SAP SuccessFactors

PUBLIC

Document Version: 1H 2025 – 2025-04-11

Implementing Business Rules in SAP SuccessFactors

THE BEST RUN SAP

Content

1 Data Protection and Privacy for Business Rules

Find out about the data protection and privacy specifics for business rules.

Companies store a wide range of personal data on people, ranging from basic details like name and date of birth, to more potentially sensitive information such as religion or medical history. When using business rules, here are some things you should keep in mind in order to ensure good data protection and privacy for your users.

- Fields can be designated as sensitive, meaning that any information in the field will be hidden in order to ensure it cannot be seen by unauthorized users (for example – *Union Affiliation*: ******). However, if you use a rule to map the information in this field to a non-sensitive field, the information will then be freely visible in the non-sensitive field. As such, we strongly recommend that you only map sensitive fields to other sensitive fields.
- When a user runs a rule trace, a log file is created containing all the data gathered by the trace. There are two separate permissions for running the rule trace and viewing the resulting log file, and we recommend you only assign the viewing permission to the necessary users. In this way, you can restrict the number of people that can see potentially sensitive data.

2 An Introduction to Business Rules

Business rules are used to add application logic to determine the outcome of a change made to particular data in the system. For example, you can set up business rules to trigger certain actions when data is added, changed, or deleted from the system.

Business rules follow the logic 'If this data is changed in a certain way, then the system reacts in this way'. You can configure business rules that cover the following requirements:

- Legal regulations
  Example: If the employee works in the USA, then the FLSA status is required.
- Company policies
  Example: If employees move to the London office, then they get a compensation for the high cost of living.
- Other company-specific requirements

These requirements vary from customer to customer, and also depend on which objects and fields are available in the system. That's why business rules can't be delivered 'out-of-the-box', but have to be highly configurable. By configuring business rules, you can create such customer-specific rules.

> **Note**
>
> You can configure the business rule logic for various SAP SuccessFactors applications. Please refer to the application-specific documentation to check if there are application-specific guidelines for creating business rules.

> **Note**
>
> We recommend to use application-specific rule scenarios when creating rules, as rule scenarios provide more guidance about the supported objects, parameters, or actions you can use to configure the rule. For existing rules using the *Basic* rule scenario, we recommend that you change the scenario to an application-specific rule scenario (if available).

Related Information

Overview of Application-Specific Rule Scenario Documentation [page 68]

2.1 Process Overview for Configuring Business Rules

Configuring business rules is a two-step process, and consists of the creation and assignment of the business rule.

To configure business rules, you have to follow the steps described in the following table. Please note that for application-specific rules, some of these steps might be redundant. For more information, refer to the application-specific documentation.

Step	Details	Sub-Steps	Details
1. Creation of business rules	In the business rule, you define which actions the system performs when specific conditions are met.	1.1 Creation of conditions	In the *If* section of a rule, you define which conditions need to be met. Please note that some rules don't require conditions, be it for application-specific reasons or because you define the rule as "always true".
		1.2 Creation of statements	In the *Then* section of a rule, you define how the system reacts when the conditions are met. Please note that some application-specific rules don't require Then statements.
2. Assignment of business rules	The business rule is only triggered when it has been assigned to the corresponding object, field, business process, screen, or other entity as defined by the application.	2.1 Assignment of the rule to the corresponding object or entity	In some application-specific scenarios, the rule is assigned to the business process on an application-specific UI. In such a case, the underlying object isn't always visible to the user.
		2.2 Definition of the event that triggers the rule	For example, you choose the event that triggers the rule when the user calls up a specific page in the system. Or you choose the event that triggers the rule when the user saves changes to the object the rule is assigned to.

During runtime, the system executes the assigned business rules.

2.2 Use Cases for Configuring Business Rules

Business rules are configured, for example, to set default values, conditional values, or field properties, or to display error messages.

You can set up business rules to do the following:

- **Set default values**
  You can define default values for specific fields.
  Example: The start date field that needs to be maintained by the user is automatically set to the current date.
- **Set conditional values**
  You can define which default value is set when a specific condition is met: If this condition is met, then this is how the system should react.
  Example: When the admin selects the business unit *ENG*, the job classification is automatically set to *Engineering*.

- **Set field properties**
  You can dynamically default a field as visible or required.
  Example: If the company is *COMP_USA*, the phone extension is always required.

  > **Note**
  >
  > However, hiding all fields in a card using a business rule isn't supported and potentially causes unexpected behavior in the system. You must have at least one field on this object enabled to avoid inconsistent behavior.

- **Display error messages**
  You can define that an error message is displayed.
  Example: The user forgot to maintain a required field and gets a customer-specific error message displayed when trying to save.
- **Calculate transient fields**
  You can define transient fields that are calculated "on the fly" when the user opens a page. The calculated values aren't meant to be written to the database, as they aren't fixed values.
  Example: The user can see the employee's current age in the system.
- **Validate consistency of fields**
  You can define that all relevant fields are provided.
  Example: If an admin selects a *Contract Type* with fixed term validity, the *Contract End Date* needs to be provided. This is automatically checked.
- **Trigger events**
  You can define that certain actions in the system trigger an event to the corresponding application.
  Example: When the start date or hiring manager of a new hire changes, the onboarding process is restarted automatically.

2.3 Processing Order of Business Rules

The different parts of a business rule are processed from top to bottom, as displayed on the *Configure Business Rules* page. An exception are variables, which are processed before the section in which they're used.

For any deviations or specific considerations regarding the processing order, refer to the application-specific documentation of the corresponding business process.

Related Information

Order of MDF-Based Rule Execution

3 Implementation of Business Rules

There are some initial settings you need to make before you can configure business rules.

Business rules are used by many applications to configure system behavior. Please note that each application might use business rules in a different way, so make sure you understand the application-specific aspects as described in the application-specific documentation.

1. Setting Up Provisioning [page 13]
   Enable generic objects in Provisioning in order to configure business rules.
2. Permissions for Business Rules [page 14]
   You can only work with business rules if you make a few configuration settings and if you have the following permissions.
3. Setting Up the Application [page 16]
   You need to enable and set up the application for which you want to create business rules.

3.1 Setting Up Provisioning

Enable generic objects in Provisioning in order to configure business rules.

Prerequisites

> → Remember
>
> As a customer, you don't have access to Provisioning. To complete tasks in Provisioning, contact your implementation partner or Account Executive. For any non-implementation tasks, contact Technical Support.

Procedure

1. Go to Provisioning.
2. Select your company.
3. Under *Edit Company Settings*, select *Company Settings*.
4. Select *Enable Generic Objects*.
   This permission is the basis for using the Metadata Framework (MDF) on which business rules are based.
5. Scroll up and choose *Save Feature*.

Task overview: Implementation of Business Rules [page 13]

Next: Permissions for Business Rules [page 14]

3.2 Permissions for Business Rules

You can only work with business rules if you make a few configuration settings and if you have the following permissions.

Prerequisites

Permissions

- You have the *Administrator Permissions* > *Metadata Framework* > *Configure Business Rules* permission.

Securing the Rule Object

Under *Configure Object Definitions* search for **`Object definition`** and **`Rule`**. Select *Take Action* > *Make Correction*. Then do the following:

- In the *Security* section, set *Secured* to *Yes*. The default setting is *No*.
- Set the *Permission Category* to *Business Rules Object Permissions*. If the *Permission Category* is set to *No Selection*, the rule is placed under the *Miscellaneous Permissions* permission category.
- *CREATE Respects Target Criteria* must be set to *Yes*.
  By defining target criteria, customers can now separate rule authorizations according to application areas. For example, Time Management admins, Recruiting admins, and so forth. Refer to **Restricting Data Access of a Role with Target Population or Criteria** in the Related Information section.

Configuration Settings for Granting Permissions

For a user to be able to view or edit rules, assign the following permissions to the permission role for the rule object.

Permission Action	View Current	View History	Create	Insert	Correct	Delete
View Rules	x	x	n/a	n/a	n/a	n/a
Edit Rules	x	x	x	x	x	x

> ⚠ Caution
>
> Permission actions for rule objects behave differently than for other MDF objects. To view or edit a business rules, the user needs every necessary permission action listed in the respective table row. For example, a user is not able to change a business rule if they have only the Correct permission. The user needs every permission action (View Current, View History, Create, Insert, Correct, Delete).

> ⓘ Note
>
> In the current version, Recruiting doesn't support the enhanced authorization check when assigning a rule.

> ⓘ Note
>
> Field-level overrides aren't respected.

Actions and Required Permissions

Desired Action	Required Permission
Create MessageDefinition objects for raising messages	*Manage Data*
Create Look up tables	*Configure Object Definitions*
Secure the Rule Object	*Configure Object Definitions*
Access the log that shows how business rules run	*Access to Business Rule Execution Log* > *Configure* Since the log can contain potentially sensitive user data, we recommend you assign the view permission only where necessary.
Create the rule trace that logs the business rule execution	*Access to Business Rule Execution Log* > *Configure*
Create the rule trace and view the resulting log	*Access to Business Rule Execution Log* > *Including downloading the log*
Import and export data for generic objects or business rules using the *Import and Export Data* link in the Admin Center	*Import Permission on Metadata Framework*
Create Sequence objects for defining sequences	*Manage Sequence* ⓘ Note Both *MessageDefinition* objects and *Sequence* objects are Metadata Framework objects.

Parent topic: Implementation of Business Rules [page 13]

Previous task: Setting Up Provisioning [page 13]

Next task: Setting Up the Application [page 16]

Related Information

List of Role-Based Permissions
Restricting Data Access of a Role with Target Population or Criteria

3.3 Setting Up the Application

You need to enable and set up the application for which you want to create business rules.

Procedure

Set up the application as described in the corresponding implementation guide.

> ⓘ Note
>
> Please note that each application might use business rules in a different way, so make sure you understand the application-specific aspects as described in the application-specific documentation.

Task overview: Implementation of Business Rules [page 13]

Previous: Permissions for Business Rules [page 14]

4 Business Rules Admin Page

The *Business Rules Admin* page is the central location where you can view and manage the business rules in your company.

The page is located under *Admin Center* > *Configure Business Rules*. All business rules in your system are listed here.

On top of the page is the header section with a search box and a list of default filters. Below the header section is the list of business rules.

You can search, filter, sort, and group business rules on this page, as well as see the assignment status of a business rule. You can also view the details of a business rule, and create any new business rules as necessary. When you select a rule you're taken to the *Configure Business Rules* page. There you can perform further actions such as modifying and copying rules, and deleting rule records.

5 Creating Business Rules

Create business rules to configure system behavior.

Prerequisites

- You've enabled *Enable Generic Objects* in Provisioning.

  > → Remember
  >
  > As a customer, you don't have access to Provisioning. To complete tasks in Provisioning, contact your implementation partner or Account Executive. For any non-implementation tasks, contact Technical Support.

- You've assigned the corresponding permissions under *Administrator Permissions* > *Metadata Framework*. For more information, please refer to Introduction to Using RBP.
- You've set up the application for which you want to create business rules as described in the application-specific documentation.

  > ⓘ Note
  >
  > Each application might use business rules in a different way, so make sure you understand the application-specific aspects as described in the application-specific documentation.

Procedure

1. Go to *Admin Center* > *Configure Business Rules*.

   You get to the *Business Rules Admin* page.
2. Choose + *(Create New Rule)*.
3. Select a scenario suitable for the business rule you want to create.

   In the top-right corner, the system displays user entry fields for required information about the rule you want to create.
4. Make the required entries for the rule and choose *Continue*.

   You get to the *Configure Business Rules* page.
5. **Optional:** Add variables in the *Variables* section.
6. Create conditions in the *If* section.

> ⓘ Note
>
> Move the cursor to the upper right corner of the *If* section and select *Always True* if the rule is to be triggered when any change is made to the object or entity the rule is assigned to.

7. Create statements in the *Then* section to define how the system reacts when the conditions are met.

> ⓘ Note
>
> Some rule scenarios don't require a Then statement if the application has already defined the system's reaction.
>
> For more information, please refer to the application-specific documentation.

8. **Optional:** Add Else If conditions if you want to combine several conditions in the same rule.

> → Tip
>
> You can copy and paste the If conditions to the Else If conditions if they share similar expressions.

9. **Optional:** Add Else statements to define how the system reacts when none of the conditions are true.

> → Tip
>
> You can copy and paste the Then statements to the Else statements if they share similar expressions.

10. Save the rule.

Next Steps

To make the rule effective, you must assign it to the corresponding target object or entity.

Related Information

Creating Conditions [page 20]
Creating Statements [page 22]
Assigning Business Rules [page 55]
Creating Variables [page 24]

5.1 Creating Conditions

Define which conditions have to be met in the If or Else If section of the business rule so that the system executes the actions defined in the Then statement.

Prerequisites

You've created a business rule and are on the *Configure Business Rules* page in edit mode.

Context

In some cases, you want the statements of a rule to be executed when any change is made to the object or entity the rule is assigned to. To achieve this behavior, you set the conditions to *Always True*.

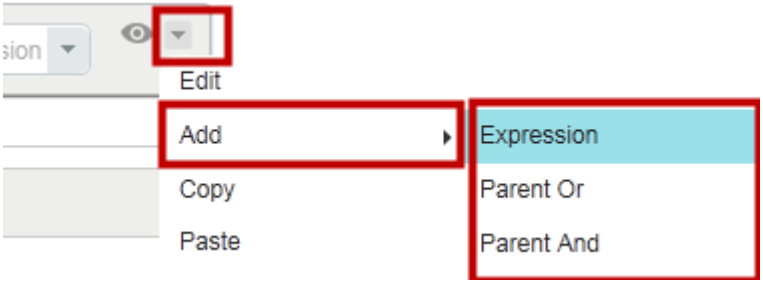
You can use this option, for example, to default values or set field properties that are displayed to the user as soon as the user opens a page. If you can use this option at all is defined by the rule scenario you've chosen.

Procedure

1. On the *Configure Business Rules* page, go to the *If* section of the rule and choose one of the following options:
   - To define that the statements of the rule are executed when any change is made to the object, field, or entity the rule is assigned to, move the cursor to the upper right corner of the *If* section and select *Always True*.
     With *Always True* selected, no more input is required in the *If* section and you can proceed to create the *Then* statement.
   - To create conditions, open the dropdown list of the left expression and choose the corresponding value.
2. Compare the left and right expression by selecting the corresponding comparison operator.

   For example, *is equal to*.
3. Select the corresponding right expression.
4. **Optional:** To add additional If conditions, open the dropdown menu of the If condition, select *Add*, and select the corresponding logical operator, which connects the If conditions.

   For example, you can add a parent If condition that connects with an And logical operator.



> → Tip
>
> From the dropdown menu, you can also copy and paste the If conditions to another If condition if they share similar expressions.

5. **Optional:** To move the position of conditions you've already created, open the dropdown menu of the corresponding If condition and select *Move*.
6. **Optional:** To add Else If conditions that are considered if the If conditions you've defined so far aren't met, select *Add Else If* at the bottom of the rule.

   Else If conditions allow you to combine several conditions in the same rule. Only the Then statements following the first If condition that is true are executed. All other Then statements are skipped.

Next Steps

1. If applicable, proceed with defining the Then statement for your business rule.
2. Assign the business rule to the corresponding object or entity.

Related Information

Comparing Values Using Left and Right Expressions [page 27]
Creating More Complex Rules Using Rule Functions [page 33]
Collection Filters [page 35]
Comparison Operators [page 32]

5.1.1 Logical Operators for Connecting Conditions

And or *Or* logical operators define the relationship between the conditions in the *If* or *Else If* section of a rule.

Logical Operators for Connecting Conditions

Option for Adding Logical Operators	Result
Expression > *And*	Joins conditions, using *And* as logical operator.
Expression > *Or*	Joins conditions, using *Or* as logical operator.
Expression > *Insert Before*	Adds a new condition before the currently selected condition.
Expression > *Insert After*	Adds a new condition after the currently selected condition.
Parent Or	Adds a parent *Or* logical operator.
Parent And	Adds a parent *And* logical operator.

5.2 Creating Statements

Define how the system reacts if the conditions of a business rule are met. For example, an error message is raised, a field is set to a specific value, or new data is created.

Prerequisites

You've created a business rule and are on the *Configure Business Rules* page in edit mode.

Context

Some rules don't require a Then statement if the application has already defined the system's reaction. For example, in eligibility rules for variable pay, you want to filter the employees that are eligible to take part in a variable pay program or bonus plan. Such a rule consists only of If conditions as the variable pay application defines for which variable pay program or bonus plan the employee is eligible for.

For more information, please refer to the application-specific documentation.

Procedure

1. On the *Configure Business Rules* detail page, go to the *Then* section and select the *Edit Expression* icon.

   The entry field for selecting actions appears.
2. Open the dropdown menu to select an action.

   For example, select *Raise Message*.

   > ⓘ Note
   >
   > You see only the actions that are available for the rule scenario you've selected.
3. Fill out the required fields, or select the corresponding expressions.
4. **Optional:** To add additional Then actions, open the dropdown menu of the Then statement and select *Add Expression Above*.

   > → Tip
   >
   > From the dropdown menu, you can copy and paste the Then statement to another Then statement if they share similar expressions.
5. **Optional:** To change the order of Then statements you've already created, select *Move* from the dropdown menu of the corresponding Then statement.
6. **Optional:** Add an Else statement to define how the system reacts if the conditions of the rule are **not** true.

If the conditions defined in the If sections aren't met, the Then statements are skipped and the system executes what is defined in the Else statement.

7. Save your changes.

Next Steps

Assign the business rule to the corresponding object or entity.

Related Information

Actions for Then and Else Statements [page 23]
Assigning Business Rules [page 55]
"Execute" Functions [page 103]

5.2.1 Actions for Then and Else Statements

In the Then and Else statement, you define how the system should react when the conditions of a business rule are met.

Here's an overview of the available actions. Please note that the scenario defines which of these actions are available.

Overview of Actions

Action	Result
Set	Changes the data record for a field to a specific value. For example, you can set the manager to *Carla Grant (cgrant)*. **Note** As of Q2 2019 release, the system is able to detect rules that set values to system-defined read-only fields. If you open such a rule, an error message appears on the read-only field. We recommend that you modify the rule definition to avoid setting unwanted values to read-only fields. This only applies to system defined read-only fields, such as MDF Auto Number type fields. MDF fields that you define as read-only in Configure Object Definition are not affected. **Note** As right expression (source field) you can only select values that have the same data type as the left expression (target field).
Raise Message	Raises an error message that is displayed to the user.

Action	Result
Create	Creates data on the database when the rule is triggered. For example, you can define that when the employee's location changes to London, a new pay component is created.
Delete	Deletes data from the database when a rule is triggered. For example, you can remove a pay component when the employee moves away from London. **Note** If the object selected for deletion is a collection, you need to specify a collection filter. During rule execution, the system deletes all entries that meet the collection filter criteria.
Execute	Carries out a specified action when the rule is triggered. Refer to the list of "Execute" functions to find out which actions are available.
Add to	Adds items to a collection when a rule is triggered. For example, you can create a single rule to assign multiple learning courses to new hires.

Related Information

Creation of a Rule That Raises a Message [page 38]
Creating a Rule That Adds Multiple Items to a Collection [page 43]
"Execute" Functions [page 103]

5.3 Creating Variables

Simplify business rules by storing calculation results of complex and repeating tasks in variables.

Prerequisites

You've created a business rule and are on the *Configure Business Rules* page in edit mode.

Context

The *Variables* section of a business rule serves as a processing block before the first *If/Else If/Then/Else* section in which it is first used. You can store calculation results of complex or repeating tasks in variables and call them

in the If/Else If/Then/Else sections of the rule. In doing so, you can simplify the rule definition and improve the performance of rule executions. You can define any number of variables.

Procedure

1. On the *Configure Business Rules* page, choose the *Variables* section, and choose *Add Variable*.
2. Enter the name of the variable. The name starts with the prefix "var_" and can contain only alphanumeric characters and underscores.
3. In the *Select Expression* field, define the complex or repeating tasks using expressions. You can choose the eye icon to switch to view mode.

Results

You can now use the variable in the If conditions and Then statements by selecting it from the dropdown list of the corresponding field where you want to use it.

> **Note**
>
> You can use variables as a read-only element in the rule definition. You can't create, edit, or delete values of variables.

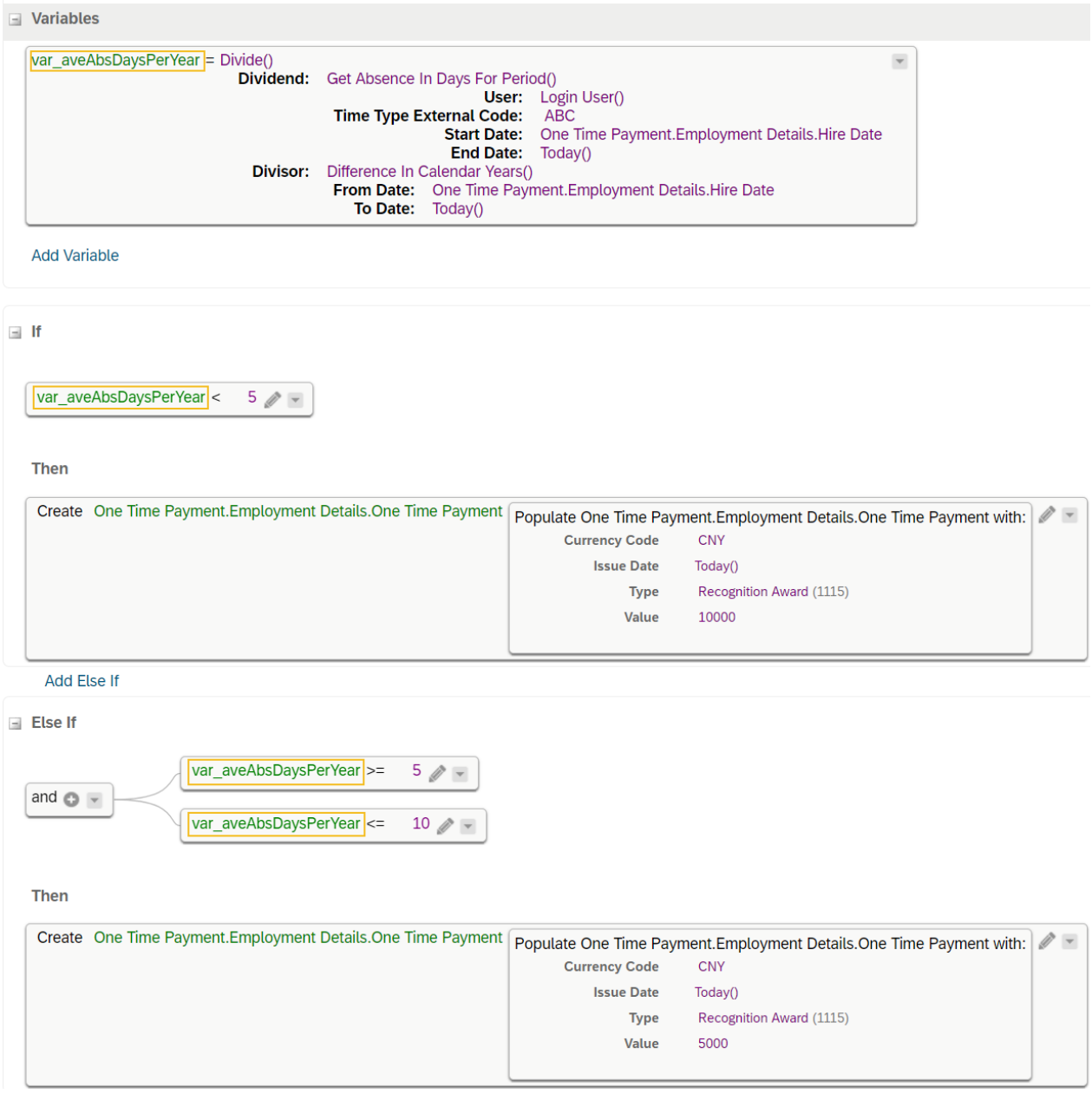
Example

You want to calculate the average annual absence days of employees and use this number to determine the amount of payment given to them as service awards. Instead of defining a rule that calculates the average absence days in each If condition, you can create a variable before the rule definition and use it in each If condition.

In the *Variables* section, you select the Divide() function and make the following entries:

Input Parameter	User Entry	Input Parameter	User Entry
Dividend	Get Absence in Days For Period()	*User*	Login User()
		Time Type External Code	ABC
		Start Date	One Time Payment.Employment Details.Hire Date
		End Date	Today()
Divisor	Difference in Calendar Years()	*From Date*	One Time Payment.Employment Details.Hire Date
		To Date	Today()

In the *If* section, you define that if the variable returns a result smaller than 5, then a one-time payment of type Recognition Award is created. Here's an example of what this rule could look like in the system:



5.4 Comparing Values Using Left and Right Expressions

Define rule statements following the logic "A is equal to B", with A being the left expression, and B being the right expression, connected by the comparison operator "is equal to".

Prerequisites

You've created a business rule and are on the *Configure Business Rules* page in edit mode.

