination. Usually, MSS users do not have access to inactive employees. But Admin users may have access to edit Job Information/Job Information history record. In such cases, users must be careful while adding any record after Termination record as this can cause a reset of employee status to Active, depending on the Event Reason and associated Employee Status.

There are other reasons which can cause the employee status to be active post Termination e.g.,

- Termination Event does not have a valid External Code.
- Employee Status picklist has incorrect External Code set against Termination status etc.

Please refer to configuration guide for other reasons and approach to fix these issues.

If any data correction is needed post termination, a data change event with employee status “No Selection” can be used.

Data issues also arise in Job Information entity due to upload of incorrect data. eg, data is loaded for inactive employee with “Active” status for a date post termination date etc. Employee data imports should be carefully validated before upload.

6.13 Position Management “Right to Return” feature and Impacts on Job Information

In the position management settings you have the option to unassign an employee from a position when they are on a leave of absence or on a global assignment. This is configured under Right to Return tab.

Back to: Admin Centre
Position Management Settings

General Hierarchy Adaptation Synchronisation UI Customising **Right to Return** Transition Period Integration Import

You can use the options on this tab to regulate the right to return from a leave of absence or global assignment. You only need them if you are using Position Management in conjunction with Time Off, Global Assignments or Higher Duty Assignments.

Right to Return for Leave of Absence

Unassign from Position:

Create Right to Return:

Right to Return for Global Assignment

Unassign from Position:

Create Right to Return:

Event Reason for unassigned Position:

Event Reason for assign Position:

What is the impact on Job Info?

In the example below the employee has been unassigned from their position because they are on a leave of absence:

Change History	Job Information Changes
<p>22 Oct 2021</p> <p>Parental Leave</p> <p>Position Automotive Production Manager Leeds Plant-UK (Position000039)</p> <p>10 Sep 2021</p> <p>Location Change</p> <p>Location Bristol Plant UK (MC_UK_BRI) Leeds Regional HQ-UK (MC_UK_LEE)</p> <p>Cost Centre MC Sales UK (MC_100009) MC Auto Production UK (MC_100015)</p>	<p>Last updated by EC ADMIN (ECADMIN) on Friday, 22 October 2021 03:39:49 EDT</p> <p>Effective as of: 22 Oct 2021</p> <p>Employee Status</p> <p>Employee Status: Unpaid Leave</p> <p>Event</p> <p>Event: Leave of Absence</p> <p>Event Reason: Parental Leave (LOAPAR)</p> <p>Position Information</p> <p>No data</p> <p>Organisation Information</p>

If the Job Information record needs to be updated while the person is still on a leave of absence and if the position field is marked as mandatory in the data model, the system will display this error:

⊗ 1 error(s) occurred. ?

1. Position is required

Close

If you have set the position field as mandatory in the data model and you are using the right to return functionality the following solution can be implemented:

- Make the position field non mandatory in the data model

* Identifier: position

Label: Position

Section: Position Information (jobInfo_...)

Enabled: Yes

Mandatory: No

Default Label: Position

- Create two business rules to control whether the position field is mandatory - one to trigger on hire and one to trigger on update of job information.

Employee Information Model	onInit	PositionNonMandatoryHIRE (PositionNonMandatoryHIRE)	Yes	Details
Job Information Model	onView	PositionMandatoryCHANGE (PositionMandatoryCHANGE)	Yes	Details

PositionNonMandatoryHIRE (PositionNonMandatoryHIRE)
 Scenario: Rules For Recruit/Rehire [Change Scenario](#)

Basic Information
 Start Date: 01/01/1900
 Description:
 Base Object Type: model

Parameters

Name	Object
Context	System Context
Employee Information Model	Employee Information Model

☒ Variables

☒ If

Employee Information Model.Job Information.Event Reason.Value.Event is equal to Leave of Absence (916)

Then

Set Employee Information Model.Job Information.Position.Required to be equal to false

Else

Set Employee Information Model.Job Information.Position.Required to be equal to true

PositionMandatoryCHANGE (PositionMandatoryCHANGE)
 Scenario: Basic [Change Scenario](#)

Basic Information
 Start Date: 01/01/1900
 Rule Type:
 Description:

Parameters

Name	Object
Context	System Context
Job Information Model	Job Information Model

☒ Variables

☒ If

Job Information Model.Event Reason.Value.Event is equal to Leave of Absence (916)

or

Job Information Model.Event Reason.Value.Event is equal to Termination (941)

Then

Set Job Information Model.Position.Required to be equal to false

Else

Set Job Information Model.Position.Required to be equal to true

7 REFERENCES

SAP Help Portal

Implementing Employee Central Core:

<https://help.sap.com/viewer/b14dd15ca58f43e0856184a740a4b212/LATEST/en-US/c922480d9aea440c818aff2b9a9a771e.html>

SAP SuccessFactors Data Model Reference Guide:

<https://help.sap.com/viewer/b05b0831c7a540739a2d19f01fbeatff/LATEST/en-US/72f0dabee3e442e4baf13380157db041.html>

What's New

[SAP SuccessFactors Release Information - SAP Help Portal](#)

SAP Notes/KBA

[2160734 - Forward Propagation of Job Information or Compensation Information Data - SAP ONE Support Launchpad](#)

2541909 - Job Info: How to configure the Event Reason (event-reason) - SAP ONE Support Launchpad

2333910 - Unable to calculate Compa ratio value - SAP ONE Support Launchpad

www.sap.com/contactsap

© 2021 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. See www.sap.com/copyright for additional trademark information and notices.