Context

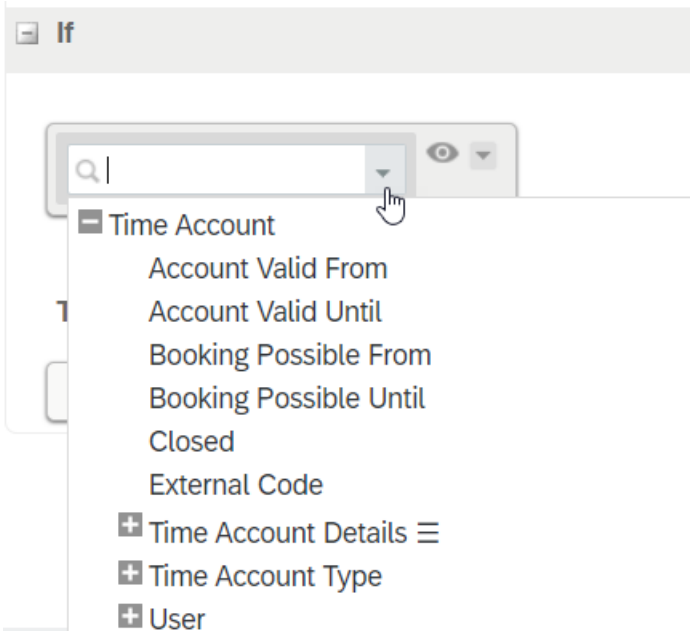
Left and right expressions are available in the following parts of a business rule:

- *If* section
- *Then* section for the *Set* action

In general, the dropdown list for the right expression contains the same values as the left expression, but is limited to the entities that have the same field type as the left expression. For functions, the return type needs to be the same as the field type of the left expression. In addition, there are some options in the dropdown list specific to the right expression.

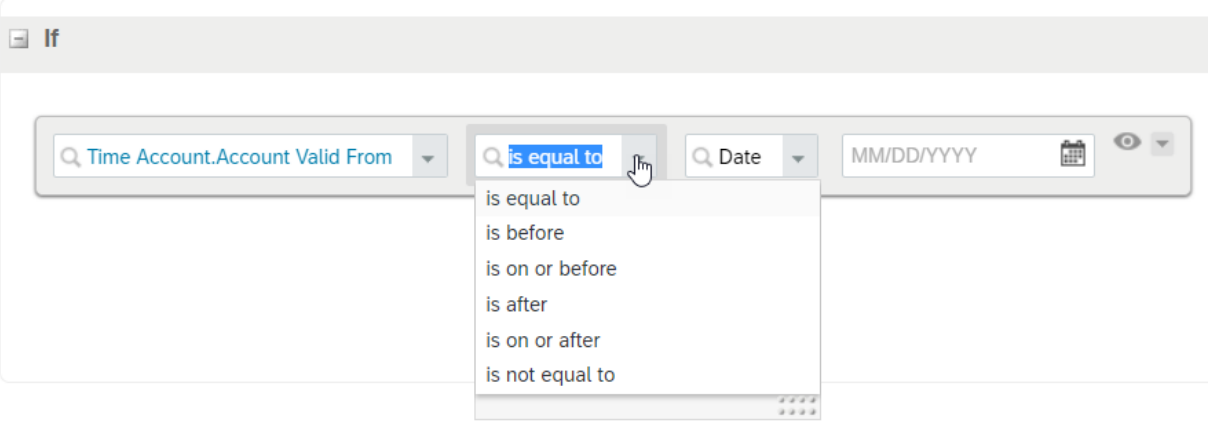
Procedure

1. On the *Configure Business Rule* page, open the dropdown list of the left expression and select the corresponding value.



2. Open the dropdown list for the comparison operator that logically connects the left and right expression, and select one.

   For example, *is equal to*.



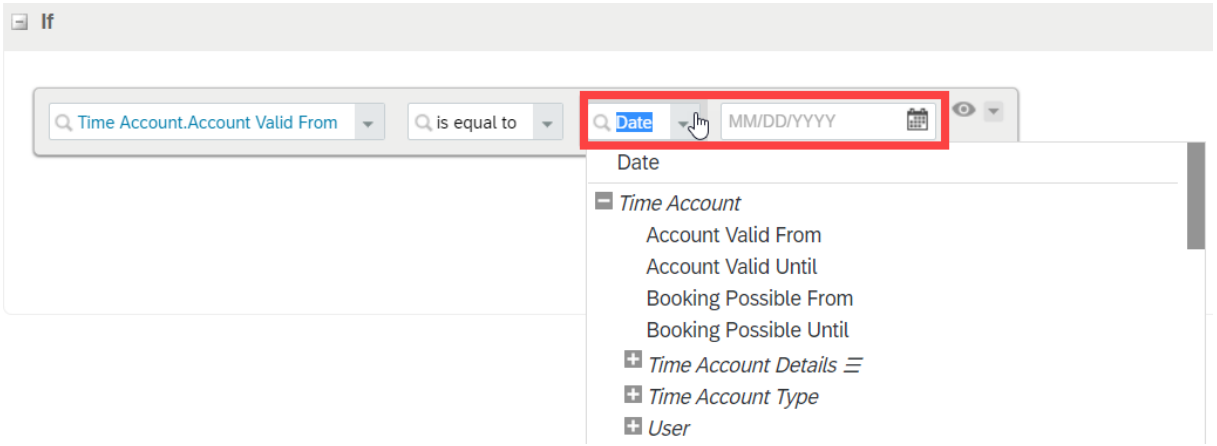
3. For the right expression, you have the following options:

   - Select a specific value. This specific value can be either a value you enter yourself (for example, a number), or a value predefined by the system you can select from a dropdown list (for example, *Yes* or *No* for a Boolean field). In this case, the system uses the fix value you've chosen during runtime.
   - Select a dynamic value. For example, an object field or a function. In this case, the system retrieves the value dynamically at runtime from the selected object field or function.
   - Select *Null* to set field values back to null.

     > **Note**
     >
     > Please note that "null" doesn't mean the number zero in the context of business rules. Instead, it represents a missing value. A field with a null value is a field with no value.

> **Note**
>
> The values in the left and right expression must have the same field type. That's why what you've defined in the left expression controls the fields and functions you can select in the right expression.

In this example, you can select fields of type *Date* in the dropdown list of the right expression to get a dynamic value for the date field. To use a specific value, you can enter a specific date in the user entry field next to the dropdown list:



Next Steps

Continue with setting up your business rule, and save your changes.

Values for Left Expressions [page 29]
Overview of the possible values you can select from the dropdown lists in a business rule for the left expression.

Additional Values for Right Expressions [page 31]
Overview of the additional values you can select from the dropdown list in a business rule for the right expression.

Comparison Operators [page 32]
Comparison operators are used to define the relationship between values in the left and the right expression of a business rule.

5.4.1 Values for Left Expressions

Overview of the possible values you can select from the dropdown lists in a business rule for the left expression.

> **Note**
>
> The values you can select in the dropdown list are limited by the scenario you've chosen and the selections you've made in the rule. That's why you might not see all the possible values indicated in this table in the system.

Values for Left Expressions

Value in Dropdown List	Example	Details
Objects and fields	*Time Account Details*	These are objects and fields related to the rule's parameter. For rules using the basic rule scenario, you can add additional objects as parameters to the rule. Please refer to the application-specific documentation first to understand which parameters are supported. **Note** You can select only fields that you've defined as visible.
Context > *Current User*	*Current User*	*Current User* is only available for rule scenarios that explicitly support this option. The information about the current user is derived from the base object. In the *Basic* rule scenario, this option is displayed but at runtime it is run for Employee Central objects only. **→ Remember** We don't recommend that you use the context.user path to navigate to an Employee Central object if the Employee Central objects are available as rule parameters. Instead use the path from the rule parameter itself. The previous value is available in model-based rules only. **Example** CompInfo is the base object. To navigate to `jobInfo`, do not use `context.user.jobInfo` but instead use `compInfo.employmentDetails.jobInfo`.
Context > *Effective Date*	*Effective Date*	This is the effective date passed for the rule execution.
Variables	var_aveAbsDaysPerYear	If you've created variables, they're listed in the dropdown list.

Value in Dropdown List	Example	Details
Functions	*Multiply()* *Create Date()*	Functions are used to set up more sophisticated rules that include mathematical calculations and formulas, for example. You can identify functions by the brackets that follow the function name; for example, *Add()*. You can find a list of available functions under *Functions A-Z* and *Functions By Groups*.

Parent topic: Comparing Values Using Left and Right Expressions [page 27]

Related Information

Additional Values for Right Expressions [page 31]
Comparison Operators [page 32]

5.4.2 Additional Values for Right Expressions

Overview of the additional values you can select from the dropdown list in a business rule for the right expression.

> ⓘ Note
>
> The values you can select in the dropdown list are controlled by the scenario you've chosen and the selections you've made in the rule for the left expression. That's why you might not see all the possible values indicated in this table in the system.

In addition to what you can choose from in the left expression, you can select the following options listed below for the right expression.

Values Specific to the Right Expression

Value in Dropdown List	Example	Details
Field type	*Decimal* *Boolean*	When the field type is selected, you can enter or select a specific value that doesn't change during runtime.

Value in Dropdown List	Example	Details
Null	*Null*	*Null* sets field values back to null. There is a limitation for functions: "Null" is only available for optional function input parameters, for which you can set the field value to null by selecting *Null* or leaving the field empty. **Note** Please note that "null" doesn't mean the number zero in the context of business rules. Instead, it represents a missing value. A field with a null value is a field with no value.

Parent topic: Comparing Values Using Left and Right Expressions [page 27]

Related Information

Values for Left Expressions [page 29]
Comparison Operators [page 32]

5.4.3 Comparison Operators

Comparison operators are used to define the relationship between values in the left and the right expression of a business rule.

The following comparison operators are supported:

Comparison Operator	Field Types Supported
Is equal to	All field types
Is not equal to	All field types
Is greater than: >	Number, Decimal
Is less than: <	Number, Decimal
Is greater than or equal: >=	Number, Decimal
Is less than or equal: <=	Number, Decimal
Is before	Date
Is on or before	Date

Comparison Operator	Field Types Supported
Is after	Date
Is on or after	Date

Parent topic: Comparing Values Using Left and Right Expressions [page 27]

Related Information

Values for Left Expressions [page 29]
Additional Values for Right Expressions [page 31]

5.5 Creating More Complex Rules Using Rule Functions

Use functions to perform specific tasks on the object or field of a rule. For example, you can perform calculations, application-specific tasks or date/time-related tasks.

Prerequisites

You've created a business rule and are on the *Configure Business Rules* page in edit mode.

Context

Functions are an optional part of a rule and can be used in any of the rule sections.

Procedure

1. On the *Configure Business Rules* page, open the dropdown list of the field to which you want to add the function.

   In general, functions are available in the following parts of a business rule:
   - If and Else If conditions
   - Then and Else statements

> **Note**
>
> "Execute" functions are only available for the *Execute* action of a Then statement.

- Variables

2. In the dropdown list, scroll down to the list of functions, which appears at the bottom of the dropdown list, and select a function.

   You can identify functions by the round brackets that follow the function name; for example, *Add()*. When you've selected the *Execute* action, only "Execute" functions are listed in the dropdown list.

   > **Note**
   >
   > The functions available in the right expression are restricted by the field type of the field you selected. For example, if the field is of type Boolean, you can only use a function that returns a Boolean value.

   If the function requires input parameters, these are displayed below the function name.

3. For each required input parameter, select or enter the corresponding value. You can use the same values as listed for right expressions.

   There is a limitation for functions: "Null" is only available for optional function input parameters, for which you can set the field value to null by selecting *Null* or leaving the field empty.

   > **Note**
   >
   > The function defines the field type of its input parameters, and with that it restricts what you can enter or select as input parameter. You can find more information on the supported input for a specific function under *Functions A-Z*.

Next Steps

Continue setting up the rule and save your entries.

Related Information

Functions A-Z [page 106]
Additional Values for Right Expressions [page 31]
"Execute" Functions [page 103]

5.6 Collection Filters

Collection filters are used to get a unique value from a list of values. You need to define a collection filter for fields that contain collections.

> **Note**
>
> In a collection filter, you can only select literal values or "null" for comparisons. If you need to reference an object, you have to create a variable first, and then reference that variable in the collection filter.

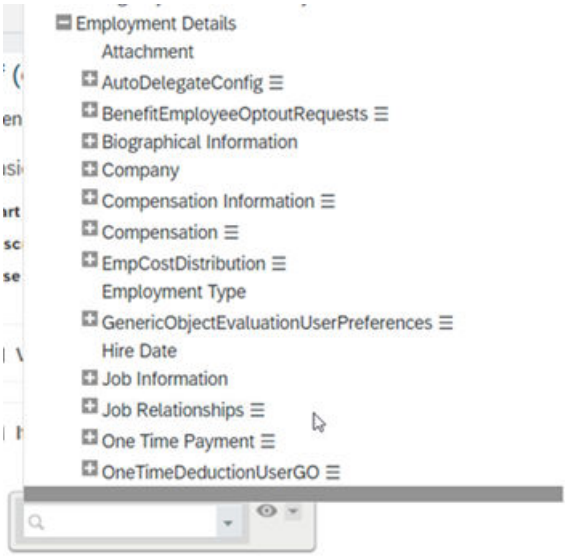
Understanding Collection Filters

As a collection can contain more than one entry, you need to define which entry the rule should use.

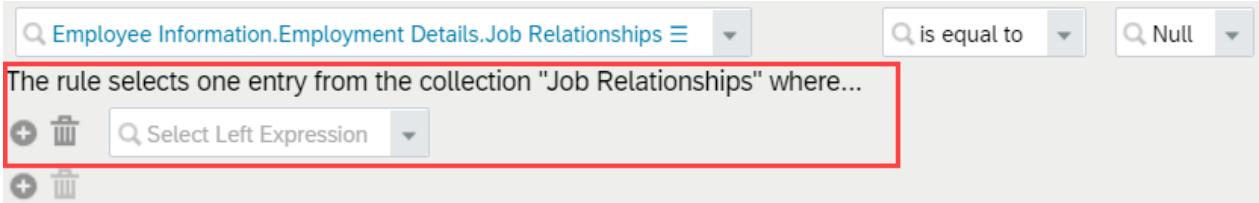
Let's look at the example of the object Job Relationships in Employee Central. Job Relationships is the child object of Employment Details. You can have multiple job relationships for a single employment. The collection filter lets you define which job relationship to use in the rule - for example, HR Manager. If you don't define which job relationship you want to use for your rule, the system picks one job relationship by itself when the rule is executed, and you can get other results from the rule execution than expected.

Collection Filters on the User Interface

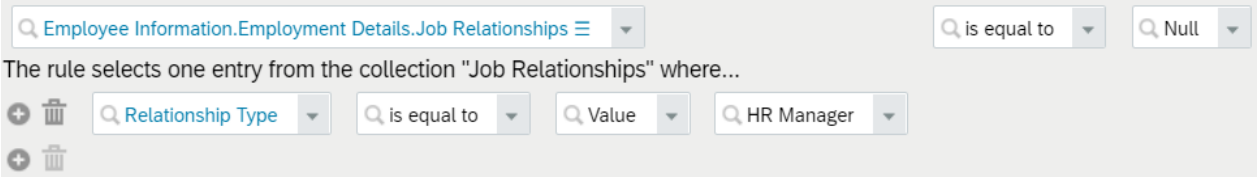
When you're editing a business rule on the *Configure Business Rules* page, you get a list of the available objects for that rule when you open the dropdown list of an entry field. Fields that contain collections have an icon next to them.



When you select the corresponding parent-child object, the collection filter is displayed under the object field, where you need to define which value you want to use in the rule. In this example, you need to define which job relationship you want to use.



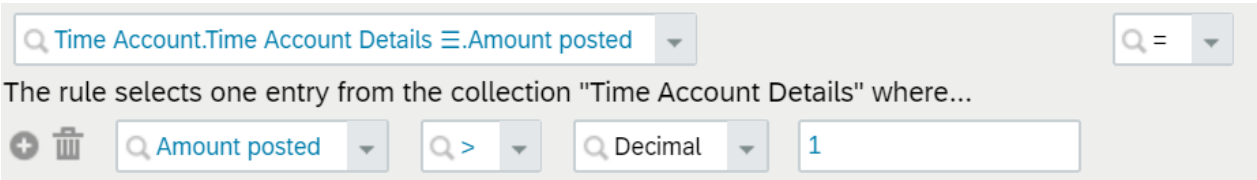
In this example, you chose to use the relationship type HR Manager in the rule.



Correct Definition of Collection Filters

If you're too unspecific in the collection filter, the system returns a list with more than one result. The system always picks the first result from this list. You can't control which result the system defines as the first result.

Therefore, when creating the rule, make sure the collection filter returns only **one** value. In the following example, you can see a collection filter that returns a list with **many** values, because you ask for any amount posted that is larger than 1. As probably most of the posted amounts are larger than 1, this collection filter returns more than one result. Only the first result of the returned list is picked for the rule, and the user can't control which result is the first result. The results following the first result are ignored. Here's the example of how **not** to do it:



5.7 Parameters

The parameters of a business rule limit which fields, attributes, and related objects you can use to define the rule.

The parameters correspond mainly to the objects available in the system.

You can see the parameters of a rule listed on the rule detail screen of the *Configure Business Rules* page. They comprise the following objects:

- System context
  A read-only attribute that every rule requires.

- Base object
  The first parameter of a rule. The rule scenario defines the sequence of the rule parameters. The base object often corresponds to the object where the rule is assigned to and triggered.
- Additional objects
  Some rule scenarios require additional objects. You can find more information in the application-specific documentation.

> **Note**
>
> Parameters that are read-only are marked with a lock symbol. Read-only parameters can't be used in the left expression of a *Set*, *Create*, or *Delete* action.

Base Objects [page 37]
The base object is the first parameter of a rule, in which it's the rule scenario that defines the sequence of the rule parameters. The base object often corresponds to the object where the rule is assigned to and triggered.

5.7.1 Base Objects

The base object is the first parameter of a rule, in which it's the rule scenario that defines the sequence of the rule parameters. The base object often corresponds to the object where the rule is assigned to and triggered.

For application-specific scenarios, the scenario defines the base object, or limits which base objects you can choose when you're creating a rule.

5.8 Examples for Creating Business Rules

Find some examples for creating business rules with specific purposes.

Creation of a Rule That Raises a Message [page 38]
A rule that raises a message when the rule is triggered is defined in a two-step process.

Creating a Rule That Adds Multiple Items to a Collection [page 43]
Create a business rule that automatically adds multiple items to a collection using the *Add To* action in the Then or Else statement.

Creation of a Rule with Lookup Function [page 44]
The Lookup function is used to look up values from a table. The table must be a Metadata Framework object.

Creating Effective-Dated Rules [page 48]
Create effective-dated rules to get rules that change depending on when they're triggered.

Creating a Sequence Object [page 49]
Create a Sequence object to define where the counting for the next value of a number or code starts, and the step width used to get the following value.

Creating a Rule That Validates the Format of User Entries [page 50]

Define the format in which users are allowed to enter information on the user interface using the Matches function. For example, you can define that only digits are allowed in the phone number field, or that special characters can't be entered.

5.8.1 Creation of a Rule That Raises a Message

A rule that raises a message when the rule is triggered is defined in a two-step process.

A typical use case for creating such rules is to validate the user's entries. For example, you could create a rule that displays an error message when the person's age is less than 18.

To create such a rule, you first create the message text, and then create a rule that refers to the message text you've created.

1. Creating a Message Text [page 38]
   Create the message text to be displayed when a rule is triggered by creating a Message Definition object.
2. Creating a Message Rule [page 40]
   Define a rule that displays a message when it's triggered.

5.8.1.1 Creating a Message Text

Create the message text to be displayed when a rule is triggered by creating a Message Definition object.

Context

The Message Definition object is a Metadata Framework object that contains the message text.

> **Note**
>
> You can't expose the Message Definition object to APIs.

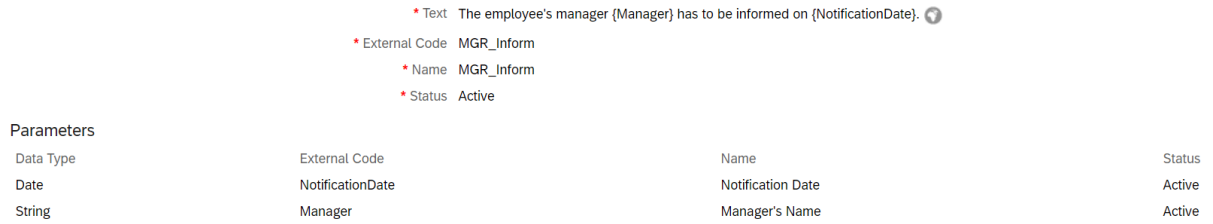
Procedure

1. Go to *Admin Center* > *Manage Data*.
2. In the *Create New* field, select *Message Definition*.
3. In the *Text* field, enter the message text that you want to be displayed to the user. If you use parameters in the message text, place the external code of the parameters you're going to define in step 6 inside braces.

   > **Note**
   >
   > Parameters are like placeholders that are filled with dynamic values when the message is called up.

This Message Definition object contains the message text "The employee's manager {Manager} has to be informed on {NotificationDate}", where the corresponding manager's name and the date are filled during runtime.

Message Definition: MGR_Inform (MGR_Inform)



4. Enter an external code for the Message Definition object.
5. Enter a name for the Message Definition object.
6. If you've used parameters in the message text in step 3, define their properties under *Parameters*.

> ### Note
>
> In this step, you only define what type of parameters you want to use in this message (for example: Date as data type). The concrete parameters are defined when you assign the message to the rule (for example: Position Entry Date).

7. Save the message.

Parameter Definition for Message Definition Objects [page 40]

If you use parameters in the message text of a Message Definition object, you need to define their properties on the *Manage Data* page.

Task overview: Creation of a Rule That Raises a Message [page 38]

Next task: Creating a Message Rule [page 40]

5.8.1.1.1 Parameter Definition for Message Definition Objects

If you use parameters in the message text of a Message Definition object, you need to define their properties on the *Manage Data* page.

Parameter Definition

Properties	Description
Data Type	The data type of the parameter. For example, you select *Date* for a date parameter, *String* for a text entry, and so on. **Note** If you want to use the Message Definition object in the *Raise Message* action of the business rule, you can only use the following types: • String • Number • Decimal • Boolean • Date • DateTime • Time
External code	The external code for the parameter that you reference in the message text of the Message Definition object.
Name	The parameter's field label that is displayed on the *Configure Business Rules* page when you assign the message to a rule.
Status	The parameters are used when the rule is called up when their status is *Active*.

5.8.1.2 Creating a Message Rule

Define a rule that displays a message when it's triggered.

Prerequisites

You've created a Message Definition object that contains the message text to be displayed.

Procedure

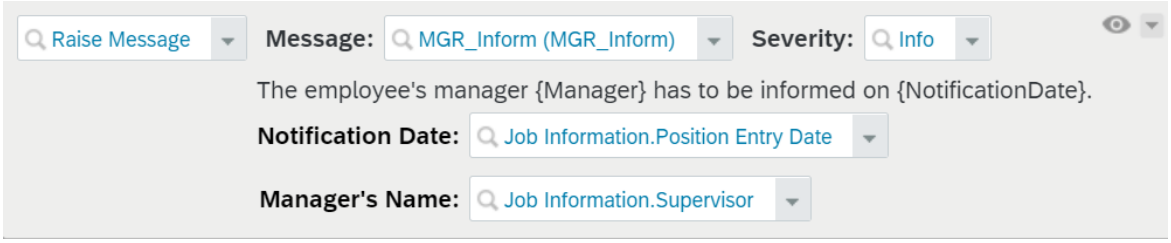
1. Go to *Admin Center* > *Configure Business Rules*.

   You get to the *Business Rules Admin* page.
2. Choose + *(Create New Rule)*.
3. Select a scenario suitable for the business rule you want to create.

   In the top-right corner, the system displays user entry fields for required information about the rule you want to create.
4. Make the required entries for the rule and choose *Continue*.

   You get to the *Configure Business Rules* page.
5. Enter the conditions that cause the error message to pop up in the If conditions.
6. Define the Then statement where you refer to the message. Select the following:
   a. In the *Select Action* field, select *Raise Message*.
   b. As *Message*, select the Message Definition object that you've created before.
   c. If you've defined parameters for the message when you created the Message Definition object, enter or select the concrete parameters that you want to be used in this rule.

      Here's an example, where the date is the position entry date, and the manager is derived from the Supervisor object field. The message text has been defined as "The employee's manager {Manager} has to be informed on {NotificationDate}."

   d. As *Severity*, select whether the message is an error, warning, or info message.

      > **Note**
      >
      > You can define that messages with several severities are raised simultaneously for the same object. However, if you do this, only one dialog is shown containing all the messages, and the strictest severity is used to determine the dialog type. For example, if there's one *Info* and one *Warning* message raised at the same time, the message is shown as a *Warning* dialog box, containing both messages.
7. Save the rule.

   > **Note**
   >
   > When you save the rule, an association to the used message definition is added to the rule. With that, the associated message definition is considered when the rule is exported as well as in case the rule is added to a transport bundle and transported using the Configuration Transport Center.

Next Steps

Assign the rule to the corresponding object or entity.

> **Note**
>
> Once a message definition is used in a business rule, it is no longer possible to delete the message definition, or to change the external code of the message definition.
>
> If you want to do either of those things, you first need to remove the message definition from any business rules.

Message Severity [page 42]
: The severity defines whether a message is an error, warning, or info message.

Task overview: Creation of a Rule That Raises a Message [page 38]

Previous task: Creating a Message Text [page 38]

5.8.1.2.1 Message Severity

The severity defines whether a message is an error, warning, or info message.

Severity	Result
Error	The user can't save the data changes as long as the error still exists.
Warning	The user can accept the warning and save the data changes. If the user doesn't accept the warning, the data change isn't saved.
Info	The system displays an information dialog that automatically fades away after 5 seconds. The user can also dismiss it manually by choosing the OK button.

> **Note**
>
> *Error* is the strictest severity, *Info* the lowest severity.

5.8.2 Creating a Rule That Adds Multiple Items to a Collection

Create a business rule that automatically adds multiple items to a collection using the *Add To* action in the Then or Else statement.

Context

As an admin, you're asked by business to automate the process of assigning multiple items to an entity. For example, several new hires have recently onboarded in your organization. You need to find a way to assign a series of learning courses to them automatically. Instead of creating a business rule for each course, you can define one single rule and add all courses to a collection.

What is a collection?

A collection represents a field in an object to which multiple instances or items of the same data type can be assigned.

Procedure

1. Go to *Admin Center* > *Configure Business Rules*.

   You get to the *Business Rules Admin* page.
2. Choose *Create New Rule*.
3. Select a scenario.

   > **Note**
   >
   > Please note that the scenario defines whether you can add items to a collection.
4. On the *Configure Business Rules* page, define the conditions in the *If* section.
5. In the Then statement, select the *Add To* action.
6. Select the collection you wish to add entries to, and add entries to the collection.

   Collections are displayed in the dropdown list with an icon next to them.

   You can add as many items to the collection as required.
7. **Optional:** Add another expression in the Then statement and add items to a different collection. To do so, choose the expand icon on the top-right corner of the current expression and choose *Add* > *Expression Above*.
8. Save the rule.

5.8.3 Creation of a Rule with Lookup Function

The Lookup function is used to look up values from a table. The table must be a Metadata Framework object.

The benefits of using the Lookup function in a rule instead of creating various Else and Else-If statements are:

- The rule is less complex.
- A less complex rule often improves performance.
  For example, if you need multiple pieces of information from the lookup, you can wrap all the information in an object, and perform the lookup only once in the variable section. Then, all the information can be used in the rule by using the fields of the variable.
- The lookup table can be reused in multiple rules.

To use the lookup function in a business rule, execute the following steps.

> **Note**
>
> If the Metadata Framework object already exists and its instances are set up as required, you can skip the first two steps.

1. Creating a Lookup Table [page 44]
   Create a lookup table as a Metadata Framework object under *Configure Object Definitions* to use the Lookup function in a rule.
2. Defining Value Combinations for Lookup Tables [page 45]
   Define the value combinations for a lookup table under *Manage Data*.
3. Creating a Rule with Lookup Function [page 47]
   Create a rule under *Configure Business Rules* that uses lookup tables to consider additional values in a rule.

5.8.3.1 Creating a Lookup Table

Create a lookup table as a Metadata Framework object under *Configure Object Definitions* to use the Lookup function in a rule.

Prerequisites

You have the *Administrator Permissions* > *Metadata Framework* > *Configure Object Definitions* permission.

Procedure

1. Go to *Admin Center* > *Configure Object Definitions*.
2. Select *Create New: Object Definition*.

3. Enter a code and label for the lookup table.
4. Make sure that the status of the object is *Active*.

   Inactive objects are ignored.
5. In the *Fields* section, create custom fields for the different condition values that should be part of your lookup table.

   You create one field per condition.
6. If you want to reference an existing Metadata Framework object in your lookup, select *Details* to add details for the fields you've just created.
   a. As *Data Type*, enter the data type of the Metadata Framework object you want to reference.
   b. As *Valid Values Source*, enter the external code of the Metadata Framework object you want to reference.

   If you need multiple information from the lookup table in the rule, we recommend to use a parent-child relationship. In the rule, you can fill a variable by using the lookup and returning the child object. Then, the fields of the variable can be used in the rule.

   Another advantage of such a parent-child relationship is that - if you use a 1:n parent-child relationship - you have a table-like user interface to enter the values.
7. Save your changes.

Next Steps

Define the combination of values of the custom fields you've just created under *Manage Data*.

Task overview: Creation of a Rule with Lookup Function [page 44]

Next task: Defining Value Combinations for Lookup Tables [page 45]

5.8.3.2 Defining Value Combinations for Lookup Tables

Define the value combinations for a lookup table under *Manage Data*.

Prerequisites

- You've created a lookup table as Metadata Framework object. For more information, please refer to Creating a Lookup Table [page 44].
- You have the *Administrator Permissions* > *Metadata Framework* > *Manage Data* permission.

Procedure

1. Go to *Admin Center* > *Manage Data*.
2. Under *Create New*, select the Metadata Framework object you've created as lookup table.
3. Enter one of the value combinations you want to allow.
4. Save your changes.
5. Repeat the previous steps for all value combinations.

> **Note**
>
> Under *Manage Data*, you create one instance for every value combination of your lookup table.
>
> However, if you use a 1:n parent-child relationship, you can enter all entries for one parent. The advantage of this approach is that you have a table-like user interface, and don't need to repeat the previous steps for each value combination. Furthermore, you can use multiple pieces of information from the lookup child object with only one access to the database. To achieve this, you use the lookup on the child object to fill a variable, and then use the fields of the variable in the rule.

Next Steps

You can now use the lookup table in a business rule.

Task overview: Creation of a Rule with Lookup Function [page 44]

Previous task: Creating a Lookup Table [page 44]

Next task: Creating a Rule with Lookup Function [page 47]

Related Information

Lookup Example: Rule to determine an entitlement amount with lookup object
Lookup Example: Using Lookup Tables
Lookup Example: Accrual Based on Seniority

5.8.3.3 Creating a Rule with Lookup Function

Create a rule under *Configure Business Rules* that uses lookup tables to consider additional values in a rule.

Prerequisites

1. You've created a custom Metadata Framework object that serves as a lookup table under *Admin Center* > *Configure Object Definitions*.
2. You've defined the desired value combination for the lookup table under *Admin Center* > *Manage Data*.

Procedure

1. Go to *Admin Center* > *Configure Business Rules*.

   You get to the *Business Rules Admin* page.
2. Choose + *(Create New Rule)*.
3. Select a scenario suitable for the business rule you want to create.
4. Make the required entries in the top right corner of the page, and choose *Continue*.

   You get to the *Configure Business Rules* page.
5. Add the Lookup function where you need it in your business rule. In the Lookup function, select the Metadata Framework object that serves as lookup table.

   > ⓘ Note
   >
   > - You have to use a collection filter to define which object you're looking for.
   > - In the lookup function, you can only select objects with one-to-one associations. If you need to reference an object with one-to-many associations, create a variable first, and then reference that variable in the lookup function.
   > - If the Metadata Framework object is effective-dated, you'll see a field called *Effective Date is equal to*. Here you can select one of the following options:
   >   - *System Context.Effective Date* (Recommended)
   >   The effective date of the business process which triggered the rule. For example, if the rule is triggered for an organizational reassignment, the start date of the organizational reassignment.
   >   - *Today()*
   >   The rule will take the current date. That is, the day on which the rule is run.
   >   - *As Specified in Expressions Below*
   >   You specify the effective date or period in the expressions of the Lookup collection filter. With this option, you can consider data records whose effective date deviates from the effective date of the business process, or from the day on which the rule is run.
   >   If you don't explicitly specify a date in the expressions, the system considers data records that exist throughout the whole validity period of the object.

6. Save the rule.

Next Steps

Assign the rule to the corresponding target object or entity.

Task overview: Creation of a Rule with Lookup Function [page 44]

Previous task: Defining Value Combinations for Lookup Tables [page 45]

Related Information

Collection Filters [page 35]
Lookup Example: Rule to determine an entitlement amount with lookup object
Lookup Example: Using Lookup Tables
Creating Variables [page 24]
Assigning Business Rules [page 55]
Lookup Example: Accrual Based on Seniority

5.8.4 Creating Effective-Dated Rules

Create effective-dated rules to get rules that change depending on when they're triggered.

Prerequisites

- You've created a rule with a start date as of when this rule becomes effective.
- You've ensured that any effective-dated objects referenced by the rule, such as picklists, are valid throughout the time period when the rule is used. Otherwise, the rule doesn't find the corresponding entries of the referenced object when it's run, and fails.

Context

For example, in Time Off, new accrual policies are required with the beginning of a new calendar or fiscal year. You can achieve that the system considers changes to the accrual policies by creating effective-dated rules.

Procedure

1. Go to *Admin Center* > *Configure Business Rules*.

   You get to the *Business Rules Admin* page.
2. Select the rule you want to be effective-dated.

   You get to the *Configure Business Rules* page that displays the details of the rule.
3. To insert a new record for this rule, choose *Insert New Record*.
4. Enter a start date that lies after the start date of the first rule version.
5. Adjust the remaining rule details as required.
6. Save your changes.

Results

You now have a rule that changes depending on when the rule is triggered. The system considers the date of the object where the change happens.

If a rule is triggered before the start date, the system behaves as if there was no rule assigned to the object.

5.8.5 Creating a Sequence Object

Create a Sequence object to define where the counting for the next value of a number or code starts, and the step width used to get the following value.

Prerequisites

You require the *Administrator Permissions* > *Metadata Framework* > *Manage Sequence* permission from *Admin Center* > *Manage Permission Roles*.

Context

To use the Get Next Value function in a business rule, you need to create a Sequence object first.

Procedure

1. Go to *Admin Center* > *Manage Sequence*.
2. Make the required entries.
   - *externalCode*: Enter an external code for the Sequence object that serves as unique identifier.
   - *externalName*: Enter a name or short description for the Sequence object.
   - *start*: Enter the first value of the sequence. For example: **`1000`**.
   - *step*: Enter the step width of the sequence that is used to calculate the next sequence number. For example, if *step* is 5, and *start* is **`1000`**, the number sequence would be 1000, 1005, 1010, 1015, and so on.
   - *current*: This field displays the value that will be used as the next sequence number when the Sequence object is called up. For example, if the sequence with the above values has already been called up twice, this field shows the value *1005*.
3. Save your changes.

Next Steps

You can now use the Sequence object you've created in the *Get Next Value* function of a business rule.

5.8.6 Creating a Rule That Validates the Format of User Entries

Define the format in which users are allowed to enter information on the user interface using the Matches function. For example, you can define that only digits are allowed in the phone number field, or that special characters can't be entered.

Prerequisites

You've defined the message text to be displayed as an error message when the user has entered information in a format other than the format you define. For example: "You can enter only numbers in the phone number field."

Context

You want to make sure that the information in specific fields follows a specified format. Examples include the following fields:

- Phone number
- Postal code
- Bank account number

- First and last name
- E-mail addresses

Procedure

1. Create a rule with the desired base object.
   For example:
   - *Phone Information* to define the format for the phone number field
   - *Personal Information* to define the format for the first and last name fields
2. As If condition, define the user entry field.
   In the example of defining the format for the phone number field, you could define:

   If *Phone Information.Phone Number* is not equal to *Null*
3. As If And condition, select the *Matches* function, and inside the Matches function define which field you want to be checked.
   For example:

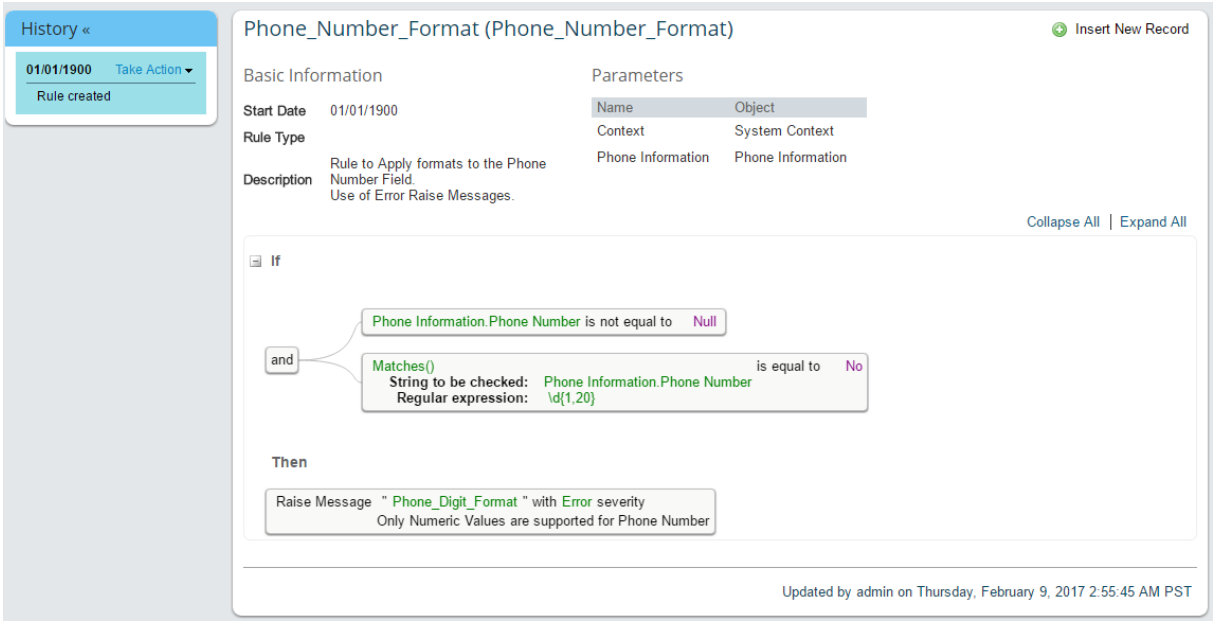
   And *Matches*

   String to be checked: *Phone Information.Phone Number*

   Is equal to *No*
4. In the *Regular expression* field of the Matches function, define the format you want to specify.

   For example, enter **`\d{1,10}`** to allow the user to enter 1–10 digits.
5. As Then statement, add the message you've created as a prerequisite.

   Your final rule could look like this example, where the user can only enter 1–20 digits as phone number:



6. Assign the rule to the appropriate object (for example, as onSave rule for the corresponding Employee Central, Metadata Framework, or Recruiting object).

For the example rule above, you assign the rule to the Employee Central object *Phone Information* under *Admin Center* > *Manage Business Configuration*.

Results

If a user enters information in a format other than the format you've defined and tries to save the changes, the system displays an error message.

Related Information

Creating a Message Text [page 38]
Examples of Java Regular Expressions [page 52]

5.8.6.1 Examples of Java Regular Expressions

Find some examples of Java regular expressions that you can use with some rule functions.

> ⓘ Note
>
> This table gives just an overview over some of the Java regular expressions you might find useful to validate user input. You can find a complete overview of Java regular expressions in the official Java documentation on the internet.

> → Recommendation
>
> As there's no syntax check of the regular expression in the system, we recommend that you test your Java regular expression in an online testing tool first.

Allowed Format for User Entry	Possible Java Regular Expressions
Only digits	• `\d` for exactly one digit • `\d+` for one or more digits • `\d*` for no digit at all, or one or more digits • `\d{1,10}` for 1-10 digits

Allowed Format for User Entry	Possible Java Regular Expressions
Only alphabetic characters	• **`[A-Za-z]`** for exactly one upper or lower case character • **`[A-Za-z]+`** for one or more upper or lower case characters • **`[A-Za-z]*`** for no character at all, or one or more upper or lower case characters • **`[A-Z]`** for upper case only • **`[A-Za-z]{1,20}`** for 1-20 alphabetic characters, upper and lower case
Special characters	• **`[!@#$]`** for special characters (in this example, exclamation mark, @-sign, hashtag, and dollar sign) **Note** • If you want to check that there are no special characters present, you need to add a circumflex **^** (symbol for negation in Java regular expressions) at the first position inside the square brackets, for example: **`[^!@#$]`** • Some special characters like square brackets, backslash, apostrophe, and hyphen (**`[]\'-`**) are considered special characters in a Java regular expression. Therefore, these characters only work in the rule if you add a backslash **** before them. For example, if you want to consider backslash **** as special character in a rule, you need to enter: **`[\\]`** and not just: **`[\]`** • **`[^()][\']`** for allowing some special characters (for example, apostrophes) and disallowing others (for example, round brackets)
Whitespace characters	**`[\sA-Za-z]*`** for alphabetic characters (upper and lower case) and a whitespace character **Note** To disallow whitespace characters, specify the format that you want to allow (for example, **`[A-Za-z]{1,20}`** to allow 1–20 alphabetic characters). When you specify the allowed format like this, whitespace characters are automatically disallowed.

Allowed Format for User Entry	Possible Java Regular Expressions
Specific format	You can combine several Java regular expressions to define a specific format for user entries. For example, `[A-Za-z]{2}[\d]{22}` This example means that you define the following specific format: • `[A-Za-z]{2}`: Only two alphabetical characters (upper or lower case) • `[\d]{22}`: Followed by exactly 22 digits

6 Assigning Business Rules

A business rule is only triggered when it's assigned to the corresponding object or entity.

Prerequisites

- You've created a business rule with an application-specific rule scenario.
- For business rules created with the *Basic* rule scenario, you can't assign the rule using the rule assignment information icons. Please refer to the application-specific documentation for how to assign the rule.

> **Note**
>
> As rules using the basic rule scenario are more error-prone, we recommend using an application-specific scenario instead, provided there's one that meets your requirements. You can change the underlying scenario of a business rule using the *Change Scenario* link on the *Configure Business Rules* page of the corresponding business rule. Only if there's no application-specific scenario that fits your needs, use a basic rule scenario.

Context

Depending on how the application-specific scenario you're creating the rule for is set up, the rule needs to be assigned to an object, business process, event, action, screen, or some other entity. Therefore, please check the application-specific documentation to understand where the rule needs to be assigned to.

Procedure

1. You can assign or reassign business rules from either one of the following places in the system:
   - On the *Configure Business Rules* page, choose the rule assignment information icon next to the rule name.
   - On the *Business Rules Admin* page, choose the rule assignment information icon in the *Assigned* column of the corresponding rule.

   The rule assignment information icons have the following meaning:
   - *The rule is not assigned*
   - *The rule is assigned*
   - *There is no assignment information available*

> **Note**
>
> All rules based on application-specific scenarios have assignment information displayed. For rules based on the Basic scenarios, no assignment information is displayed.

2. If the rule isn't assigned (● *The rule is not assigned*), choose the icon and then the *Assign Rule* button to see a list of assignment options. If navigation is supported, select an option and assign the rule in the target application.
3. **Optional:** If the rule is already assigned (✔ *The rule is assigned*) and you want to reassign it, or check where it's assigned to, choose the icon to see a list of places where the rule is assigned. If navigation is supported, you can select an item to go to the target application for further actions.

Results

The next time the conditions defined in the assigned business rule are met, the rule is triggered.

7 Searching for Business Rules

Search and filter business rules on the *Business Rules Admin* page.

Context

On top of the *Business Rules Admin* page is the header section with a search box and a list of default filters. The search filters can contain different types of filters, such as date picker, select (from a dropdown list), and value help.

Procedure

1. Go to *Admin Center* > *Configure Business Rules*.

   You get to the *Business Rules Admin* page that lists all business rules.

   When you first open the page, the current date is set as the default date in the *As Of Date* filter, which is a required field.
2. **Optional:** To change which search filters are displayed in the header section, select *Adapt Filters*. Select which filters to show on the filter bar, and then select *Go*.

   > **Note**
   >
   > You can revert your changes by selecting *Restore*.

   The filters you've selected are displayed in the header section of the *Business Rule Admin* page.
3. Enter or select your search criteria using the search filters in the header section.
4. **Optional:** For fields with the value help icon ( ), you can perform advanced searches using conditions.

   > **Note**
   >
   > You can directly enter the input notation and value in the search field.
   >
   > For example: Enter `!(=1234)` in the *Rule ID* search field to exclude the rule with the ID "1234" from the search.

   a. Choose the value help icon ( ) on a field.
   b. In the *Define Conditions* dialog box, select the corresponding operators and enter the corresponding values you want to include or exclude from your search.

      For some fields, you have to choose the *Define Conditions* tab first.
   c. Select *OK*.
5. Select *Go* to show the results based on your search criteria.

   The business rules returned as a result of the filters are displayed in the list section.

6. **Optional:** To change the settings of the list, select ⚙ *Settings*. On the *View Settings* screen that opens, you can change the default columns to be displayed, sort the list by one or more fields, and group the list.

Next Steps

You can export the filtered list of business rules.

To view rule details and perform further actions, select a rule. You get to the *Configure Business Rules* page of the selected rule.

Related Information

Exporting Lists of Business Rules [page 61]
Search Filters [page 58]
Operators and Input Notations [page 59]

7.1 Search Filters

Search filters are displayed in the header section of the *Business Rules Admin* page.

You might have to adapt the filters to see all search filters available.

List of Available Search Filters

Category	Filter
Basic (Default filters)	Search
	As Of Date This filter is required. Default value is the current date.
	Scenario
	Last Modified By
	Rule Type
	Base Object
Additional	Rule ID
	Description
	Start Date
	Last Modified On

Category	Filter
	Created By
	Rule Name

7.2 Operators and Input Notations

Operators and input notations are used for search fields with value help on the *Business Rules Admin* page.

> **Note**
>
> - The texts entered as conditions are case-sensitive.
> - There's no explicit "not equal to" operator. Instead, it's available as the "equal to" operator in the *Exclude* section of the *Define Conditions* tab. When you enter a "not equal to" condition in the input field and press ENTER, the condition **`!=value`** is automatically converted to **`!(=value)`**.

Operators and Input Notations for Defining Search Conditions

Operator	Input Notation	Example
between	value...value	10...100
equal to	=value	=123
contains	*value*	*Rule*
starts with	value*	Rule*
ends with	*value	*Rule
less than	<value	<100
less than or equal to	<=value	<=100
greater than	>value	>100
greater than or equal to	>=value	>=100
not equal to	!=value	!=100

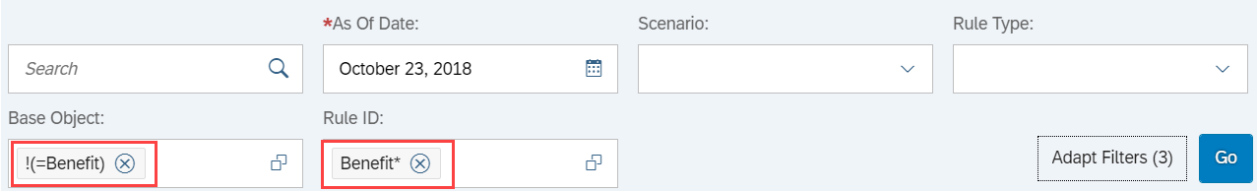
Example

In this example, you want to find business rules that are valid as of October 23, 2018 and whose IDs start with string "Benefit", excluding rules with base object Benefit. In the search fields, enter the following input notations:

Search Field	User Input
As Of Date	October 23, 2018
Base Object	**`!(=Benefit)`**

Search Field	User Input
Rule ID	`Benefit*`

Here's an example of what this looks like on the screen:



8 Export of Rule Information

When exporting rule information, you can either export lists of rules or lists of rules including the assignment information of the rules.

8.1 Exporting Lists of Business Rules

Export the list of business rules displayed on the *Business Rules Admin* page.

Procedure

1. Go to *Admin Center* > *Configure Business Rules*.

   You get to the *Business Rules Admin* page that lists all business rules.

   > ⓘ Note
   >
   > When you first open the page, the current date is set as the default date in the *As Of Date* filter, which is a required field.

2. **Optional:** To export only a selection of the business rules, enter search criteria in the search filters, and choose *Go*.

   The list of business rules is reduced to the ones that fit your search criteria.

3. **Optional:** To change which columns are displayed in the results list, select *Settings*. On the *View Settings* screen that opens, you can change the default columns to be displayed, sort the list by one or more fields, and group the list.

4. To export the list of business rules displayed, select *Export to Spreadsheet*.

Results

The business rules displayed on the *Business Rules Admin* page are downloaded in a spreadsheet. The exported list contains all filtered rules, not just the ones you've selected on the *Business Rules Admin* page.

> **Note**
>
> All columns displayed on that page are exported except the *Assigned* column.

8.2 Exporting List of Business Rules Including Assignment Information

Export list of all rules including the information about rule assignments to understand where the rules are assigned.

Context

In the Business Rules Admin page, you can export a list of all rules including their assignment information. When you use this export feature, a list of all rules is exported as a CSV File.

Procedure

1. Go to *Admin Center* > *Configure Business Rules*.
   You get to the *Business Rules Admin* that lists all business rules.
2. Choose *Export rules including assignment information as CSV file*.
3. Choose *Export* in the *Export Rules Including Rule Assignment Information* popup to initiate the export of business rules.
   You get to the *Scheduled Job Manager* > *Job Monitor* tab, which shows the job that exports the business rules.

   > **Note**
   >
   > It can take some time until your job shows up in the list. If it isn't listed, refresh the page.
4. Choose *View Details* in the *Action* column.
5. Choose *Download Status* in the *Run Details* popup to download the list of rules.

Results

The business rules are downloaded as a CSV file. The corresponding file name starts with`jobResponse`, followed by the job ID number, for example, `jobResponse274624.csv`. The exported list contains all rules that are valid on the date that was selected in the **As Of Date** filter of the **Business Rules Admin** page.

> **Note**
>
> When opening the downloaded CSV file in spreadsheet tools such as Microsoft Excel or other third-party software, make sure that you use the vendor's recommended way to open CSV files. Otherwise, you might end up with a corrupted display of the file with respect to line breaks, formatting, and so on.

> **Note**
>
> The assignment column of the downloaded list contains different information based on the kind of rule (basic rule or scenario-based rule) as well as the assignment itself.

Assignment Information and Its Meaning

Assignment Information	Meaning
`Rule for Defining Copy-Relevant Position Fields" on the "UI Customizing" tab under "Position Management Settings"`	Example of assignment information for a Position Management Rule based on an application-specific scenario.
`This rule is not assigned.`	The rule is based on an application-specific scenario, but not yet assigned.
`As this rule is based on the basic rule scenario, no assignment information is available. Move this rule to a specific rule scenario. You can either use the Change Scenario link on the Configure Business Rules page or the Check Tool check on the Migration tab of the Check Tool.`	The rule is based on the Basic scenario. The rule might be assigned, but the assignment information isn't available. The rule should be moved to a proper rule scenario. See the section *Change of Rule Scenarios* in this document.
`There was an issue with retrieving the assignment information. That might be a temporary issue. If the issue persists, please check the rule manually for configuration issues.`	It isn't possible to determine the assignment information (for example, because of technical errors). This happens rarely and only temporarily. If the status persists, check the rule manually for issues or contact support.
`There is an issue with the rule scenario assigned this rule. Retrieving the assignment information is not possible. Please check the rule manually for configuration issues.`	It isn't possible to determine the assignment information (for example, the rule scenario that is assigned is invalid). This happens rarely, for example, if you import rules and somehow the value for the rule scenario is corrupted. Use *Manage Data* to check the value in the *Scenario* field.
`As this rule is based on an inactive rule scenario, no assignment information is available. Most likely, the application the rule scenario belongs to was activated once, but was deactivated after the rule was created.`	The rule scenario isn't active (anymore). This happens if an application, such as Onboarding, was activated in the system and rules were created using the application-specific rule scenarios. If the application was deactivated after the rule was created, the assignment information can't be evaluated, because the application-specific rule scenarios are now also inactive. You can delete such a rule unless you plan to reactivate the application again and the rules must be kept for that reason.

9 Change of Rule Scenarios

If you've used the wrong scenario for a rule, or selected the wrong scenario attribute, you can change this to the correct application-specific scenario.

How to Proceed

For changing the scenario of an individual rule, follow the procedure for changing a rule scenario from the *Configure Business Rules* page.

You can find more information in the linked documentation.

> ⓘ Note
>
> Not all applications provide application-specific rule scenarios yet, nor the corresponding check for migration.

Related Information

Mass-Changing Rule Scenarios [page 64]
Changing Rule Scenarios Individually [page 65]

9.1 Mass-Changing Rule Scenarios

Change the underlying scenario of several rules at once to the expected application-specific scenario.

Prerequisites

You've enabled the *Check Tool*. You can find more information under Using the Check Tool to Solve Issues.

> ⓘ Note
>
> Not all applications provide application-specific rule scenarios yet, nor the corresponding check for migration.

Procedure

1. Open the *Migration* tab under *Admin Center* > *Check Tool*.

   You get a list of pending migrations. If the last run has found rules that should be assigned to application-specific scenarios, the corresponding checks will show an error here.
2. Select the corresponding checks and follow the proposed solution to solve the issue.

Next Steps

If there is no migration check for your application area, change the scenarios individually from the *Configure Business Rules* page, using the *Change Scenario* link.

Related Information

Changing Rule Scenarios Individually [page 65]

9.2 Changing Rule Scenarios Individually

Change the underlying scenario of an existing rule to an application-specific scenario to avoid errors.

Prerequisites

You've completed any pending migrations for basic rules in the Check Tool. You can find more information under Mass-Changing Rule Scenarios [page 64].

> **Note**
>
> The migration checks in the *Check Tool* offer the advantage of changing several rules at once, and directly suggesting the scenario that you should use. Note that not all applications might offer this option.

Context

> **Note**
>
> Not all applications provide application-specific rule scenarios yet. Please verify that there's a valid application-specific scenario for your rule before you undertake the change process.

You can also change from one application-specific scenario to another application-specific scenario, for example, if you've used the wrong scenario for a rule or selected the wrong scenario attribute.

Procedure

1. Go to *Admin Center* > *Configure Business Rules* and select your rule.

   The *Configure Business Rules* page is displayed.

   > **Note**
   >
   > If you see a message on top of the page about rules that can be migrated to an application-specific scenario, we recommend you first follow the link to the *Check Tool* to mass-change Basic rules.

2. Select the *Change Scenario* link that is displayed below the name of the rule.

   > **Note**
   >
   > You don't see this link if you are in edit mode.

   The change scenario wizard opens.

3. As a first step, we recommend you back up your data by choosing *Import and Export Data*, then go to the next step.
   - We strongly recommend you back up your data when changing from a *Basic* scenario to an application-specific scenario, as you won't be able to change the scenario back to *Basic* if the new rule scenario doesn't work as planned.
   - If you change from one application-specific scenario to another application-specific scenario, backing up your data beforehand is less important. However, you can do it for the case that the change to a new application-specific scenario leads to errors that haven't existed before.
4. Choose a new rule scenario, and go to the next step.
5. Select any mandatory attributes for the new scenario, and go to the next step.
6. Based on the selected scenario and the rule definition, the system carries out a series of validation checks to confirm that the rule scenario you chose can be used with the rule. When it's done, you get a message:

If you get this message	Proceed as follows
This scenario can be assigned to the rule. The rule doesn't contain any errors.	1. Choose *Submit*. 2. If you change from a *Basic* scenario to an application-specific scenario, remember that, once you change the rule scenario, you won't be able to change it back. You can export the rule before proceeding, if you haven't done so yet. If you want to proceed, confirm by choosing *Submit* again.
This scenario can be assigned to the rule. However, please note that there will be some errors in the rule that you should fix afterwards.	Make a note of the errors listed onscreen and then **either**: • Choose *Submit* and correct the errors within the rule once the new scenario has been assigned.

If you get this message	Proceed as follows
	• Choose *Cancel*, correct the errors, and then run the change process again. **Example** The rule raises an error message, but raising a message isn't supported or allowed by the selected rule scenario.
Sorry, you can't assign this scenario to the rule. There is at least one critical error within the rule that would prevent it from working correctly afterwards. Please fix the errors listed below, and then try to assign the new scenario again.	1. Make a note of the errors listed onscreen. 2. Choose *Cancel*, and either correct the errors within the rule, or choose another rule scenario. **Example** The rule uses a parameter that isn't supported by the selected rule scenario.

7. Once you confirm by choosing *Submit*, you get a message letting you know that the new rule scenario has been changed successfully. If you choose *Done*, you're brought back to the *Configure Business Rules* screen where you can see the updated rule.

9.3 Rule Scenarios

Rule scenarios help you create rules correctly, based on the rule context and parameters for a given scenario.

There are application-specific scenarios and the legacy basic scenario.

Most legacy business rules have been created with the default *Basic* scenario. Since the basic scenario doesn't provide any guidance about the various objects, parameters, and actions you can use to configure the rule, the resulting rules can often produce errors. As such, we recommend that you use application-specific rule scenarios instead.

→ Recommendation

If there's an application-specific scenario where you've used a basic scenario before, we recommend you change the basic scenario to an application-specific scenario.

Please note that not all applications provide specific scenarios, so in cases like this you would still have to use the basic scenario.

The scenarios available in your system are displayed when you create a new rule under *Admin Center* > *Configure Business Rules*, where they're categorized by application. Here you can see only the scenarios for the applications that you've enabled.

9.4 Overview of Application-Specific Rule Scenario Documentation

More information on rule scenarios can be found in the application-specific documentation of the category the rule scenario belongs to.

As new rule scenarios are created almost every release, the following list can be incomplete. It is meant to help you get started when looking for application-specific documentation.

> **Note**
>
> Please note that you can see only the scenarios for the applications or features that you have enabled in your system.

Category for Rule Scenario	Link to Documentation
Advances Management	Defining Rules to Determine Eligibility for an Employee
Benefits Management	Implementing and Configuring Global Benefits in Employee Central, under *Using Rules*
Calibration	Configuring Calibration Alert Rules in the System
Contingent Workforce Management	• Synchronizing Data Between Work Order and Job Information of Contingent Workers • Creating a Rule for Work Order Expiration Notification
Custom Tile (Decision Rules)	Configuring Custom Tile Behavior Based on Business Rules
Document Generation	Creating a Scenario Using a Business Rule
Email Services	Business Rules in Email Services
Employee Central Core	Rule Scenarios for Employee Central Core
Intelligent Services Center	Creating a Flow Rule for Intelligent Services
Location-Based Payments	Configure New Business Rules
Metadata Framework	MDF Rule Scenario
Onboarding 1.0	Verifying the Rule Registration Location Information
Onboarding	Rule Scenarios in Onboarding
Position Management	Rule Scenarios Available in Position Management
Performance Management Weighted Rating	Scenarios with Overall Customized Weighted Rating Calculation
Recruiting	Business Rules in Recruiting
Reward and Recognition	Configuring Rule Scenario to Define Eligibility for Spot Award Programs
Time Management	Implementing Employee Central Time Management
Compensation and Variable Pay	Eligibility Rules with Employee Central

10 Copying Rules

Copy existing rules to save work when creating similar rules.

Procedure

1. Go to *Admin Center* > *Configure Business Rules*.

   You get to the *Business Rules Admin* overview page.
2. Search and select the rule you want to copy.

   You get to the *Configure Business Rules* details page.
3. From the *Take Action* dropdown menu, select *Copy Rule*.
4. In the *Copy* dialog box that opens, enter a rule ID and a rule name for the new rule.
5. Choose *Copy*.
6. Change the rule copy as required, and save your changes.

11 Deleting a Rule Record

Delete records of a business rule on the *Configure Business Rules* page.

Prerequisites

If the rule has only one record, you've removed the assignment of that rule to the target object.

> **Note**
>
> Deleting the only record of a rule also deletes the entire rule. If this rule is assigned, you must unassign it before deleting the only record.

You can check the assignment for rules of certain rule scenarios.

Procedure

1. Go to *Admin Tools* > *Configure Business Rules*.

   You get to the *Business Rules Admin* page.
2. On the *Business Rules Admin* page, select a rule to get to the rule details on the *Configure Business Rules* page.
3. In the *History* section, choose *Take Action* > *Permanently Delete Entry* on the record you want to delete.
4. Confirm the deletion.

Results

The record is permanently deleted. If you're deleting the only record of the rule, the entire rule is also permanently deleted.

Related Information

Assigning Business Rules [page 55]

12 Deleting Rules Completely

Delete a rule or multiple rules completely on the *Business Rules Admin* page.

Prerequisites

You've removed all assignments of the rule.

> ⚠ Caution
>
> Do not delete a rule that is still assigned.

You can check the assignment for rules of certain rule scenarios.

Procedure

> ⚠ Caution
>
> When you delete a rule, all records of the rule are permanently deleted. If you want to delete only a record of the rule, please go to the *Configure Business Rules* detail page and delete it from there.

1. Go to *Admin Tools* > *Configure Business Rules* .

   You get to the *Business Rules Admin* page.

2. Select the rules and choose *(Delete)*.

   You can select up to 100 rules at a time for deletion.

3. Confirm the operation.

Results

The rules are deleted permanently. You are prompted with a pop-up dialog that shows the result. If a rule is still assigned, an error message appears.

Related Information

Assigning Business Rules [page 55]

Deleting a Rule Record [page 70]

13 Importing, Exporting, and Transporting Rules

13.1 Exporting Rules

You can copy rules from one company instance to another using the MDF import and export framework. Example, copying rules from the test instance to production instance.

> **Note**
>
> Before copying rules, please ensure that the Object Definitions, Language Package, and Picklists used in rules must be in sync. If the base object of the rule is an Employee Central object based on the succession data model, you must ensure to sync up the related data model before copying rules.

You can associate rules with MDF objects so that they are triggered at different object events such as on initialize, save, postsave, validate, or delete, or on individual field events such as on field change.

To export rules based on generic objects:

1. Go to the *Admin Center* > *Import and Export Data*.
2. In the *Import and Export Data* page, from the *Select action to perform* dropdown list, choose *Export Data*.
3. Specify the value for the fields that appear on UI.

Field	Description
Select Generic Object	Select *Rules*.
Include dependencies	Choose *Yes*.
Include immutable IDs	It is recommended to leave this field to its default value of *No*
Exclude reference objects	It is recommended to leave this field to its default value of *No*
Select all data records	Choose *No*. An additional *Select objects* field appears, where you can select the rules of your choice. Choose *Yes*. To export all the rules.

4. Click *Export*.

Results

An export job is triggered and added to the Job Scheduler queue. Check the result in *Scheduled Job Manager*. Once the export job is completed, you can download the object definition package using the *Download Status* link in *View Details*. The object definition package is a zip file with an individual CSV file for each individual object definition entities structure.

13.2 Importing Rule-Related Generic Objects

Once you export both the rules and object definitions from the source instance, you must import them to the target instance.

To import rule-related generic objects:

1. Go to *Admin Center* > *Import and Export Data*.
2. In the *Import and Export Data* page, from the *Select action to perform* dropdown list, choose *Import Data*.
3. In the form, choose the file type and the required generic object package.
4. Click *Import*.

Results

An import job is triggered and added to the Job Scheduler queue. Check the result in *Scheduled Job Manager*. Once the import job is completed, you can verify whether the rule-related generic objects have been imported by going to the *Configure Object Definition* page.

13.3 Importing Rules

You can import rules that you have previously exported from a source instance.

Procedure

1. Go to *Admin Center* > *Import and Export Data*.
2. On the *Import and Export Data* page, from the *Select action to perform* dropdown list, choose *Import Data*.
3. In the form, choose the tab for the file type you want to import.
4. If you import a CSV file, select *Rule* in the *Select Generic Object* field, and select the purge type.

> **Note**
>
> If you use an import file that was extracted from a source instance before the 2H 2022 upgrade, you must use the purge type *Full Purge*. If you want to use the purge type *Incremental Load*, make sure that the

import file contains all fields of the technical rule definition existing in the current release. For example, in 2H 2022 the field *legacyPicklistOptionMode* was added to the technical rule definition.

5. Choose the file.
6. Choose *Import*.

Results

An import job is triggered and added to the Job Scheduler queue. Check the result in *Scheduled Job Manager*. Once the import job is completed, you can verify whether the rule-related generic objects have been imported by going to the *Configure Object Definition* page. Once both the rules and object definition packages have been imported, the rules are migrated from the source instance to the target instance.

Next Steps

If you've imported a file that was extracted from a source instance before the 2H 2022 upgrade, you have to follow these steps:

- If the rules use option IDs for legacy picklist values, manually adjust the option IDs on the *Configure Business Rules* page. Else the rules might not deliver the results you expect.
- Run the *RuleTransportStabilization* migration check in the check tool. This check finds the rules that don't fulfill the current technical requirements to be stable in transport and import. Use the *Quick Fix* to migrate the rules to the current technical requirements to be stable in transport and import.

  > ⓘ Note
  >
  > Running the migration updates the *updated by* (*last modified by*) information with the user who executes the *Quick Fix*. The *last modified on* information is updated with the time stamp of when the migration was performed.
  >
  > Rules that contain configuration errors can't be migrated automatically with the quick fix. For these rules, open the results of the check in the *Check Tool* and select the corresponding result. The *Configure Business Rules* page opens with the corresponding rule, and the system automatically checks the rule for any potential errors. Manually correct the errors and save the rule, which automatically saves your correction and also automatically adapts the rule to fulfill the technical requirements for transport and import.
  >
  > Adopting the rule to be stable for transport and import doesn't change your configuration of the rule.

13.4 Constraints on Rules Imports

This section gives an overview on how to avoid a failing rule import.

Import can fail in the following scenarios:

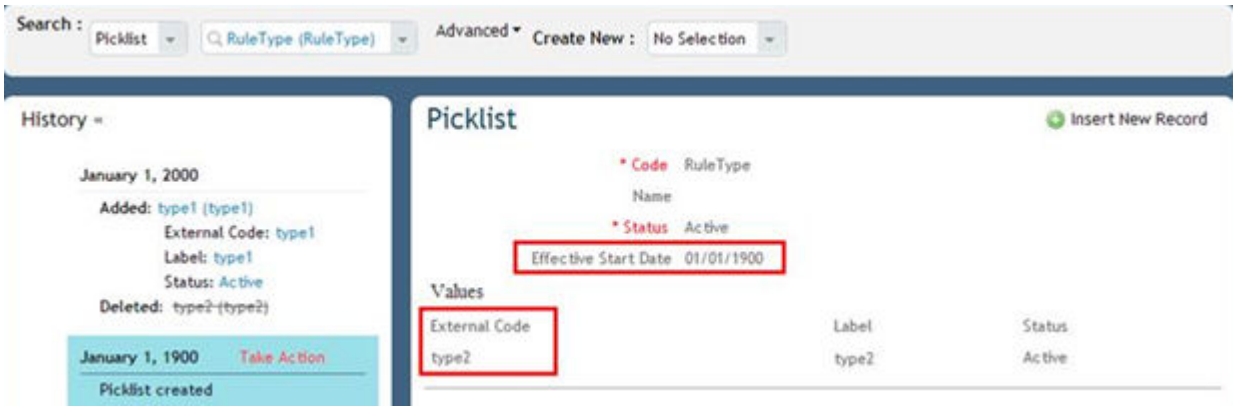
Effective Start Date

You already have a rule type whose effective start date is later than the effective start date of the rule type in the importing package, and the latter rule type doesn't have the value for the rule that is to be imported.

Example

In the following figure, the effective start date of the rule type is *01/01/2000* and it has only the value *type1*.

In the following figure, the effective start date of the rule type in the importing package is *01/01/1900* and it has the value *type2*. This value is for the rule that is to be imported.



The rule cannot be imported because the system cannot find type2. This is because it is not currently effective. You need to go into the rule and configure the rule type with only one record with the effective start date *01/01/1900*.

Legacy Platform Picklists

Previously, there was a constraint on importing rules that used legacy picklist values. If the base object of the rule was an Employee Central entity such as job information, and the rule used a legacy picklist field, the option ID of the picklist value was stored. As the option ID is different from instance to instance, you had to correct the corresponding rules in the target system to replace the source system picklist value option ID with the target system picklist value option ID.

This constraint is no longer valid. Instead of the option ID of the legacy picklist value, the external code of the picklist value is stored. As the external code is unique for all instances, no additional maintenance of the rules after importing in the target system is necessary. However, if you import old rules that still use option IDs for legacy picklist values, follow the steps mentioned in Importing Rules.

13.5 Adding a Rule to a Transport Bundle

Add a rule to a transport bundle so that you can transport it to a paired system using the Configuration Transport Center.

Prerequisites

You've created a transport bundle for the corresponding MDF object in Configuration Transport Center.

Context

You can use bundles to transport the configuration of a source system to a paired target system so that you don't need to manually configure it.

Procedure

1. Go to *Admin Center* > *Manage Data* and open a rule.
2. Select *Take Action* > *Add to Transport Bundle* to add the selected rule to an existing bundle.

   A list of available transport bundles are displayed.
3. Select the bundle you want to add the configuration to and select *Save*.

   Your configuration is successfully added to the transport bundle. A success message is displayed.

   If the rule has a Used Message Definition, then this is added to the transport bundle with the rule.
4. Select *Close*.

Results

The rule is added to the transport bundle.

14 Troubleshooting

Find out what to do when a rule doesn't work as expected.

Procedure

1. Create a business rule execution log. In the log file, trace the several execution steps of a rule to find possible errors in the rule definition.

   You can see if the rule is triggered and through which steps it runs. This way you can check if the rule does what you think you've defined in its If and Then sections.
2. If you can't find the error in the business rule execution log, check that there are no common errors in your rule.

   Creating Business Rule Execution Logs [page 79]
   Find possible errors in the rule definition when the rule doesn't work as expected.

   Common Errors [page 81]
   A list of the most common errors and how to prevent them.

   Fixing Issues with Migration for Stable Transport and Import [page 83]
   Fix the underlying issue when the *Business Rules Admin* page shows a warning message for check *RuleTransportStabilization*.

   Improving Performance [page 85]
   Guidelines on how to improve system performance during rule execution.

   Do All Permission Roles have Consistent Authorizations for Working with Business Rules? [page 88]
   The *PermissionRolesHaveConsistentAuthorizationsForRules* check validates if all permission roles have consistent authorizations for working with business rules.

14.1 Creating Business Rule Execution Logs

Find possible errors in the rule definition when the rule doesn't work as expected.

Prerequisites

You have the *Access to Business Rule Execution Log* permission, which belongs to the *Metadata Framework* group.

There are two separate permissions:

- *Configure*: This permission allows you to create the rule trace.
- *Including downloading the log*: This permission allows you to both create the rule trace **and** view the resulting log.

Since the log can contain potentially sensitive data on your users, we recommend that you assign the view permission only to the necessary people.

Context

Using this admin tool, you can specify which rules you want to be logged. Every time one of the specified rules is run, a log is created. You can then download logs in CSV format.

Procedure

1. Go to *Admin Center* > *Business Rule Execution Log*.
2. Choose *Create New* > *Rule Trace*.
3. On the *Rule Trace* screen, make the following entries:
   a. A name and a code for the rule trace.
   b. A start date and an end date. These dates define the time period for which the rule trace is active.

      > **Note**
      >
      > You can set up a rule trace for a maximum time period of **2 days**. This is to ensure that only new and up-to-date traces are active in the system at any particular time.

   c. A user. The rule trace is active only for this user. Rules executed by a different user aren't logged.

      > **Note**
      >
      > "User" here refers to the user who is executing the rule, not the user whose data is evaluated or changed by the rule.

      > **Note**
      >
      > When rules are executed in the context of a scheduled job, the technical user of the job execution needs to be specified (for example "v4admin").

   d. A trace mode. Select *Complete* to get all logs. Select *Short* to only get two log statements for each rule execution: The first log is written at the beginning of the rule execution and contains the parameters and objects that the rule was executed with. The log is written at the end of the rule execution and contains the execution status and a summary of the actions being executed.
   e. The rule or rules to be included in the log.

      If you want to trace **all** rules for the specified user, just leave this field empty.

      > **Note**
      >
      > The log file for a rule trace has a maximum threshold of 5 MB. If the log file exceeds this threshold, you must manaually delete some entries to remain under the threshold. So, if you're either tracing many or even all rules, or the rules are quite complex, then you might run into the problem that the log isn't

complete. You don't even see the specific rules that were executed. In this case, you have the following options:

1. Set the trace mode to *Short*. This mode reduces the number of logs significantly. Details of the rule executions aren't logged anymore, but you still see which rules were executed, which parameters were passed, which actions were executed, and the status for each rule execution.
2. Create separate rule traces where each rule trace only logs a single rule.

4. Save the rule trace you've defined.

Results

When you save the rule trace, it's active for the specified time period and creates a log file.

Next Steps

Reproduce the steps that execute the rules you want to trace. Choose **Download** to see the content of the log file.

> ⓘ Note
>
> You can delete the log file whenever you don't need it anymore by choosing **Delete (next to Download)**. This only deletes the log file, while the rule trace definition remains and can be used for further traces.

Task overview: Troubleshooting [page 79]

Related Information

Common Errors [page 81]
Fixing Issues with Migration for Stable Transport and Import [page 83]
Improving Performance [page 85]
Do All Permission Roles have Consistent Authorizations for Working with Business Rules? [page 88]

14.2 Common Errors

A list of the most common errors and how to prevent them.

- **A rule that sets the value of a foundation object can't be executed. The rule is defined as follows: 'set jobInfo.department.name = ABC' (base object is jobInfo).**

There's a limitation for setting foundation objects: You can only set foundation object values if the base object is equal to the foundation object. For example, if you redefine the rule in the following way, the rule is triggered:
Base object =Department
Then statement: 'Set Name = ABC'

- **When you export (using MDF Import and Export Data) and transport (using the Configuration Center) rules from the source instance and open them in the target instance, you get an error.**
  It's necessary to ensure that the base object, picklist, and foundation object are the same as in the source instance. Check the following:
  - The base object and the objects used in the rules exist in the target instance.
  - Picklists used in the rules are the same in the target instance.

    > **Note**
    >
    > If the option IDs of the picklist entries differ from target to source system, you can do one of the following:
    > - Use the Stable Transport Migration Check Tool check.
    > - Adjust the rules in the target system by selecting the corresponding option ID.

  - Foundation object values are the same in the target instance.

    > **Note**
    >
    > The system doesn't support the use of onChange business rules with foundation objects.

  - Message definitions are associated with the rules.
    In the source instance, you can run the *Rules have associations to all used message definitions* Check Tool check to find all rules that do not have an association to the message definitions, that are used in the rule configuration. Run the Quick Fix of this check to create these associations.
- **You use the rule to generate a sequence code for the Position Code, but the Position Code is updated every time you update the position data.**
  You can avoid this by triggering the rule using the 'save' event, or you can create one more rule for the initialization ('init') of the position. Here's an example:
  - Init Rule: If (always true) Then set positionCode = 'Automatic'
  - Save Rule: If (positionCode = 'Automatic') Then set positionCode = 'POS'+pos_seq
- **You've defined negative values for the sequence number on the *Configure Business Rules* page (for example, start number and step value), but the rule isn't executed.**
  Business rules are based on the Metadata Framework (MDF), which can't validate negative numbers. Therefore, avoid using negative numbers in Sequence.
- **You've defined different message types (such as warning, error), but you can't see any difference on the UI when the messages are displayed.**
  Employee Central doesn't support message types other than error messages. Therefore, regardless of what message type you select, all messages are treated as error messages.
- **You have 3 objects: A, B, C. Their association is: A is grandparent, B is parent, and C is child. A is the base object. When you try to set a value (for example, 'set A.B.C.field = 123'), the system returns an error.**
  If B doesn't exist, the system tries to create B. If you want to prevent this, add a condition which checks that B is not "null".
- **When you open a rule that raises a message, you see the following error message: "This rule contains errors, which are highlighted in the rule details. Please correct them."**

This message appears when there are unknown fields used in the rule, or when the message definition used in this rule has been changed. For example, a parameter has been added or removed, or the message definition has been deleted. In this case, you need to edit the business rule, and select the message definition again.

- **You've defined a rule with lookup function that used to work and suddenly fails.**
  Create a business rule execution log and check the log file to find out which part of the rule is failing.
  If the lookup function references effective-dated objects, such as picklists, ensure that the effective-dated object is valid throughout the time period when the rule is used. Otherwise, the rule doesn't find the corresponding entries of the referenced object when it's run, and fails.
- **When you use a collection filter or a lookup function, the required object isn't displayed in the right expression of the filter condition. How can you navigate to that object?**
  When using a collection filter or the lookup function, you can only navigate to objects with one-to-one associations. To navigate to objects with one-to-many associations, create a variable first, then reference that variable in the collection filter or lookup function.

Parent topic: Troubleshooting [page 79]

Related Information

Creating Business Rule Execution Logs [page 79]
Fixing Issues with Migration for Stable Transport and Import [page 83]
Improving Performance [page 85]
Do All Permission Roles have Consistent Authorizations for Working with Business Rules? [page 88]
Fixing Issues with Migration for Stable Transport and Import [page 83]
Creating a Message Rule [page 40]

14.3 Fixing Issues with Migration for Stable Transport and Import

Fix the underlying issue when the *Business Rules Admin* page shows a warning message for check *RuleTransportStabilization*.

Context

From 2H 2022, rules can be migrated to fulfill the technical requirements for a stable transport and import. For example, business rules refer to the unique, system-independent external code of picklist values instead of option IDs of picklist values.

But some rules might still not fulfill these technical requirements. Possible reasons are:

- You didn't run the migration so far.

- You've imported an old import file that was extracted from the system before the migration happened.
- There are auto-generated rules that don't fulfill all the technical requirements for the transport/import yet.

In this case, a warning message is displayed at the top of the *Business Rules Admin* page, informing you about open migrations for the check RuleTransportStabilization.

Procedure

1. Select the link contained in the message displayed on the *Business Rules Admin* page informing you about open migrations for the check RuleTransportStabilization.

   The *Check Tool* page opens.
2. On the *Migration* tab of the Check Tool, run the following migration check: *Rules are technically stable for transport and import. (RuleTransportStabilization)*.
3. Select *Quick Fix* and follow the guided procedure to perform the migration.

Results

The rules have been adapted to fulfill the technical requirements for a stable transport and import. For example, any usage of option IDs for legacy picklist values has been replaced by their unique external codes.

> **Note**
>
> Running the migration updates the *updated by* (*last modified by*) information with the user who executes the Quick Fix. The *last modified on* information is updated with the time stamp of when the migration was performed.
>
> In some cases it might happen that you see a picklist value with prefix "invalid_option_ID" (for example "invalid_option_ID_5739"). Reason is, that there was once a picklist entry with option ID 5739, that was used in the rule. Later on it was removed from the picklist without adjusting the rule, which made the rule inconsistent.
>
> This is detected by the migration and the invalid picklist value gets the prefix "invalid_option_ID". You are informed about this issue by a warning on the *Configure Business Rules* page.
>
> Note, that this doesn't have any effect on the rule execution. Before and after the migration this rule deals with the invalid picklist value in the same way.
>
> To make the rule consistent again it's necessary to correct the rule by selecting a valid picklist value.

Next Steps

Rules that contain configuration errors can't be migrated automatically with the quick fix. For these rules, open the results of the check in the *Check Tool* and select the corresponding result. The *Configure Business Rules* page opens with the corresponding rule, and the system automatically checks the rule for any potential errors. Manually correct the errors and save the rule, which automatically saves your correction and also automatically adapts the rule to fulfill the technical requirements for transport and import.

> **Note**
>
> Adopting the rule to be stable for transport and import doesn't change the functionality of the rule.
>
> The configuration issues that are displayed on the UI are not the result of the migration, but existed before. You can only complete the migration if you correct the errors and save the rule.

> **Note**
>
> Your system doesn't face any issues if you have a mixture of migrated rules and rules that aren't migrated. Rules that aren't migrated continue to run as expected. The migration only affects the transport and import of rules into another system. You still need to manually adjust not migrated rules in the target system because they use the option IDs of legacy picklist values instead of the unique, system-independent external code of picklist values.

Task overview: Troubleshooting [page 79]

Related Information

Creating Business Rule Execution Logs [page 79]
Common Errors [page 81]
Improving Performance [page 85]
Do All Permission Roles have Consistent Authorizations for Working with Business Rules? [page 88]

14.4 Improving Performance

Guidelines on how to improve system performance during rule execution.

Placing conditions at the top of a rule

If possible, place the following conditions at the top of a rule:

- Normal field from base object
  Normal fields means primitive type of field (such as string, Boolean, number, and so on), but not foundation objects or Metadata Framework objects. When you have a normal field in a long list of And/Or conditions, then please **move this to the top** as it can be executed fast, and the business rule checks this first. If it is false, then the rest of And conditions don't need to be executed. If it is true, then the rest of Or conditions don't need to be executed.
  See the example rule below, where *FTE = 1*.
- Employee Central picklist from base object
  When you use an Employee Central picklist in a long list of nested And/Or conditions, move this condition **to the top** of the rule. As an Employee Central picklist can be executed fast by the rule, it can be checked first. If

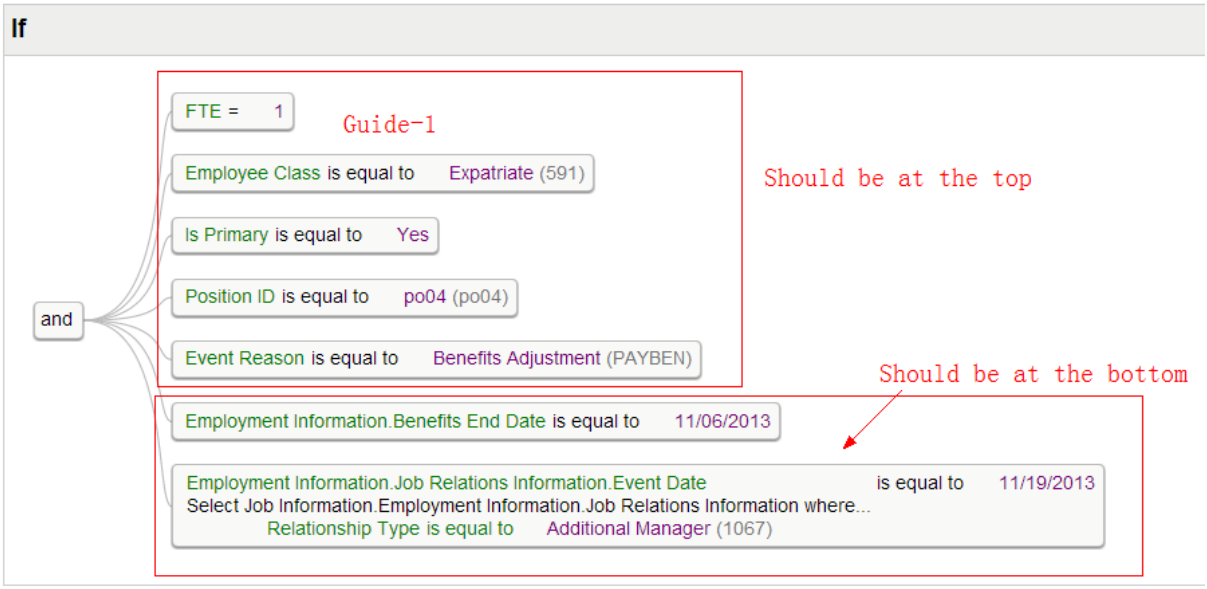
it is false, then the rest of And conditions don't need to be executed. If it is true, then the rest of Or conditions don't need to be executed.
See the example rule below, where *Employee Class is equal to Expatriate*.

- Comparing two Metadata Framework object fields
  When you compare two Metadata Framework object fields (for example, *jobInfo.position.legalEntity* is equal to *position.legalEntity*), move this condition **to the top**. It can be checked fast because the Metadata Framework object is not loaded, but instead, the system compares the internal code.
- Right expression for Metadata Framework object contains concrete value
  When the left expression is a Metadata Framework object, it is faster to select a concrete value for the right expression (for example, *ABC*, *01/01/2012*, and so on). If you select another field for the right expression, the system has to retrieve the concrete value itself, which takes more time.
  See the example rule below, where *Position ID is equal to po04*, which can be executed fast.

Placing conditions at the bottom of a rule

If possible, place the following conditions **at the bottom** of a rule:

- object.object.fields (where "object" is an Employee Central object)
  When objects are related to other objects in Employee Central, or when you create cross-card rules, for example, you navigate from one object to another object in Employee Central to get to a specific field (as in the example below: *EmploymentInformation.JobRelationsInformation.EventDate*). In such a case, the Employee Central to Employee Central API needs more time to get data from another big table, therefore you should put such a condition to the bottom of the rule.
- Collection Filters
  Collection filters call the Employee Central API to get the collection data first and then to filter the matched data according to the condition of the collection. This can be time-consuming if the collection size is big. You should put this to the bottom.
  Here is an example of a rule that considers the before-mentioned guidelines:



Keeping number low

If possible, keep the number of the following examples low in a rule, as these slow down system performance:

- object.object.fields and collection filters (where "object" is an Employee Central object)
  Because of the high amount of input for a business rules batch call, we don't suggest to use the following in the rule definition:
  - object.object.field (for example, *jobInfo.employmentInfo.custom_string1*)
  - Collection filters

  If you cannot avoid using these in a rule (for example, when you define cross-portlet rules), move them to the bottom of the rule.
- Define a function that has an SQL execution
  If you call batch execution, try to avoid using functions that require access to the database (for example, like the *Lookup* or *Get Next Value* functions), as they call an SQL script. According to the rule execution policy for batches, the batches are executed one by one, which means the function will be called many times.
- Load many rules at the same time
  It decreases the performance of the underlying Metadata Framework when you call up several rules at once, so please don't assign many rules to an object.

Parent topic: Troubleshooting [page 79]

Related Information

Creating Business Rule Execution Logs [page 79]
Common Errors [page 81]
Fixing Issues with Migration for Stable Transport and Import [page 83]
Do All Permission Roles have Consistent Authorizations for Working with Business Rules? [page 88]

14.5 Do All Permission Roles have Consistent Authorizations for Working with Business Rules?

The *PermissionRolesHaveConsistentAuthorizationsForRules* check validates if all permission roles have consistent authorizations for working with business rules.

Check ID	Check Name	Recommended Solution
PermissionRolesHaveConsistentAuthorizationsForRules	All Permission Roles have Consistent Authorizations for Working with Business Rules	If you find the following error in your system after running this check: • "We couldn't check the permission roles." go to *Admin Center* > *Configure Object Definitions* and ensure that both *Secured* and *CREATE Respects Target Criteria* are set to *Yes* for the rule object. Refer to *Permissions for Business Rules* in the Related Information section.
		If you find the following error in your system after running this check: • "Metadata Framework permission "Configure Object permission" is not granted." select the role name in the result list. This takes you to the *Manage Permission Roles* UI, where you can assign the *Configure Business Rules* permission. Instead of clicking on a role name, you can also navigate go to *Admin Center* > *Manage Permission Roles* > *Administrator Permissions* > *Metadata Framework* and assign the *Configure Business Rules* permission. Refer to *Permissions for Business Rules* in the Related Information section on how to set up the permissions.

Check ID	Check Name	Recommended Solution
		If you find the following errors in your system after running this check: • "Neither the view nor the edit permission is configured for the rule object." • "View/Edit permission is not correctly configured for the rule object." (For example, the user has only *View Current* of the View permissions, or *Correct* of the Edit permission.) select the role name in the result list. This takes you to the *Manage Permission Roles* UI, where you can configure the permissions consistently. Alternatively, go to *Admin Center* > *Manage Permission Roles*. Refer to *Permissions for Business Rules* in the Related Information section on how to set up the permissions.

> **Note**
>
> This check doesn't have a quick fix. The check runs asynchronously and might take some time until completion.

Parent topic: Troubleshooting [page 79]

Related Information

Creating Business Rule Execution Logs [page 79]
Common Errors [page 81]
Fixing Issues with Migration for Stable Transport and Import [page 83]
Improving Performance [page 85]
Permissions for Business Rules [page 14]

15 Functions By Groups

To help you get a quick overview of which functions are available to reach a specific aim, we have sorted the functions into groups. When you know you want to perform a calculation, for example, consult the mathematical functions overview to know which functions you can choose from.

Mathematical Functions [page 90]
: Functions to perform calculations.

Module-Specific or Feature-Specific Functions [page 91]
: Functions that can only be used when the corresponding module or feature is enabled.

String Functions [page 100]
: Functions you can use for text strings.

Time-Related and Date-Related Functions [page 101]
: Functions you can use for working with date/time fields.

"Execute" Functions [page 103]
: Functions you can choose when you select the **`Execute`** action as part of the Then statement in your rule.

Other Functions [page 105]
: Other functions that don't fall into one of the other groups.

15.1 Mathematical Functions

Functions to perform calculations.

For fields of type...	...you can use this function:	Choose this function to:
Decimal Number	Add [page 106]	Add values
Number	Digitsum [page 168]	Determine the sum of digits in the number
Decimal Number	Divide [page 169]	Divide values
Decimal Number	Math Expression [page 353]	Define your custom formula
Decimal	Minimum [page 354]	Find the smallest value in a list of values
Decimal Number	Minus [page 355]	Subtract values

For fields of type...	...you can use this function:	Choose this function to:
Number	Modulo [page 356]	Determine the remainder of a number divided by a divisor.
Decimal Number	Multiply [page 358]	Multiply values
Decimal	Opposite Sign [page 360]	Get a value with the opposite sign
Random	Random [page 361]	Return a random number
Decimal Number	Round [page 365]	Define how values are rounded
Decimal Number	Sum of Collection Field Values [page 370]	Sum up those values of a collection field that fulfill the filter conditions

Parent topic: Functions By Groups [page 90]

Related Information

Module-Specific or Feature-Specific Functions [page 91]
String Functions [page 100]
Time-Related and Date-Related Functions [page 101]
"Execute" Functions [page 103]
Other Functions [page 105]

15.2 Module-Specific or Feature-Specific Functions

Functions that can only be used when the corresponding module or feature is enabled.

Please note that for some of these functions, you have to enable the corresponding module or feature in Provisioning before you can use them on the *Configure Business Rules* page.

> → Remember
>
> As a customer, you don't have access to Provisioning. To complete tasks in Provisioning, contact your implementation partner or Account Executive. For any non-implementation tasks, contact Technical Support.

Employee Central

For fields of type...	...you can use this function:	Choose this function to:
Decimal	Amount from Pay Scale Structure [page 110] Note: Use only when you have defined pay scale structures in Employee Central	Get the amount from the pay scale structure
Text	Currency from Pay Scale Structure [page 162] Note: Use only when you have defined pay scale structures in Employee Central	Get the currency from the pay scale structure
	Frequency from Pay Scale Structure [page 177] Note: Use only when you have defined pay scale structures in Employee Central	Get the frequency from the pay scale structure
Value	Get Foundation Object Default Value [page 230] Note: Use only with Employee Central	Get the default value of a foundation object at runtime
	Get Generic Object Default Value [page 232] Note: Use only with Employee Central	Get the default value of a generic object at runtime
	Get Legacy Picklist Default Value [page 240] Note: Use only with Employee Central	Get the default value of a picklist for HRIS entities at runtime
	Get MDF Picklist Default Value [page 246] Note: Use only with Employee Central	Get the default value of the MDF picklist at runtime
Decimal	Get Pensionable Salary [page 289] Note: Use only with Employee Central	Calculate the pensionable salary on a certain effective date
Decimal	Get Pensionable Salary with Global Assignment [page 291] Note: Use only with Employee Central and Global Assignment	Calculate the pensionable salary on a certain effective date, considering global assignments

For fields of type...	...you can use this function:	Choose this function to:
Decimal Number	Get Work History Days ADD ALL [page 312] Note: Use only with Employee Central	Sum up days of all employments (current and previous)
Decimal Number	Get Work History Days CONTINUOUS [page 314] Note: Use only with Employee Central	Sum up days of all employments back until a specified event reason
Decimal Number	Get Work History Days CURRENT [page 318] Note: Use only with Employee Central	Sum up days of the current employment
Decimal Number	Get Work History Days PREVIOUS [page 321] Note: Use only with Employee Central	Sum up days of previous employments
Boolean	Is Employee Full Time Worker [page 342] Note: Use only with Employee Central	Determine whether the employee is full-time on a specific date
Text	Validate Higher Duty/Temporary Assignment Compensation Information [page 382] Note: Use only when you have enabled Employee Central and Higher Duty or Temporary Assignment.	Validate the compensation information and ensure that any change in the pay group of the nominal assignment is also updated in the compensation information of the currently active and future-dated higher duty or temporary assignments.
Text	Validate Higher Duty/Temporary Assignment Employment Information [page 384] Note: Use only when you have enabled Employee Central and Higher Duty or Temporary Assignment.	Validate the nominal and higher duty or temporary assignment employment information
Text	Validate Higher Duty/Temporary Assignment Job Information [page 385] Note: Use only when you have enabled Employee Central and Higher Duty or Temporary Assignment.	Validate the job information and ensure that any change to the company or country is not allowed if an active or future-dated higher duty assignment exists for the employee.

Employee Central Payroll

For fields of type...	...you can use this function:	Choose this function to:
Date	Get Next Possible Date for Changes [page 254] Note: Use only with Employee Central and Employee Central Payroll	Get the next possible date for payroll-specific data changes.
Value	Get Pay Calendar Info [page 277] Note: Use only with Employee Central and Employee Central Payroll	Get payroll calendar information that can be used to validate if payroll-specific data changes are allowed.
Value	Get Payroll Area Control Record [page 283] Note: Use only with Employee Central and Employee Central Payroll	Get the complete payroll control record from the payroll system
Value	Next Pay Level from Pay Scale Structure [page 358] Note: Use only with Employee Central and Employee Central Payroll	Get the next pay scale level from the pay scale structure based on the pay scale level, pay component, and effective date
Decimal (float)	Number from Pay Scale Structure [page 359] Note: Use only with Employee Central and Employee Central Payroll	Get the number from the pay scale structure based on the pay scale level, pay component, and effective date
Decimal (float)	Percentage from Pay Scale Structure [page 359] Note: Use only with Employee Central and Employee Central Payroll	Get the percentage from the pay scale structure based on the pay scale level, pay component, and effective date
Decimal	Rate from Pay Scale Structure [page 359] Note: Use only with Employee Central and Employee Central Payroll	Get the rate from the pay scale structure based on the pay scale level, pay component, and effective date
Value	Unit from Pay Scale Structure [page 359] Note: Use only with Employee Central and Employee Central Payroll	Get the unit of measure from the pay scale structure based on the pay scale level, pay component, and effective date

Position Management

For fields of type...	...you can use this function:	Choose this function to:
Decimal	Calculate FTE based on Standard Hours [page 146] Note: Use only with Position Management	Include positions in standard hours calculation
Text	Format Date for Position to Job Requisition Mapping [page 174] Note: Use only with Position Management	Convert date format to the format recognized by Requisition
Text	Get Incumbent By Position [page 233] Note: Use only with Position Management	Get the user ID of the incumbent of the position
Text	Get Matrix Position Code By Type [page 244] Note: Use only with Position Management	Get the matrix position code
Text	Get Next Available Manager By Position [page 253] Note: Use only with Position Management	Get the user ID of the next available manager in the position hierarchy
Decimal Number	Get Number Of Child Positions [page 263] Note: Use only with Position Management	Get the number of child positions
Text	Get Pay Range Attributes [page 281] Note: Use only with Position Management and pay ranges set up	Get the attributes of a pay range such as Minimum Pay, Maximum Pay, Mid Point, Currency, and Frequency
Decimal	Get Pay Range By Position [page 282] Note: Use only with Position Management	Get the associated pay range from the position
Boolean	Is Position Below User's Position In Hierarchy [page 343] Note: Use only with Position Management	Determine whether a position is below user's position in user's hierarchy

Time Management

For fields of type...	...you can use this function:	Choose this function to:
Decimal	Average Full-Time Equivalent for Accruable Period() Note: Use only with Time Off	Calculate the average FTE for the accruable period based on calendar days
Decimal	Average Full-Time Equivalent for Accruable Period Based on Months() Note: Use only with Time Off	Calculate the average FTE for the accruable period based on calendar months
Decimal	Average Full-Time Equivalent for Accrual Period() Note: Use only with Time Off	Calculate the average FTE for the accrual period of the time account based on calendar days
Decimal	Average Full-Time Equivalent for Accrual Period Based on Months() Note: Use only with Time Off	Calculate the average FTE for the accrual period based on calendar months
Decimal	Calculate Average Value For Numeric Job Info Field() [page 119] Note: Use only with Time Off	Calculate the average value of any numeric job information field
Decimal	Calculate Average Value For Numeric Job Information Field Based on Months() [page 124] Note: Use only with Time Off	Calculate the average value of any numeric job information field based on months
Decimal	Calculate Balance [page 139] Note: Use only with Time Off	Calculate the balance for a time account on a specific date
Decimal	Calculate Entitlement Balance() Note: Use only with Time Off	Retrieve the entitlement balance from a given time account or time type for a given date
Decimal	Cap Accrual [page 148] Note: Use only with Time Off	Determine the accrual value within the accrual balance limit
Text	Generate External Code For Time Off [page 178] Note: Use only with Time Off	Generate external code for Accruals in Time Off
Decimal	Get Absence Days() [page 179] Note: Use only with Time Off	Calculate how long an employee has been absent within one year for a specific time type
Decimal	Get Absence In Days For Period() [page 180] Note: Use only with Time Off	Calculate the cumulated absence days for the given period and a given time type

For fields of type...	...you can use this function:	Choose this function to:
Value	Get Absence In Days For Period Based On Calendar Days For Time Types() [page 182] Note: Use only with Time Off	Calculate the total number of absence days in a given duration for the time types that are provided
Decimal Number	Get Absence In Days For Period Based On Working Days For Leave Of Absence Time Types() [page 185] Note: Use only with Time Off	Calculate the number of absence days during a given duration for the given time types
Number	Get Absence In Days For Period Based On Working Days For Time Types() [page 188] Note: Use only with Time Off	Calculate the cumulated absence days for the given duration and given time types
Decimal	Get Absence In Days For Period Based On Working Days For Time Types Excluding Weekdays() Note: Use only with Time Off	Exclude the weekdays of your choice when getting the sum total of absence days
Value	Get Absence in Days For Period with Threshold() [page 194] Note: Use only with Time Off	Calculate the sum of absence days in a given period where time duraction exceeds a threshold value
Decimal	Get Absence In Hours For Period() [page 201] Note: Use only with Time Off	Calculate the cumulated absence hours for the given date period and given time types
Number	Get Absence In Hours For Period For Time Types() [page 203] Note: Use only with Time Off	Calculate the cumulated absence hours for the given duration and given time types
Decimal	Get Balance For Posting Types In Period() Note: Use only with Time Off	Get the balance of postings in a given period
Decimal Number	Get Completed Calendar Weeks Between (ISO Standard)() Note: Use only with Time Off	Calculate the number of weeks the employee works in the accrual period
Value	Get Completed Months Of Time Types In Period() [page 216] Note: Use only with Time Off	Calculate the number of complete months for which the employee was absent in a given period for the time types provided
Decimal Number	Get Completed Remaining Calendar Weeks (ISO Standard)() Note: Use only with Time Off	Calculate the number of weeks the employee works in the hire period

For fields of type...	...you can use this function:	Choose this function to:
Number	Get End Time Of Working Day () [page 225] Note: Use only with Time Off	Calculate the end time of the working day on a given date
Date	Get Job Info Date Field Value On Key Date() Note: Use only with Time Off	Get the job info date field for a specific key date
Number	Get Job Info Numeric Field Value On Key Date() Note: Use only with Time Off	Get the job info numeric field for a specific key date
Text	Get Job Info String Field Value On Key Date() Note: Use only with Time Off	Get the job info string field for a specific key date
Decimal Number	Get Months From Hire Date Taking Account Of Threshold() Note: Use only with Time Off	Calculate the number of months the employee is working in the hire period
Decimal Number	Get Months Taking Account Of Threshold() Note: Use only with Time Off	Calculate the number of months the employee is working in the accrual period
Number	Get Number of Absences for Period for Time Types() [page 260] Note: Use only with Time Off	Get the number of absences of a list of time types within a given period
Decimal Number	Get Number of Allowances in Period() [page 261] Note: Use only with Time Sheet	Get the number of allowances a user has within a given period of time
Number	Get Number Of Days For Year Of Date() [page 264] Note: Use only with Time Off	Get the number of days in the year for the date specified
Decimal	Get Number Of Eligible Days() [page 264] Note: Use only with Time Off	Calculate the number of eligible days
Decimal Number	Get Number Of Months From Hire Date() [page 267] Note: Use only with Time Off	Calculate the number of months the employee is working as of the hire date
Number	Get Number Of Postings For Posting Types in Period() [page 269] Note: Use only with Time Off	Get the number of postings in a given period

For fields of type...	...you can use this function:	Choose this function to:
Decimal	Get Number of Valuated Hours for Time Sheet() [page 271] Note: Use only with Time Sheet	Get the time valuation results from the time sheet base object that satisfy the optional parameters
Decimal Number	Get Purchased Leave Quantity or Equivalent Quantity (in Weeks) for Period [page 296] Note: Use only with Time Off	Get the sum of the time unit (in hours or days), or the equivalent quantity in weeks for purchased leave
Number	Get Start Time Of Working Day () [page 304] Note: Use only with Time Off	Calculate the start time of the working day on a given date
Decimal	Get Sum Of Accruals Since Last Transfer() [page 307] Note: Use only with Time Off	Get the sum of accruals of a given time account with entitlement method "Entitled as Transferred" or time type since the last transfer date (ad-hoc transfer or regular transfer, whichever is later)
Decimal	Get Sum Of Advances() [page 308] Note: Use only with Time Off	Get the advance balance from a given time account on a given date
Number	Get Time Management Earliest Recalculation Date () [page 309] Note: Use only with Time Off	Get the earliest date only if recalculation is active
Number	Get Time Sheet Approval Workflow Configuration [page 310] Note: Use only with Time Sheet	Get the workflow configuration when a time sheet is submitted
Text	Get Working Time Account on Key Date() [page 325] Note: Use only with Time Sheet	Get the working time account type from the time type profile assigned to the user's job info profile on the key date
Boolean	Has Absences In Period() [page 328] Note: Use only with Time Off	Check whether any active absences exist in a given period or after a given start date
Boolean	Has Allowances In Period() [page 330] Note: Use only with Time Sheet	Check whether a user has any allowances within a given period of time
Boolean	Has Payouts in Period() [page 331] Note: Use only with Time Off	Determine whether any active payouts exist in a given period or after a given start date
Boolean	Is Time Management Recalculation Active () [page 344] Note: Use only with Time Off	Validate whether time management recalculation is active or not

Parent topic: Functions By Groups [page 90]

Related Information

Mathematical Functions [page 90]
String Functions [page 100]
Time-Related and Date-Related Functions [page 101]
"Execute" Functions [page 103]
Other Functions [page 105]

15.3 String Functions

Functions you can use for text strings.

For fields of type...	...you can use this function:	Choose this function to:
Text	Concatenate [page 150]	Combine strings to one string
Text	Format [page 170]	Combine different objects to a meaningful text
Text	Format Number [page 175]	Format numbers following a custom template and sequence
	Get Picklist Value Label [page 293]	Get the label of a picklist value instead of the external code
Decimal Number	Index Of [page 337]	Get the index within a string of the first occurrence of the specified substring, starting at the specified index
Decimal Number	Length [page 347]	Get the length of a specified string
Text	Replace [page 362]	Search and replace a string
Text	Substring [page 368]	Return a new string that is a substring of the given string
Text	To Lowercase [page 372]	Convert text all lowercase
Text	To Uppercase [page 374]	Convert text all uppercase

Parent topic: Functions By Groups [page 90]

Related Information

Mathematical Functions [page 90]
Module-Specific or Feature-Specific Functions [page 91]
Time-Related and Date-Related Functions [page 101]
"Execute" Functions [page 103]
Other Functions [page 105]

15.4 Time-Related and Date-Related Functions

Functions you can use for working with date/time fields.

For fields of type...	...you can use this function:	Choose this function to:
Text	Convert Days To YY/MM/DD [page 152]	Convert a number of days into the format of years/months/days
Date	Create Date [page 159]	Create a date for a given day, month, and year
Date	Date Plus [page 164]	Add/subtract days or months to/from a given date
Number	Day Of Month [page 166]	Determine the number of the month for a specific date
Number	Day Of Week [page 166]	Determine the number of the weekday for a specific date
Date Decimal Number	Difference In Calendar Years [page 166]	Calculate the time difference between 2 dates in calendar years
Date Decimal Number	Difference In Years Round Down [page 167]	Calculate the time difference between 2 dates in calendar years, rounding down the result
Date Decimal Number	Difference In Years Round Up [page 168]	Calculate the time difference between 2 dates in calendar years, rounding up the result
Decimal Number	Get Completed Weeks Between Dates [page 216]	Calculate the number of completed weeks within a period
Date	Get First Day Of Month [page 228]	Get the first day of the same month of a specified date
Date	Get Fiscal Year Start or End Date [page 229]	Get the country/region-specific fiscal year start or end date
Decimal Number	Get Number Of Calendar Days() [page 263]	Calculate the number of calendar days between the start date and end date which are included in the calculated result
Number	Get Number Of Days For Year Of Date() [page 264]	Get the number of days, 365 ot 366, in the year of the date under consideration
Decimal Number	Get Number Of Months From Start Date Until End Date [page 269]	Determine the number of months between two given dates, excluding the end date from the calculation
Decimal	Get Number Of Working Days Or Hours For Period() [page 273]	Get the number of days or hours for an employee in a given period
Date	Get Pay Calendar Begin or End or Check Date [page 277]	Retrieve the start, end, or check date of a pay period that contains the given input date

For fields of type...	...you can use this function:	Choose this function to:
Date	Get Start Date of Next Pay Period [page 300]	Get the start date of the next pay period in the pay calendar
Date	Get Start Date of Next Pay Period Based on Pay Check Issue Date [page 302]	Get the start date of the next pay period in the pay calendar based on the pay check issue date
Date	Latest Date [page 346]	Get the latest date of two dates
Decimal Number	Month of Year [page 357]	Determine the month of the year for a given date
Text	Timestamp Current Time UTC plus Offset Minutes [page 371]	Create a current or future timestamp in UTC time standard
Date	Today [page 372]	Get the date of today
Decimal Number	Week Of Year ISO [page 386]	Determine the number of the week for a date (ISO-defined)
Decimal Number	Week of Year US [page 387]	Determine the number of the week for a date (US-defined)

> **Note**
>
> You can now define rule functions that accept the Time/DateTime MDF DataType as parameters and as a return type. This can be used, for example, in Time Off calculations.

Parent topic: Functions By Groups [page 90]

Related Information

Mathematical Functions [page 90]
Module-Specific or Feature-Specific Functions [page 91]
String Functions [page 100]
"Execute" Functions [page 103]
Other Functions [page 105]

15.5 "Execute" Functions

Functions you can choose when you select the **Execute** action as part of the Then statement in your rule.

> **Note**
>
> For some of these functions, you might have to enable the corresponding module or feature before you can use them on the *Configure Business Rules* page.

You can use this function...	...to do this:
Create Benefit Tracker for Employment Info Changes [page 154]	Create a record with user ID for the Benefits Employee Master Data Change Tracker object.
Create Benefit Tracker for Employment Info Changes with Effective Date [page 155]	Create a record with effective date and user ID for the Benefits Employee Master Data Change Tracker object.
Create Benefit Tracker for Personal Info Changes [page 156]	Create a record with person ID for the Benefits Employee Master Data Change Tracker object.
Create Benefit Tracker for Personal Info Changes with Effective Date [page 158]	Create a record with effective date and person ID in the Benefits Employee Master Data Change Tracker object.
Initiate PM Form On Job Change Event [page 338]	Launch a Performance Management form after job information changes in Employee Central.
Trigger Add Global Assignment Event	Publish the Add Global Assignment Event.
Trigger Create Concurrent Assignment Event	Publish the Create Concurent Assignment Event.
Trigger Email Notification for Onboarding 1.0 Processes	Send an email to notify the appropriate recipients to complete a task or provide them with information about specific activities completed in Onboarding 1.0.
Trigger Employee Time Alert Event	Publish the Employee Time Alert event.
Trigger End Global Assignment Event	Publish the End Global Assignment event.
Trigger Event for Restarting Onboarding Process [page 378]	Publish the Event for Restarting Onboarding Process.
Trigger First Time Manager Event	Publish the First Time Manager event.
Trigger Individual Contributor Becomes Manager Event	Publish the Individual Contribution Becomes Manager event.
Trigger Manager Becomes Individual Contributor Event	Publish the Manager Becomes Individual Contributor event.
Trigger MDF Alert Event	Publish the MDF Alert Event.
Trigger Name Change Event	Publish the Name Change Event.
Trigger New Hire Event	Publish the New Hire event.

You can use this function...	...to do this:
Trigger New Work Order Event	Publish the New Work Order Event.
Trigger Rehire Event	Publish the Rehire Manager event.
Trigger Start Probation Event	Publish the Start Probation Event.
Trigger Transfer Event	Publish the Transfer Event.
Trigger Work Order Expiration Event	Publish the Work Order Expiration Event.
Trigger Worker Absence Event [page 380]	Publish the Worker Absence event.
Trigger Worker Business Unit Change Event	Publish the Business Unit Change event.
Trigger Worker Department Change Event	Publish the Department Change event.
Trigger Worker Division Change Event	Publish the Division Change event.
Trigger Worker Job Classification Change Event	Publish the Worker Job Classification Change event.
Trigger Worker Job Title Change Event	Publish the Worker Job Title Change event.
Trigger Worker Location Change Event	Publish the Worker Location Change event.
Trigger Worker Long Term Disability Event [page 380]	Publish the Long Term Disability event.
Trigger Worker Manager Change Event	Publish the Worker Manager Change event.
Trigger Worker Short Term Disability Event [page 381]	Publish the ShortTerm Disability event.
Trigger Worker Termination Event	Publish the Worker Termination event.
Update of Employee Competency Assessment	Publish the Worker Competency Assessment Update Event.

Parent topic: Functions By Groups [page 90]

Related Information

Mathematical Functions [page 90]
Module-Specific or Feature-Specific Functions [page 91]
String Functions [page 100]
Time-Related and Date-Related Functions [page 101]
Other Functions [page 105]

15.6 Other Functions

Other functions that don't fall into one of the other groups.

For fields of type...	...you can use this function:	Choose this function to:
Text	Generate Unique Identifier [page 178]	Generate a unique identifier that consists of a 32-character long combination of numbers and letters
Number	Get Next Value [page 256]	Get next value from a Sequence MDF object
Boolean	Is Empty [page 343]	Check if an input field is empty
Boolean	Is User in Permission Group [page 344]	Check if the user is part of a specified permission group
Text	Login User [page 348]	Get the currently logged-in user
All field types	Lookup [page 348] Note: Use only for MDF objects	Get values from a lookup table (that is stored as MDF object)
Decimal	Treat Null As [page 375]	Give a default value for an empty field

Parent topic: Functions By Groups [page 90]

Related Information

Mathematical Functions [page 90]
Module-Specific or Feature-Specific Functions [page 91]
String Functions [page 100]
Time-Related and Date-Related Functions [page 101]
"Execute" Functions [page 103]

16 Functions A-Z

Find out more information about the corresponding functions; the functions are listed A-Z.

16.1 Add

This function adds two values.

Limitations

You can only use the US format for decimals (using dots, not commas).

Input Parameters

Input Parameter	Type	Required	User Entry
First	Decimal	Yes	Enter or select a number.
Second	Decimal	Yes	Enter or select a number.

Behavior in Case of Errors

If the first or second value is "null", the function returns "null".

> **Note**
>
> Please note that "null" does not mean the number zero in the context of business rules. Instead, it represents a missing value. A field with a null value is a field with no value.

16.2 Add Duration in Seconds to DateTime()

This rule function calculates a finishing UTC timestamp based on a starting UTC timestamp and a duration in seconds. Please note that if the duration is negative, the finishing timestamp will fall before the starting timestamp.

Input Parameters

Input Parameter	Type	Required	User Entry
DateTime	DateTime	Yes	Enter the timestamp.
Duration in Seconds	Decimal	Yes	Enter the duration in seconds to be added to the starting timestamp. Negative values are possible and will result in a finishing timestamp that falls before the starting timestamp.

Behavior in Case of Errors

If one of the input parameters is "null", the rule stops with the status FAIL.

> **Note**
>
> Please note that "null" does not mean the number zero in the context of business rules. Instead, it represents a missing value. A field with a null value is a field with no value.

Example: Negative Duration in Seconds

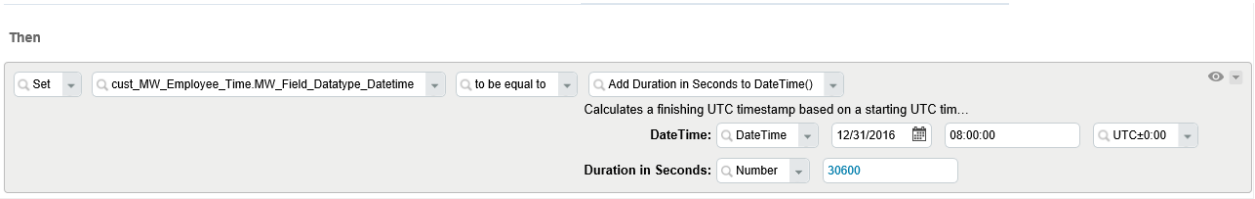
- *DateTime*: 01/01/2017 00:00:00 UTC
- *Duration in Seconds*:-30
- Result: 12/31/2016 23:59:30 UTC

Example: Positive Duration in Seconds

- *DateTime*: 12/31/2016 08:00:00 UTC
- *Duration in Seconds*: 30600

- Result: 12/31/2016 16:30:00 UTC

Here's an example of how this function is used in the Then statement of a rule, using the Set action:



16.3 Add Multiple

This function calculates the sum of multiple numbers.

Input Parameters

For this parameter...	Of type...	Which is...	Make this entry:
Number	Decimal	Mandatory	You can do one of the following: • Select *Decimal* and enter a number. • Select a field that contains a number, for example: FTE of Job Information. **Note** If the field you selected as input parameter is "null", this function stops and also returns null as result. You can prevent this by making sure that a possible null value of a field is treated as the number zero. To achieve this, first select the function *Treat Null As*, then select the field that could return null.

Example

Input Parameters	Result
Add Multiple (11.11,11.11,11.11,11.11)	44.44

Input Parameters	Result
Add Multiple (11.999,12.999,13.999)	38.997
Add Multiple (1,2,3)	6

16.4 Adjust Pay Scale Level

This rule function can be used to adjust pay scale levels up or down.

It checks the Job Information History and checks every record for a pay scale level to see what needs to be adjusted. It can be used, for example, in grade step progression processes, as well as for scenarios where employees return from assignments or receive an adjustment based on seniority.

The calculation checks in the Job Information History and searches for the first record with a Pay Scale Level. As a second step, the data is given in the rule to compare with that date in the pay scale level time period. Assuming the employee was hired with a first pay scale level, the rule function checks how often the frequency (given in the rule function) fits into the time period and writes the correct pay scale level for the new record.

Input Parameters

For this parameter...	Make this entry...
Seniority Date (or any other date field)	Enter the date to start the calculation from, for example, entry date.
Frequency	How often the pay scale level was adjusted for the employee

16.5 Amount from Pay Scale Structure

This function gets the amount from the pay scale structure based on the pay scale level, pay component, and effective date. The pay scale structure is configured in the MDF object *Pay Scale Level*.

Limitations

Use only if the *Pay Scale Group* and *Pay Scale Level* fields are included in the job information.

For more information about how to set up pay scale structures, see the SuccessFactors Employee Central Payroll Integration Guide.

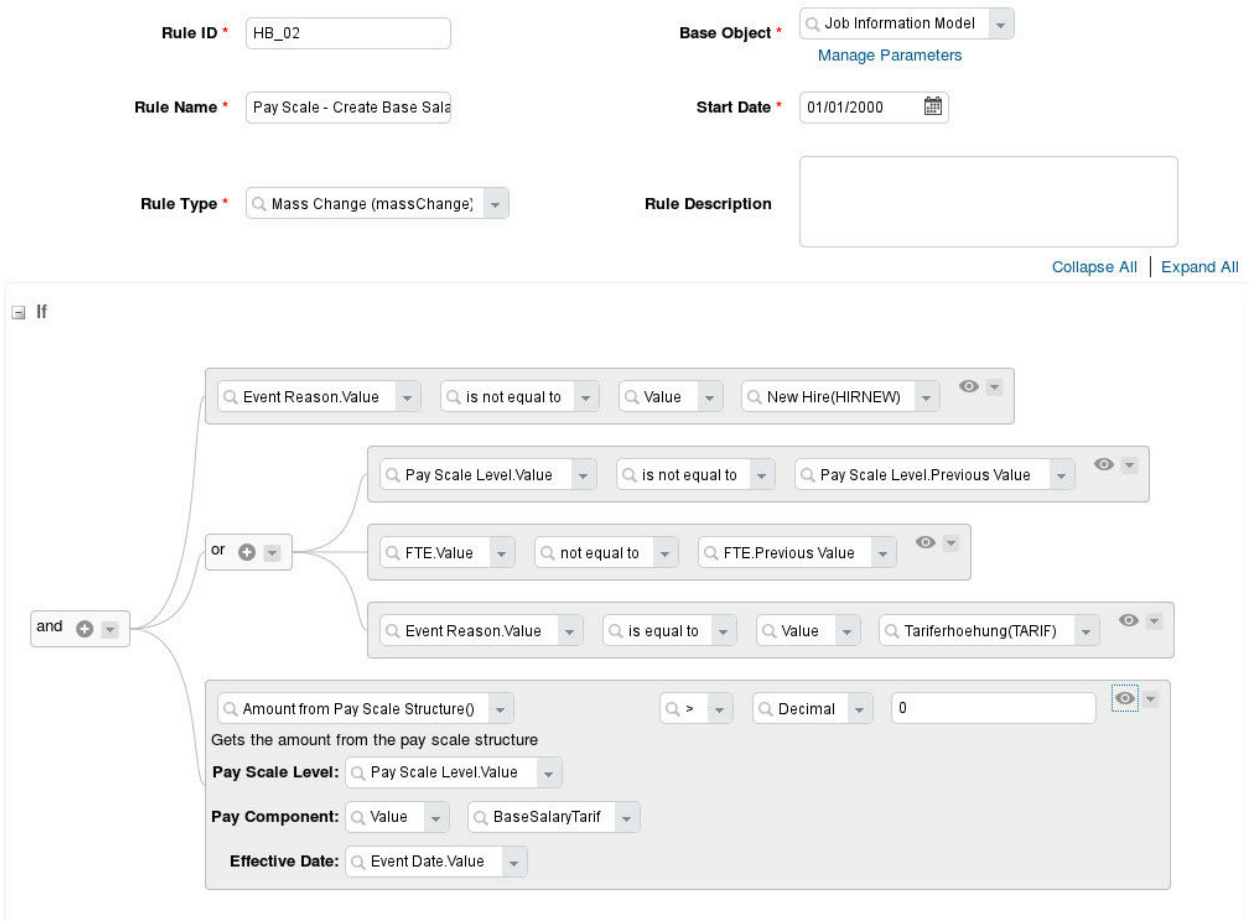
Input Parameters

For this parameter...	Of type ...	Which is...	Make this entry:
Pay Scale Level	Text	Required	Select the pay scale level you want to use. This is usually the value which is entered in the job information. The pay scale level object must be the SAP pay scale level foundation object, and cannot be a custom MDF object.
Pay Component	Value	Required	Select the *Value* field type, and select the pay component you want to search for in the pay scale structure. Usually, this is the pay component you want to create in the compensation model.
Effective Date	Date	Required	Enter or select the effective date you want to use to read the data from the pay scale structure. Usually, this is the event date of the job information.

Use Case

You can create rules that automatically assign the correct pay components to an employee based on the pay scale level and pay scale group you define for that employee.

Here's an example of what such a rule could look like – for the complete process of how to set up such rules, please refer to the SuccessFactors Employee Central Payroll Integration Guide:



Related Information

Currency from Pay Scale Structure [page 162]
Frequency from Pay Scale Structure [page 177]

16.6 Average Full-Time Equivalent for Accruable Period()

This function calculates the average full-time equivalent (FTE) as defined in the job information for the accruable period.

Overview

The accruable period is the period the employee is allowed to get an accrual for. The function is used for fields that show data in numeric form, to calculate accruals depending on FTE. The rule function counts the days for which the employee is not eligible as FTE = 0 - for example, if the employee is on parental leave for 2 months within the period.

The function is used for fields that show data in numeric form, to calculate accruals depending on FTE.

Manage Parameters

When you click *Manage Parameters*, make the following entries:

Code	Name	Object
accrualRuleParameters	accrualRuleParameters	Accrual Rule Parameters

Input Parameters

Here's a table showing the entries you need to make in the input parameters.

For this parameter:	Make this entry
User ID	Select the *User* field. This is the user for which the accrual should be created.
Time Account Type	Select the time account type the rule is valid for.
Start Date	Please use Manage Parameters accrualRuleParameters and select Start Date.
End Date	Please use Manage Parameters accrualRuleParameters and select End Date.
Accruable Start Date	Please use Manage Parameters accrualRuleParameters and select Accruable Start Date.

For this parameter:	Make this entry
Accruable End Date	Please use Manage Parameters accrualRuleParameters and select Accruable End Date

Example

Here's the data for A.N.Other's average FTE accrual calculation. The employee is hired on July 1, 2017 and the FTE is 1. The vacation account has an annual frequency period starting on January 1.

User ID: A.N. Other

Time Account Type: = Vacation

Start Date = January 1, 2017

End Date = December 31, 2017

Accruable Start Date = July 1, 2017

Accruable End Date = December 31, 2017

Result

A.N.Other's average FTE would be calculated for the period July 1 to December 31. The result would be 1.

16.7 Average Full-Time Equivalent for Accruable Period Based on Months()

This function calculates the average full-time equivalent (FTE) as defined in the job information for the accruable period.

Overview

The accruable period is the period the employee is allowed to get an accrual for. The average calculation is based on calendar months, meaning that the average of each month within the accruable period is calculated first. The rule function counts the days for which the employee is not eligible as FTE = 0 - for example, if the employee is on parental leave for 2 months within the period.

The function is used for fields that show data in numeric form, to calculate accruals depending on FTE.

Manage Parameters

When you click *Manage Parameters*, make the following entries:

Code	Name	Object
accrualRuleParameters	accrualRuleParameters	Accrual Rule Parameters

Input Parameters

Here's a table showing the entries you need to make in the input parameters.

For this parameter	Make this entry
User ID	Select the *User* field. This is the user for which the accrual should be created.
Time Account Type	Select the time account type the rule is valid for.
Start Date	Please use Manage Parameters accrualRuleParameters and select Start Date.
End Date	Please use Manage Parameters accrualRuleParameters and select End Date.
Accruable Start Date	Please use Manage Parameters accrualRuleParameters and select Accruable Start Date.
Accruable End Date	Please use Manage Parameters accrualRuleParameters and select Accruable End Date.

Example

Let's look at an example.

Here's the data for A.N.Other's FTE accrual calculation. The employee is hired on July 1, 2017 and the FTE is 1. The vacation account has an annual frequency period starting on January 1.

User ID: A.N. Other

Time Account Type: = Vacation

Start Date = January 1, 2017

End Date = December 31, 2017

Accruable Start Date = July 1, 2017

Accruable End Date = December 31, 2017

Result

A.N.Other's average FTE is calculated for the period July 1 to December 31. The result is 1.

16.8 Average Full-Time Equivalent for Accrual Period()

This function calculates the average full-time equivalent (FTE) as defined in the job information for the accrual period.

Overview

The accrual period is the period derived from the employee's time account type, For example, for an annual account starting on January 1, the accrual period would be January 1 until December 31. The rule function counts the days for which the employee is not eligible as FTE = 0 (for example, if the employee is on parental leave for 2 months within the period.

The function is used for fields that show data in numeric form, to calculate accruals depending on FTE.

Manage Parameters

When you click *Manage Parameters*, make the following entries:

Code	Name	Object
accrualRuleParameters	accrualRuleParameters	Accrual Rule Parameters

Input Parameters

Here's a table showing the entries you need to make in the input parameters.

For this parameter:	Make this entry
User ID	Select the *User* field. This is the user for which the accrual should be created.
Time Account Type	Select the time account type the rule is valid for.
Start Date	Please use Manage Parameters accrualRuleParameters and select Start Date.

For this parameter:	Make this entry
End Date	Please use Manage Parameters accrualRuleParameters and select End Date.
Accruable Start Date	Please use Manage Parameters accrualRuleParameters and select Accruable Start Date.
Accruable End Date	Please use Manage Parameters accrualRuleParameters and select Accruable End Date.

Example

Here‘s the data for A.N.Other‘s average FTE accrual calculation. The employee is hired on July 1, 2014 and the FTE is 1. The vacation account has an annual frequency period starting on January 1.

User ID: A.N. Other

Time Account Type: = Vacation

Accruable Start Date = July 1, 2017

Accruable End Date = December 31, 2017

Start Date = January 1, 2017

End Date = December 31, 2017

Result

A.N. Other's average FTE is calculated for period January – December as 0.504.

The period January to June has 181 days and the period July to December has 184 days. The result is the following formula, which yields the result of 0.504:

(181 days * 0 + 184 days * 1) / 365

16.9 Average Full-Time Equivalent for Accrual Period Based on Months()

This function calculates the average full-time equivalent (FTE) as defined in the job information for the accrual period.

Overview

The accrual period is the period derived from the employee's time account type. For example, in the case of an annual account starting on January 1, the accrual period would be January 1 to December 31. The average

calculation is based on calendar months, meaning that the average of each month within the accruable period is calculated first. The rule function counts the days for which the employee is not eligible as FTE = 0 - for example, if the employee is on parental leave for 2 months within the period.

The function is used for fields that show data in numeric form, to calculate accruals depending on FTE.

Manage Parameters

When you click *Manage Parameters*, make the following entries:

Code	Name	Object
accrualRuleParameters	accrualRuleParameters	Accrual Rule Parameters

Input Parameters

Here's a table showing the entries you need to make in the input parameters.

For this parameter	Make this entry
User ID	Select the *User* field. This is the user for which the accrual should be created.
Time Account Type	Select the time account type the rule is valid for.
Start Date	Please use Manage Parameters accrualRuleParameters and select Start Date.
End Date	Please use Manage Parameters accrualRuleParameters and select Accrual End Date.
Accruable Start Date	Please use Manage Parameters accrualRuleParameters and select Accruable Start Date.
Accruable End Date	Please use Manage Parameters accrualRuleParameters and select Accruable End Date.

Example

Here's the data for A.N.Other's FTE accrual calculation. The employee is hired on July 1, 2017 and the FTE is 1. The vacation account has an annual frequency period starting on January 1.

User ID: A.N. Other

Time Account Type: = Vacation

Start Date = January 1, 2017

End Date = December 31, 2017

Accruable Start Date = July 1, 2017

Accruable End Date = December 31, 2017

Result

Average FTE would be calculated for period January – December. The result is 0.5. The average for each of the months January – June is 0 and the average for each of the months July – December is 1. The result is the following formula, which yields the result of 0.5: (6 * 0 + 6 * 1) / 12.

16.10 Calculate Average Value For Numeric Job Info Field()

This rule function calculates the average value of any numeric Job Information field.

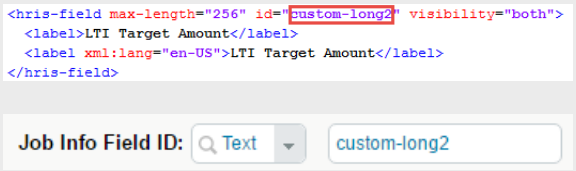
You can decide whether you want to calculate the average for the **accrual** period or for the **accruable** period. In addition, you can decide whether the eligibility status should be considered.

> ⚠ Restriction
>
> Please use this rule function only in the Accrual scenario.

Look at the examples below for further details.

Input Parameters

For this parameter	Make this entry
User	Select the *User* field. This is the user the average will be calculated for.

For this parameter	Make this entry
Job Info Field ID	**⚠ Caution** Please enter the field ID from your data model. Type it in manually - do **not** navigate to the corresponding Job Information field.  The field in your data model must be a numeric field to calculate an average. For example: custom-long1 or custom-double1.
Start Date	Please use Manage Parameters accrualRuleParameters and select Start Date.
End Date	Please use Manage Parameters accrualRuleParameters and select End Date.
Consider Eligibility	Please choose whether you want the eligibility status to be considered in the calculation or not (refer to example below).
Time Account Type	Please select the Time Account Type.
Accruable Start Date	Please use Manage Parameters accrualRuleParameters and select Accruable Start Date.
Accruable End Date	Please use Manage Parameters accrualRuleParameters and select Accruable End Date.
Calculation Period	Please choose whether you want the average value calculation for the accrual period or for the accruable period (refer to example below).

Examples

Let's look at some examples.

Case 1: Without Eligibility and for Accrual Period

- User: A. N. Other
- Job Info Field ID: standard-hours
- Start Date: January 1, 2016

- End Date: December 31, 2016
- Consider Eligibility: No
- Time Account Type: Vacation
- Accruable Start Date: January 1, 2016
- Accruable End Date: October 31, 2016 (because the employee is terminated).
- Calculation Period: Accrual Period

Job Information Data

Effective Start Date	Effective End Date	Standard Hours
01.01.2016	31.03.2016	40
01.04.2016	31.12.2016	30

Result

The average will be calculated for the complete year 2016. The employee would no longer be eligible after termination date, but the eligibility should not be considered.

For the period from January 1, 2016, until March 31, 2016, the employee has 40 standard hours each week. The period consists of 91 days.

For the period from April 1, 2016 until December 31, 2016, the employee has 30 standard hours each week. The period consists of 275 days.

40 * 91 / 366 + 30 * 275 / 366 = 9.945 + 22.54 = 32.49

32.49 is the average value of the standard hours for the year 2016.

Case 2: With Eligibility and for Accrual Period

- User: A. N. Other
- Job Info Field ID: standard-hours
- Start Date: January 1, 2016
- End Date: December 31, 2016
- Consider Eligibility: Yes
- Time Account Type: Vacation
- Accruable Start Date: January 1, 2016
- Accruable End Date: October 31, 2016 (because the employee is terminated).
- Calculation Period: Accrual Period

Job Information Data

Effective Start Date	Effective End Date	Standard Hours
01.01.2016	31.03.2016	40
01.04.2016	31.12.2016	30

Eligibility Status

Effective Start Date	Effective End Date	
01.01.2016	31.12.9999	**Yes**

The employee is eligible starting from the hire date until high date. In this scenario, a termination rule is executed. This happens before the termination entry is saved in job info and the eligibility status is updated. However, the accruable end date is already correct (= last working day) and is used to determine that the employee is no longer eligible outside this period.

Result

The average will be calculated for the complete year 2016. The employee is no longer eligible after the termination date.

For the period from January 1, 2016, until March 31, 2016, the employee has 40 standard hours each week. The period consists of 91 days.

For the period from April 1, 2016 until October 31, 2016, the employee has 30 standard hours each week. The period consists of 214 days.

40 * 91 / 366 + 30 * 214 / 366 = 9.945 + 17.54 = 27.49

27.49 is the average value of the standard hours for the year 2016.

Case 3: Without Eligibility and for Accruable Period

- User: A. N. Other
- Job Info Field ID: standard-hours
- Start Date: January 1, 2016
- End Date: December 31, 2016
- Consider Eligibility: No
  However, the employee is not eligible between January 1, 2016, and January 31, 2016, because he is on long term leave.
- Time Account Type: Vacation
- Accruable Start Date: January 1, 2016
- Accruable End Date: October 31, 2016 (because the employee is terminated).
- Calculation Period: Accruable Period

Job Information Data

Effective Start Date	Effective End Date	Standard Hours
01.01.2016	31.03.2016	40
01.04.2016	31.12.9999	30

Eligibility Status

Effective Start Date	Effective End Date	Eligibility Status
01.01.2016	31.01.2016	**No**

Effective Start Date	Effective End Date	Eligibility Status
01.02.2016	31.12.9999	**Yes**

The employee is not eligible for the first month because of any leave of absence. Afterwards, he is eligible until the high date. In this scenario, a termination rule is executed before the termination entry is saved in job info and the eligibility status is updated. However, the accruable end date is already correct (= last working day) and is used to determine that the employee is no longer eligible outside this period.

Result

The average will be calculated for the period from January 1, 2016 until October 1, 2016 (305 days). The employee is not eligible in January, but the eligibility should not be considered.

For the period from January 1, 2016, until March 31, 2016, the employee has 40 standard hours each week. The period consists of 91 days.

For the period from April 1, 2016 until October 31, 2016, the employee has 30 standard hours each week. The period consists of 214 days. November and December are outside the accruable period.

40 * 91 / 305 + 30 * 214 / 305 = 11.93 + 21.05 = 32.98

32.98 is the average value of the standard hours for the year 2016.

Case 4: With Eligibility and for Accruable Period

- User: A. N. Other
- Job Info Field ID: standard-hours
- Start Date: January 1, 2016
- End Date: December 31, 2016
- Consider Eligibility: Yes
  However, the employee is not eligible between January 1, 2016, and January 31, 2016, because she is on long term leave.
- Time Account Type: Vacation
- Accruable Start Date: January 1, 2016
- Accruable End Date: October 31, 2016 (because the employee is terminated).
- Calculation Period: Accruable Period

Job Information Data

Effective Start Date	Effective End Date	Standard Hours
01.01.2016	31.03.2016	40
01.04.2016	31.12.9999	30

Eligibility Status

Effective Start Date	Effective End Date	Eligibility Status
01.01.2016	31.01.2016	**No**

Effective Start Date	Effective End Date	Eligibility Status
01.02.2016	31.12.9999	**Yes**

The employee is not eligible for the first month because of any leave of absence. Afterwards, he is eligible until the high date. In this scenario, a termination rule is executed before the termination entry is saved in job info and the eligibility status is updated. However, the accruable end date is already correct (= last working day) and is used to determine that the employee is no longer eligible outside this period.

Result

The average will be calculated for the period from January 1, 2016 until October 1, 2016 (305 days). The employee is not eligible in January.

For the period from January 1, 2016, until January 31, 2016, the employee is not eligible, so there are no standard hours.

For the period from February 1, 2016, until March 31, 2016, the employee has 40 standard hours each week. The period consists of 60 days.

For the period from April 1, 2016 until October 31, 2016, the employee has 30 standard hours each week. The period consists of 214 days. November and December are outside the accruable period.

40 * 60 / 305 + 30 * 214 / 305 = 7.87 + 21.05 = 28.92

28.92 is the average value of the standard hours for the year 2016.

16.11 Calculate Average Value For Numeric Job Information Field Based on Months()

This rule function calculates the average value of any numeric job information field based on months. You can prorate or recalculate the leave accrual value depending on the changes to the job information field.

When using this rule function, you can decide:

- Whether you want to calculate the average for the **accrual** period or for the **accruable** period.
  - Make sure that the *Accruable End Date* isn't null and not greater than the *Accrual End Date*.
  - If you use the Accruable Period calculation method while using the Contract End Date field in the job information, you must specify the Contract End Date as Accruable End Date for the Accruable Period calculation.
  - If you use the Accrual Period calculation method, job information slices before and after the accruable period are counted as 0 independently of the content in the corresponding job information field.
- Whether to consider a particular month (or not) for calculations based on specified thresholds.
- If an employee is hired on or before a specified threshold start value, then the hire month is considered.
- If an employee's employment is terminated after a specified threshold end value, then the termination month is considered.
- If there is a change in the job information of the employee on or before a specified threshold value, then that month is considered for the new value of the job information field.

- If an employee is hired and their employment terminated in the same month, then threshold end is considered.
- The Threshold End calculation considers the Contract End Date in the job information or the Termination date of the user, whichever value is lower.
- You have to specify a threshold value. It is possible to enter a value of "0", but please keep the following behavior in mind:
  - If *Threshold Start* is 0, the rule function will **ignore** the month in which the user was hired.
  - If *Threshold End* is 0, the rule function will **include** the month in which the user was terminated.
  - If *Threshold Data Change* is 0, then the old value of the job information field will be used for the calculation.

Restrictions

- This rule function should only be used in the Accrual scenario.
- On the Termination UI, this rule function doesn't consider the termination date from the Termination UI because the rule function reads from database. So, you might see a discrepancy in the balance display before saving the termination and after saving it. However, the correct value is saved.
- This rule function does not consider eligibility status. The specified thresholds determine whether a month is considered or not. However, if the Time Account Type Eligibility Status is set to "No" for the rest of the accrual period due to termination or leave of absence, then the accruable end date is set to the date until which the employee is eligible.

Validation Checks

- The job information field ID must be a field in the data model, and it must be a numeric field (double, long, big decimal, salary).
- The start date must be earlier than or the same as the end date.
- The accruable start date must be earlier than or the same as the accruable end date, and the accruable period must fall within the accrual period.
- The accrual frequency for time account types must be set to *Annually* or *Monthly*.
- If the threshold is negative or null, the rule function won't work.
- The maximum value for each threshold is 31. Please keep the following behavior in mind:
  - If Threshold Start is 31, the rule function will include the month in which the user was hired.
  - If Threshold End is 31, the rule function will ignore the month in which the user was terminated.
  - If Threshold Data Change is 31, the old value of the job information field will be used for the calculation

Input Parameters

For this parameter	Make this entry
User	Select the user you want to calculate the average for.

For this parameter	Make this entry
Job Information Field ID	Enter the field ID from your data model. Type it in manually - **don't** navigate to the corresponding job information field. In the examples below the Job Information Field ID is mentioned as Entitlement Value. **Note** The field in your data model must be a numeric field in order to calculate an average. For example, *custom-long1* or *custom-double1*.
Start Date	Under *Manage Parameters* > *accrualRuleParameters* >, select *Start Date*.
End Date	Under *Manage Parameters* > *accrualRuleParameters* >, select *End Date*.
Time Account Type	Select the relevant time account type.
Accruable Start Date	Under *Manage Parameters* > *accrualRuleParameters* >, select *Accruable Start Date*.
Accruable End Date	Under *Manage Parameters* > *accrualRuleParameters* >, select *Accruable End Date*. You can also select the *Contract End Date* field in the Job Information as the *Accruable End Date*.
Calculation Period	Choose whether you want to calculate the average value calculation for the accrual period or for the accruable period (see example below).
Threshold Start	Month will be considered if hire date is on or before threshold.
Threshold End	Month will be considered if termination date is after the threshold.
Threshold Data Change	Month will be considered for new value if data change is on or before the threshold.
Exclude Contract End Date	This parameter is optional. It operates as a flag to decide whether the contract end date within the job info should be respected or not. The default value is false.

Now, let's have a look at the rule function in action. In this example, a hypothetical employee called Aanya Singh was hired on February 17, 2020, and has annual accruals with the following weekly hours and entitlements:

Effective Start Date	Effective End Date	Standard Weekly Hours	Entitlement Value	No. of Eligible Months in Accrual Period
17.02.2020 (hire date)	31.12.9999	40	25	10 (March to December, since Aanya was only hired in February)

Scenario #1: Calculation Based on Accrual Period

Here's what you would enter as the parameters for the rule function.

Parameter	Entry
User	Aanya Singh
Job Information Field ID	custom-double1
Start Date	01.01.2020
End Date	31.12.2020
Time Account Type	VACATION
Accruable Start Date	17.02.2020
Accruable End Date	31.12.2020
Calculation Period	Accrual period
Threshold Start	10
Threshold End	15
Threshold Data Change	14

The calculation takes place based on the following formula:

Number of eligible months in each job information/number of months in accrual period * job information field value.

In the case of the example above, that means you would get a result of **20.83** (10/12×25).

Now let's say Aanya's work schedule is changed several times over the course of the year. Her leave entitlements will change accordingly, and her accruals will need to be recalculated.

Change #1: Weekly hours changed from 40 to 35 on 16th June

Effective Start Date	Effective End Date	Standard Weekly Hours	Entitlement Value	No. of Eligible Months in Accrual Period
17.02.2020 (hire date)	15.06.2020	40	25	4 (March to June)
16.06.2020 (work schedule changed)	31.12.9999	35	22	6 (July to December)

Here's what you would enter as the parameters for the rule function.

Parameter	Entry
User	Aanya Singh
Job Information Field ID	custom-double1
Start Date	01.01.2020
End Date	31.12.2020
Time Account Type	VACATION

Parameter	Entry
Accruable Start Date	17.02.2020
Accruable End Date	31.12.2020
Calculation Period	Accrual period
Threshold Start	10
Threshold End	15
Threshold Data Change	14

Due to Aanya's changed working hours and entitlements, the result will now be **19.33** (4/12×25+6/12×22).

> **Note**
>
> Since *Threshold Data Change* is set to 14, and Aanya's work schedule was changed on 16th June, the entire month of June is counted with the old entitlement value of 25.

Change #2: Weekly hours changed from 35 to 30 on 10th August

Effective Start Date	Effective End Date	Standard Weekly Hours	Entitlement Value	No. of Eligible Months in Accrual Period
17.02.2020 (hire date)	15.06.2020	40	25	4 (March to June)
16.06.2020 (work schedule changed)	09.08.2020	35	22	1 (July)
10.08.2020 (work schedule changed)	31.12.9999	30	19	5 (August to December)

As before, you would enter these parameters for the rule function.

Parameter	Entry
User	Aanya Singh
Job Information Field ID	custom-double1
Start Date	01.01.2020
End Date	31.12.2020
Time Account Type	VACATION
Accruable Start Date	17.02.2020
Accruable End Date	31.12.2020
Calculation Period	Accrual period
Threshold Start	10
Threshold End	15
Threshold Data Change	14

But now the result will be **18.08** (4/12×25+1/12×22+5/12×19).

> **Note**
>
> Since *Threshold Data Change* is set to 14, and Aanya's work schedule was changed on 10 August, the month of August is **not** counted with the old entitlement value (25) but with the new value (19).

Change #3: Weekly hours changed from 30 to 40 on 18 November

Effective Start Date	Effective End Date	Standard Weekly Hours	Entitlement Value	No. of Eligible Months in Accrual Period
17.02.2020 (hire date)	15.06.2020	40	25	4 (March to June)
16.06.2020 (work schedule changed)	09.08.2020	35	22	1 (July)
10.08.2020 (work schedule changed)	17.11.2020	30	19	4 (August to November)
18.11.2020 (work schedule changed)	31.12.9999	40	25	1 (December)

As before, you would enter these parameters for the rule function.

Parameter	Entry
User	Aanya Singh
Job Information Field ID	custom-double1
Start Date	01.01.2020
End Date	31.12.2020
Time Account Type	VACATION
Accruable Start Date	17.02.2020
Accruable End Date	31.12.2020
Calculation Period	Accrual period
Threshold Start	10
Threshold End	15
Threshold Data Change	14

But now the result will be **18.58** (4/12×25+1/12×22+4/12×19+1/12×25).

> **Note**
>
> Since *Threshold Data Change* is set to 14, and Aanya's work schedule was changed on 18th November, the month of November is counted with the old entitlement value (19).

Change #4: Employee is terminated on 10 December

Effective Start Date	Effective End Date	Standard Weekly Hours	Entitlement Value	No. of Eligible Months in Accrual Period
17.02.2020 (hire date)	15.06.2020	40	25	4 (March to June)

Effective Start Date	Effective End Date	Standard Weekly Hours	Entitlement Value	No. of Eligible Months in Accrual Period
16.06.2020 (work schedule changed)	09.08.2020	35	22	1 (July)
10.08.2020 (work schedule changed)	17.11.2020	30	19	4 (August to November)
18.11.2020 (work schedule changed)	10.12.2020	40	25	0 (December is not considered)
11.12.2020 (termination)	31.12.9999	40	25	0

As before, here are the parameters for the rule function.

Parameter	Entry
User	Aanya Singh
Job Information Field ID	custom-double1
Start Date	01.01.2020
End Date	31.12.2020
Time Account Type	VACATION
Accruable Start Date	17.02.2020
Accruable End Date	10.12.2020
Calculation Period	Accrual period
Threshold Start	10
Threshold End	15
Threshold Data Change	14

And the final result is **16.5** (4/12×25+1/12×22+4/12×19).

> **Note**
>
> The accruable end date is 10th December. As such, December is excluded from the calculation because *Threshold End* is set to 15.

Scenario #2: Calculation Based on Accruable Period

Here we'll look at the same example - an employee called Aanya Singh who is hired on 17th February with the following weekly hours, leave entitlements, and accruals:

Effective Start Date	Effective End Date	Standard Weekly Hours	Entitlement Value	No. of Eligible Months in Accrual Period
17.02.2020 (hire date)	31.12.9999	40	25	10 (March to December, since Aanya was only hired in February)

But in this scenario you want to calculate by **accruable** period, so you enter the following parameters for the rule function:

Parameter	Entry
User	Aanya Singh
Job Information Field ID	custom-double1
Start Date	01.01.2020
End Date	31.12.2020
Time Account Type	VACATION
Accruable Start Date	17.02.2020
Accruable End Date	31.12.2020
Calculation Period	Accruable period
Threshold Start	10
Threshold End	15
Threshold Data Change	14

In this scenario, the formula used for the calculation is:

Number of eligible months in each job information period/number of months in accruable period * job information field value

As such, in Aanya's case the result would be **25** (10/10×25).

Now let's say the Aanya's work schedule is changed several times over the course of the year. Her leave entitlements will change accordingly, and her accruals will need to be recalculated.

Change #1: Weekly hours changed from 40 to 35 on 16 June

Effective Start Date	Effective End Date	Standard Weekly Hours	Entitlement Value	No. of Eligible Months in Accrual Period
17.02.2020 (hire date)	15.06.2020	40	25	4 (March to June)
16.06.2020 (work schedule changed)	31.12.9999	35	22	6 (July to December)

Here's what you would enter as the parameters for the rule function.

Parameter	Entry
User	Aanya Singh
Job Information Field ID	custom-double1

Parameter	Entry
Start Date	01.01.2020
End Date	31.12.2020
Time Account Type	VACATION
Accruable Start Date	17.02.2020
Accruable End Date	31.12.2020
Calculation Period	Accruable period
Threshold Start	10
Threshold End	15
Threshold Data Change	14

Due to Aanya's changed working hours and entitlements, the result will now be **23.2** (4/10×25+6/10×22).

> **Note**
>
> Since *Threshold Data Change* is set to 14, and Aanya's work schedule was changed on 16 June, the entire month of June is counted with the old entitlement value of 25.

Change #2: Weekly hours changed from 35 to 30 on 10 August

Effective Start Date	Effective End Date	Standard Weekly Hours	Entitlement Value	No. of Eligible Months in Accrual Period
17.02.2020 (hire date)	15.06.2020	40	25	4 (March to June)
16.06.2020 (work schedule changed)	09.08.2020	35	22	1 (July)
10.08.2020 (work schedule changed)	31.12.9999	30	19	5 (August to December)

As before, you would enter these parameters for the rule function.

Parameter	Entry
User	Aanya Singh
Job Information Field ID	custom-double1
Start Date	01.01.2020
End Date	31.12.2020
Time Account Type	VACATION
Accruable Start Date	17.02.2020
Accruable End Date	31.12.2020
Calculation Period	Accruable period
Threshold Start	10
Threshold End	15

Parameter	Entry
Threshold Data Change	14

But now the result will be **21.7** (4/10×25+1/10×22+5/10×19).

> **Note**
>
> Since *Threshold Data Change* is set to 14, and Aanya's work schedule was changed on 10 August, the month of August is **not** counted with the old entitlement value (25) but with the new value (19).

Change #3: Weekly hours changed from 30 to 40 on 18 November

Effective Start Date	Effective End Date	Standard Weekly Hours	Entitlement Value	No. of Eligible Months in Accrual Period
17.02.2020 (hire date)	15.06.2020	40	25	4 (March to June)
16.06.2020 (work schedule changed)	09.08.2020	35	22	1 (July)
10.08.2020 (work schedule changed)	17.11.2020	30	19	4 (August to November)
18.11.2020 (work schedule changed)	31.12.9999	40	25	1 (December)

As before, you would enter these parameters for the rule function.

Parameter	Entry
User	Aanya Singh
Job Information Field ID	custom-double1
Start Date	01.01.2020
End Date	31.12.2020
Time Account Type	VACATION
Accruable Start Date	17.02.2020
Accruable End Date	31.12.2020
Calculation Period	Accruable period
Threshold Start	10
Threshold End	15
Threshold Data Change	14

But now the result will be **22.3** (4/10×25+1/10×22+4/10×19+1/10×25).

> **Note**
>
> Since *Threshold Data Change* is set to 14, and Aanya's work schedule was changed on 18th November, the month of November is counted with the old entitlement value (19).

Change #4: Employee is terminated on 10th December

Effective Start Date	Effective End Date	Standard Weekly Hours	Entitlement Value	No. of Eligible Months in Accrual Period
17.02.2020 (hire date)	15.06.2020	40	25	4 (March to June)
16.06.2020 (work schedule changed)	09.08.2020	35	22	1 (July)
10.08.2020 (work schedule changed)	17.11.2020	30	19	4 (August to November)
18.11.2020 (work schedule changed)	09.12.2020	40	25	0 (December is not considered)
10.12.2020 (termination)	10.12.2020	40	25	0

As before, you would enter these parameters for the rule function.

Parameter	Entry
User	Aanya Singh
Job Information Field ID	custom-double1
Start Date	01.01.2020
End Date	31.12.2020
Time Account Type	VACATION
Accruable Start Date	17.02.2020
Accruable End Date	10.12.2020
Calculation Period	Accruable period
Threshold Start	10
Threshold End	15
Threshold Data Change	14

And the final result is **22** (4/9×25+1/9×22+4/9×19).

> **Note**
>
> The accruable end date is 10th December. As such, December is excluded from the calculation because *Threshold End* is set to 15.

Scenario #3: Calculation with Contract End Date

Now we'll look at the effect of the contract end date option. The scenario is that an employee is hired on 17th February and has annual accrual. Standard weekly hours are 40 and the custom job information field "Entitlement value" is set to 25 when the job information is saved. At first, no value is entered for the "Exclude Contract End Date" field.

Employee's accrual is recalculated due to the change of standard weekly hours and prorated according to the calculation below.:

The calculation is based on following formula: "Number of eligible months in each job information period/number of months in accrual period* job information field value".

Effective Start Date	Effective End Date	Standard Weekly Hours	Entitlement Value	Number of Eligible Months in Accrual Period
17.02.2020	31.12.9999	40	25	10 (March - December)

The number of eligible months is 10 because the employee was hired on 17th February, so January and February are excluded from the calculation.

Here are the parameters you enter for the rule function:

Parameter	Entry
User	Aanya Singh
Job Information Field ID	custom-double1
Start Date	01.01.2020
End Date	31.12.2020
Time Account Type	VACATION
Accruable Start Date	17.02.2020
Accruable End Date	31.12.2020
Calculation Period	Accrual period
Threshold Start	10
Threshold End	15
Threshold Data Change	14
Exclude Contract End Date (optional)	null

The result will be **20.83** (10/12*25).

Now, let's say the standard weekly hours are changed from 40 to 35 on 16th June and the leave entitlement value is changed to 22. The employee's accrual is recalculated due to the change of standard weekly hours and prorated according to the calculation below.

Effective Start Date	Effective End Date	Standard Weekly Hours	Entitlement Value	Number of Eligible Months in Accrual Period
17.02.2020 (Hire)	15.06.2020	40	25	4 (March-June)
16.06.2020 (Data Change)	31.12.9999	35	22	6 (July-December)

As before, you enter these parameters in the rule function.

Parameter	Entry
User	Aanya Singh
Job Information Field ID	custom-double1
Start Date	01.01.2020
End Date	31.12.2020
Time Account Type	VACATION
Accruable Start Date	17.02.2020
Accruable End Date	31.12.2020
Calculation Period	Accrual period
Threshold Start	10
Threshold End	15
Threshold Data Change	14
Exclude Contract End Date (optional)	null

This time the result will be **19.33** (4/12*25 + 6/12*22). As the data change threshold is 14 and the job information data was changed on the 16th, the old entitlement value (25) is used for June.

Now, we'll add 10th September as the contract end date. We then add the contract end date as the accrual end date, as this is what we recommend when you work with contract end dates. Otherwise, the rule function could not work correctly in case of "Accruable period" as a calculation period.

Effective Start Date	Effective End Date	Contract End Date	Standard Weekly Hours	Entitlement Value	Number of Eligible Months in Accrual Period
17.02.2020 (Hire)	15.06.2020	-	40	25	4 (March-June)
16.06.2020 (Data Change)	31.12.9999	10.09.2020	35	22	2 (July-August)

As the data change threshold is 15 and the contract end date is 10th September, September is not included in the calculation.

As before, you enter these parameters in the rule function.

Parameter	Entry
User	Aanya Singh
Job Information Field ID	custom-double1
Start Date	01.01.2020
End Date	31.12.2020
Time Account Type	VACATION
Accruable Start Date	17.02.2020
Accruable End Date	31.12.2020
Calculation Period	Accrual period

Parameter	Entry
Threshold Start	10
Threshold End	15
Threshold Data Change	14
Exclude Contract End Date (optional)	null

The result will be **12.0** (4/12*25+2/12*22).

Now, we'll change the Exclude Contract End Date parameter to true so that the contract end date is disregarded during calculation.

Effective Start Date	Effective End Date	Contract End Date	Standard Weekly Hours	Entitlement Value	Number of Eligible Months in Accrual Period
17.02.2020 (Hire)	15.06.2020	-	40	25	4 (March zo June)
16.06.2020 (Data Change)	31.12.9999	10.09.2020	35	22	6 (July to December)

There is a data change in June where the value of the hire Job Info is respected due to the data change being after the data change threshold. As there is no termination entry and the contract end date is disregarded, all months are eligible during the accrual period.

This time, you enter these parameters in the rule function.

Parameter	Entry
User	Aanya Singh
Job Information Field ID	custom-double1
Start Date	01.01.2020
End Date	31.12.2020
Time Account Type	VACATION
Accruable Start Date	17.02.2020
Accruable End Date	10.09.2020
Calculation Period	Accrual period
Threshold Start	10
Threshold End	15
Threshold Data Change	14
Exclude Contract End Date (optional)	true

The result will be **19.33** (4/12*25 + 6/12*22).

Scenario #4: Employee is Terminated and Rehired in the Same Year

Effective Start Date	Effective End Date	Standard Weekly Hours	Entitlement Value
01.01.2020 (hire date)	14.02.2020 (termination date)	40	30
15.02.2020	31.05.2020	0	15
01.06.2020 (rehire date)	31.12.9999	40	30

In this example the user is terminated on 15th February 2020. As part of termination end handling this would lead to a closure of the existing time account and an adjustment of the related accruable period as the the employment ends. After rehire the user would receive an additional time account starting 1st June 2020.

Therefore, two calculations are executed for the year 2020.The accruable period is adjusted based on the termination date and rehire date.

Here's what you would enter as the parameters for the rule function after rehire:

Parameter	Entry
User	Aanya Singh
Job Information Field ID	custom-double1
Start Date	01.01.2020
End Date	31.12.2020
Time Account Type	VACATION
Accruable Start Date	01.06.2020
Accruable End Date	31.12.2020
Calculation Period	Accrual period
Threshold Start	10
Threshold End	13
Threshold Data Change	15

With a rehire on 1st June, the user is hired before the Threshold Start and therefore June is eligible. The following months are also relevant (June to December = 7 months). As the months of the previous hire period (January + February) are outside of the accruable period, they are not considered.

The result will be **17.5** (30 * 7 / 12).

> ### Note
>
> The value outside the accruable period is not considered. This ensures that the accrual is only calculated for the current rehire period.

When calculating the accrual value for the previous employment, the parameters for the rule function are:

Parameter	Entry
User	Aanya Singh

Parameter	Entry
Job Information Field ID	custom-double1
Start Date	01.01.2020
End Date	31.12.2020
Time Account Type	VACATION
Accruable Start Date	01.01.2020
Accruable End Date	14.02.2020
Calculation Period	Accrual period
Threshold Start	10
Threshold End	13
Threshold Data Change	15

To decide whether the termination month is considered for the calculation at all, the threshold end value is relevant. In this case, the termination month is considered because the employee is terminated after the 13th. To determine now which entitlement value is considered for the calculation of the termination month, the threshold data change value is considered. In the example, the termination record in the job information starts 15.02.2020. As the threshold data change is 15, the entitlement value of the termination job info record is relevant for the calculation.

The result will be **3.75** (1/12 * 30 + 1/12 * 15).

16.12 Calculate Balance()

This rule function calculates the sum of all postings on that time account up to the calculation date passed to the rule function. Later postings are not considered.

Limitations

- Use this rule function only with Employee Central Time Off.
- This rule function does not consider accrual simulation even if it's activated for the time account type.

Input Parameters

For this parameter...	Make this entry:
Date	The date for which the balance should be calculated.

For this parameter...	Make this entry:
Time Account	You have to select the *Time Account* MDF object.

For full information on capping accruals, refer to the Implementing Time Management in SAP SuccessFactors guide under *Capping Accruals Based on Balance Limit*.

> ⚠ Restriction
>
> Do not use the Calculate Balance() rule function for a time account when creating a time account detail for that account, for example, to create the amount. If you do this, balance calculation will fail as the time account detail is already added to that account, but is empty.
>
> Instead, define the rule as follows:
>
> 1. Calculate the account and assign the result to a variable (SET-statement).
> 2. Then create the time account detail and assign the value calculated before.

Examples

These are the postings for the time account that is passed to the rule function:

Posting Date	Posting Type	Amount
Jan 1, 2022	Accrual	10 hours
Feb 1, 2022	Accrual	10 hours
Mar 1, 2022	Accrual	10 hours
May 15, 2022	Leave Booking	-8 hours

The Calculate Balance rule function results for that account look like this for different dates:

Rule Function Date	Rule Function Balance Result
Feb 20, 2022	20
Mar 20, 2022	30
May 20, 2022	22

Difference from Balance Calculation Considering Accrual Simulation

If accrual simulation is set to Yes for the time account type, the rule function balance result may differ from the balance displayed for the time account on the *Administer Time Workbench* and from the balance calculated when requesting absences.

Let´s assume that today is March 15 and the balance is calculated for May 20, 2022. With accrual simulation active, the system would generate additional simulated accruals for balance calculation. As a result, the balance displayed on the *Administer Time Workbench* for that time account is 42 instead of 22.

Posting Date	Posting Type	Amount	Simulated Accrual
Jan 1, 2022	Accrual	10 hours	
Feb 1, 2022	Accrual	10 hours	
Mar 1, 2022	Accrual	10 hours	
Apr 1, 2022	Accrual	10 hours	Yes
May 1, 2022	Accrual	10 hours	Yes
May 15, 2022	Leave Booking	-8 hours	

Using the Rule Function for Accrual Calculation Rules

In an accrual scenario, existing accruals for periods to be calculated are removed before calculating the accrual.

Let´s assume an accrual calendar with recalculation is executed for February 2022. Calculating the accrual on Feb 20, 2022 therefore returns a balance result of 10 hours instead of 20.

Posting Date	Posting Type	Amount
Jan 1, 2022	Accrual	10 hours
Feb 1, 2022	Accrual	10 hours
Mar 1, 2022	Accrual	10 hours
May 15, 2022	Leave Booking	-8 hours

When calculating accruals for multiple periods, the result of earlier periods is considered. For example, an accrual calendar is executed with recalculation for periods February and March. The system removes both existing accruals and the rule function calculates one after the other. When calculating the accrual for March, the accrual calculated for February is considered.

Related Information

Calculate Balance for Types() [page 141]

16.13 Calculate Balance for Types()

This rule function is an advanced version of the Calculate Balance() rule function. It calculates the balance based on a given time type or time account type for a target user on a specific date. It can be used in a Take rule, because the base object is EmployeeTime and you can pass the external code of the time type.

> ⚠ Caution
>
> Do not use this rule function in save rules in, for example, Time Account or Employee Time objects. Doing so can lead to high memory consumption during mass processing such as accrual or account calendar runs.

If multiple accounts are selected on that date, the balances of all accounts will be accumulated.

- For the parameter of external codes, more than one time (account) type is added.
- For a time account type there are two recurring time accounts bookable on the calculation date passed to the rule.
- There is more than one time account type assigned to the time type.

Similar to the Calculate Balance() rule function, the system considers all postings up to the date. However, when using the rule function based on time types there is an additional dependency on the value defined for time type field Balance Calculation Setting. If set to "Consider bookings after calculation date" the rule function also considers later postings.

Limitations

- This rule function does not consider accrual simulation even if it's activated for the time account type.
- This rule function calculates the balance based on the information stored in the database already. For example, when using the rule function in an absence validation rule, it does not consider postings of the employee time the user is about to create.

Input Parameters

For this parameter	Make this entry
Date	The date for which the balance should be calculated.
Selection Type	Please choose “Enum” and select between Time Type or Time Account Type.
User	The target user.
[List of] Time (Account) Type External Code	Depending on what you selected for *Selection Type*, enter an external code of either a time type or time account type. If you add more than one value to this list, the resulting balances will be accumulated

Examples

For the date and time (account) types passed to the rule, two time accounts are found considering the bookable period of the time accounts:

Time Account A

Posting Date	Posting Type	Amount
Jan 1, 2022	Accrual	10 hours

Posting Date	Posting Type	Amount
Feb 1, 2022	Accrual	10 hours
Mar 1, 2022	Accrual	10 hours
May 15, 2022	Leave Booking	-8 hours

Time Account B

Posting Date	Posting Type	Amount
Jan 1, 2022	Manual Adjustment	5 hours

The balance is calculated for every account and the rule function result is the total amount of both time accounts:

Rule Function Date	Rule Function Balance Result
Feb 20, 2022	25
Mar 20, 2022	35
May 20, 2022	27

Special Case:Time Type Balance with Balance Calculation Setting equal to "Consider bookings after calculation date"

Rule Function Date	Rule Function Balance Result
Feb 20, 2022	25
Mar 20, 2022	27
May 20, 2022	27

For the date Feb 20, 2022 the result is unchanged. The future absence does not reduce the balance because of the earlier future on Mar 1, 2022.

For the date Mar 20, 2022 the balance result is reduced to 27 because of the later leave posting.

Related Information

Calculate Balance() [page 139]

16.14 Calculate Duration in Seconds Between Two DateTime Values()

This rule function calculates the time in seconds that has elapsed between two UTC timestamps. Please note that if the first timestamp falls after the second, you will get a negative result.

Example

- *First DateTime*: 12/31/2016 08:00:00 UTC
- *Second DateTime*: 12/31/2016 16:30:00 UTC
- *Result*: 30,600 Seconds

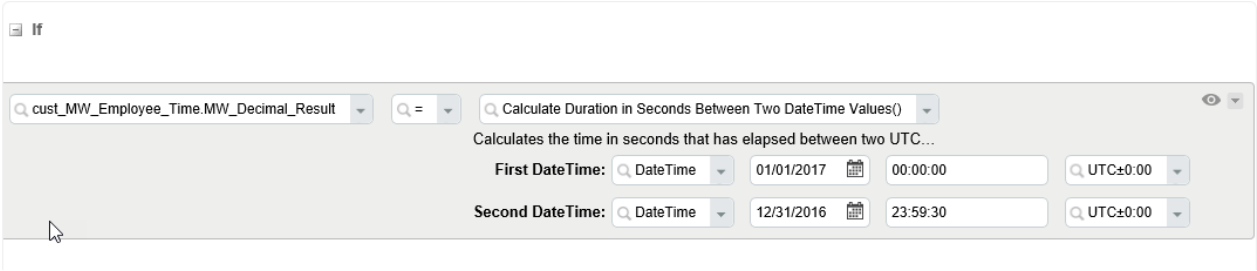
Example

- *First DateTime*: 01/01/2017 00:00:00 UTC
- *Second DateTime*: 12/31/2016 23:59:30 UTC
- *Result*: -30 Seconds

Input Parameters

Table 1

For this parameter...	Of type...	Which is...	Make this entry:
First DateTime	DateTime	Required	Enter the first timestamp.
Second DateTime	DateTime	Required	Enter the second timestamp.



16.15 Calculate Entitlement Balance()

This rule function is used to retrieve the entitlement balance from a given time account with entitlement method "Entitled as Transferred" or time type for a given date.

This is required for use cases where the accruals or entitlements need to be capped to a maximum value and the user needs to configure a rule based on current value of the entitlement balance. This rule function should return the entitlement balance that is available for use to the employee as of the given date.

The entitlement balance is calculated based on the following:

Entitlement balance = Sum of entitlement postings + sum of manual adjustments + sum of interim transfers – sum of payouts – sum of employee times postings (also in the future).

If both the time type and time account are passed, the time account takes precedence.

Input Parameters

For This Field	Make This Entry
User ID	Select the *User* field. This is the user for which the balance is calculated.
Date	Date for which the start time is required.
Time Type	Select the time type for which the balance is to be calculated.
Time Account	Select the time account for which the balance is to be calculated.
Simulate Accruals and Entitlements	Boolean indicator for Simulation

Examples

Let's look at an example.

The employee is hired on January 1, 2017, Accrual Frequency is Monthly and he accrues 1 day every month. His Next Transfer Date is on January 1, 2018.

When we call this rule function to retrieve the entitlement balance as of January 1, 2018, the rule function will return 12 days. For any date in 2017, the rule function would return 0 days, because the transfer happens on January 1, 2018.

⚠ Restriction

Do not use the Calculate Entitlement Balance() rule function for a time account when creating a time account detail for that account, for example, to create the amount. If you do this, balance calculation will fail as the time account detail is already added to that account, but is empty.

Instead, define the rule as follows:

1. Calculate the account and assign the result to a variable (SET-statement).
2. Then create the time account detail and assign the value calculated before.

16.16 Calculate FTE based on Standard Hours

With this function, you can extend the standard hours default cascading logic to include positions. This is used when the working standard hours differ on position level.

Limitations

- Use only with Position Management. Please refer to the Position Management Guide before using this function.
- The base object has to be Job Information.

Use Case

As this is an application-specific function, we recommend you stick to the use case defined in the Position Management Guide.

Related Information

Calculating Full-Time Equivalents

16.17 Cardinality

You can use this function to calculate the number of object instances found by a path. The path offers a collection filter that counts only the number of instances that match the filter condition.

Input Parameters

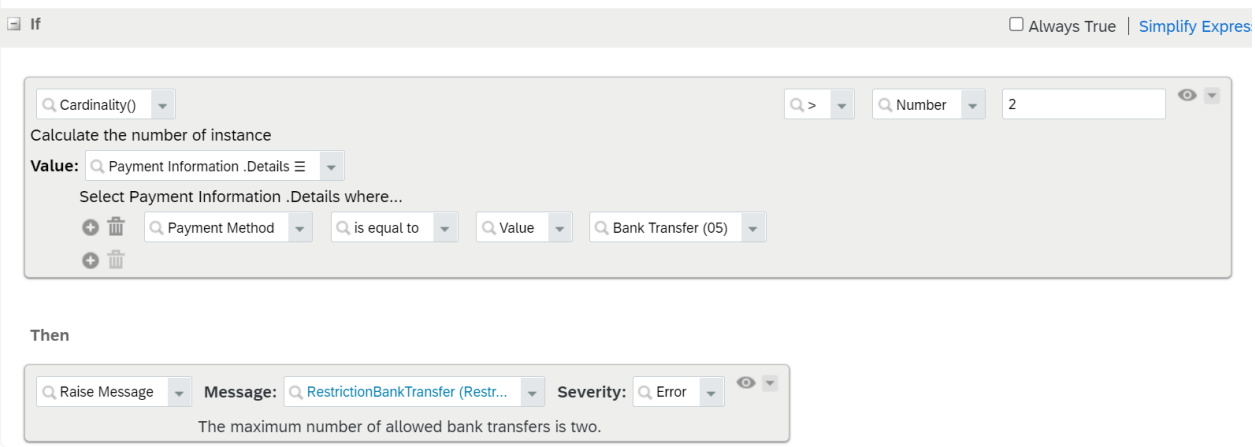
Input Parameter	Type	Required	Enter
Value	Collection	Yes	A path that ends with one of the following cardinalities: • *1:n* • *1:1:n* • *1:1:..:n* **Note** Any other cardinalities, such as *1:1* or *1:n:n*, aren`t supported.

Behavior in Case of Errors

If *Value* is empty or "null", the function returns the number zero.

Use Case

You want to restrict the allowed number of bank accounts that have a certain payment method (for example, bank transfer) to two.



16.18 Cap Accrual()

This function determines which accrual value can be posted in Time Off without exceeding the specified accrual balance limit.

For example, you determine that the maximum number of days that can be accrued is 2, while the account balance limit of 20 should not be exceeded. If the current account balance is 19, the rule function determines that the system cannot book an accrual of 2 days, but only 1 day, to guarantee that the account balance limit of 20 is not exceeded.

Input Parameters

For this parameter...	Make this entry:
Maximum Accrual	Enter the maximum accrual amount.
Account Balance	This is the account balance to a specific date. The account balance can be determined by an additional rule, Calculate Balance [page 139].
Account Balance Limit	Enter the account balance that should not be exceeded when the maximum accrual is added to the account balance.

16.19 Check If Retroactive Changes Allowed in Period

This function determines if a user is allowed to change objects based on whether time recording is allowed in the relevant period.

Configuration Requirements

You've enabled Time Tracking.

Input Parameters

Input Parameter	Type	Required	User Entry
User	User	Yes	Enter the ID of the user that you want to check.
Start Date	Date	Yes	Enter the start date of the period that you want to check.

Input Parameter	Type	Required	User Entry
End Date	Date	Yes	Enter the end date of the period that you want to check.

Behavior in Case of Errors

If any of the input parameters are "null" or inconsistent, the rule execution is stopped and an error message is shown.

Use Case

As a Time Tracking customer, you want to prevent employees and managers from making retroactive changes to attendances and allowances before the admissibility date to avoid payroll-relevant changes. However, you want to allow HR admins to make such changes in exceptional cases. You implement this use case by creating a rule on the `Employee Time Sheet Entry` or `Allowance` MDF object and checking if the output parameters are *No*. If the output parameters are *No*, you raise an error or a warning.

Similarly, you want to prevent employees and managers from making payroll-relevant retroactive changes to absences of a certain type before the admissibility date. However, you want them to be able to record other absence time types such as sickness in exceptional cases. You implement this use case by creating a rule for absence validation and checking if the output parameters are *No*. If the output parameters are *No*, you raise an error or a warning.

16.20 Concatenate

This function combines strings into one string.

Input Parameters

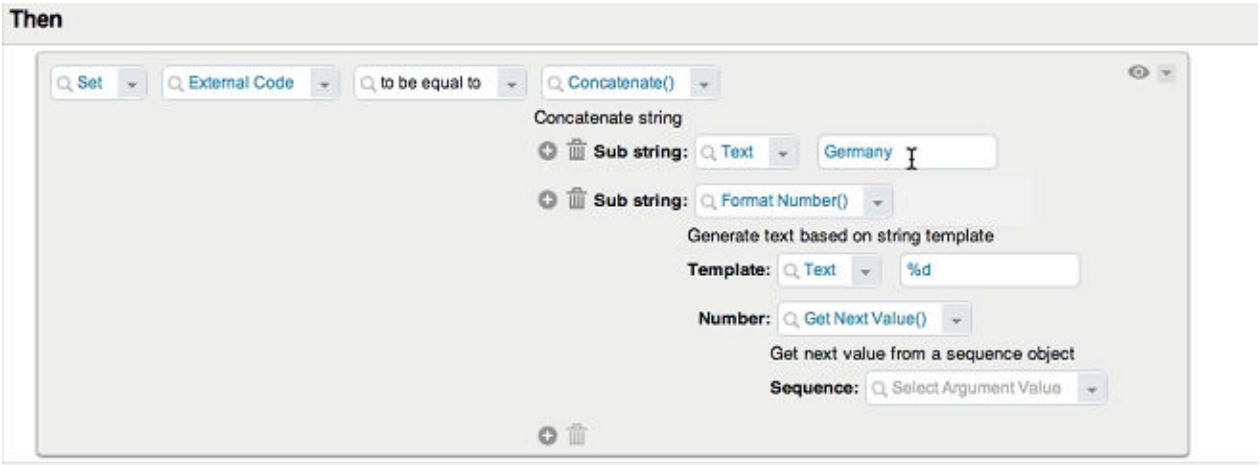
For this parameter...	Make this entry:
Substring	You can: • Select *Text* and enter a string. • Select a field of type "String". For example: *Job Code*. **Note** You need to add at least two substrings. It is also possible to add more.

Example

You want the external code to follow this format:

Germany1000, Germany1001, Germany1002, and so on.

You set up the rule as follows:



Use Case

You can use this function to generate external codes automatically.

Behavior in Case of Errors

- If any substring has the value "null", it is ignored.
- If all substrings have the value "null", the function returns an empty string.

16.21 Condense

This function removes consecutive whitepaces within a text string or replaces them with a single blank space. It also automatically removes any leading or trailing whitespaces from the string.

Sometimes users enter whitespaces (for example, blank spaces or tabs) unintentionally when creating a text string. This can cause problems when saving the input text. For example, you might end up with duplicate entries called “RoleName” and “Role<whitespace>Name”.

Input Parameters

Parameter	Type	Required	User Entry
Text	Text	Yes	The text string you want to condense.
Condense Mode	Value	No	One of the following options: • *Replace consecutive whitespaces with one single blank space* • *Remove all whitespaces* If you leave this field empty or select *Null*, the function uses the first option as default (*Replace consecutive whitespaces with one single blank space*).

Behavior in Case of Errors

If *Text* is "null", the result of the function is "null".

> **Note**
>
> Please note that "null" doesn't mean the number zero in the context of business rules. Instead, it represents a missing value. A field with a null value is a field with no value.

Example 1:

Original text string: "Role<internal consecutive whitespaces>Name"

Text string after running the Condense function to replace the middle whitespaces with a single blank space: “Role Name”

Example 2:

Original text string: "Role<internal consecutive whitespaces>Name"

Text string after running the Condense function to remove all whitespaces: “RoleName”

16.22 Convert Days To YY/MM/DD

With this function, you can enter a number of days that is converted into a more readable format of years/months/days.

Please note the following:

- 30.4375 days are calculated as one month.
- 365 and 366 days are calculated as one year.
  For example, when you have a sum of 1090 days, the system subtracts 365 days, and again 365 days, to get the number of years. If the rest is less than 365, but still larger than 360, it is calculated as one year (because 360 days divided by 30 days each month = 12 months, which is one year).

Example

Number of Days	Result
360	1/0/0
370	1/0/5
360+365=725	2/0/0
250	0/8/10

Use Case

Some of the available functions return a result in days (refer to the examples for the Get Work History functions). An employee's length of service can easily go into the thousands of days. To have a more user-friendly format, this

function can be used to convert the number of days into the number of years, months, and days, which is much easier and faster for users to read.

16.23 Count Number of Leave of Absence Days

This function calculates the number of days for a given period of time. For example, you could count the total number of days in a leave of absence (LOA).

> **Note**
>
> All days are counted, not just work days.

Leave of absence days are determined in the *Job Information*. For each leave of absence, a job information record is created (no matter whether created in Time Off or in the Leave of Absence). The LOA event reason sets the field event. To end a LOA, another job information record is created (with event reason: End of LOA), which delimits the LOA. Each day in the job info that points to the event LOA should be counted.

Input Parameters

For This Field	Make This Entry
Begin Date	Period start date
End Date	Period end date

Example

You can use this function when running automated grade step progression processes. Typically the next upgrade date is calculated based on offsets (typically the multiple of years) from specific dates for example, service date, hire dates, and so on. So an employee's upgrade to the next pay scale level takes place every year, for example, if the hire date was on September 1, 2015, then the first level-upgrade is on September 1, 2016.

This cycle is only changed if the employee had any leave of absence (LOA) in the period between the upgrades. If the employee was out from December 1, 2015 until March 31, 2016, then the absence days should be added to the calculation, meaning that the upgrade should be postponed to January 1, 2017 (September 1, 2016 + number of LOA days).

16.24 Create Benefit Tracker for Employment Info Changes

This function is used to create a record with user ID in the Benefits Employee Master Data Change Tracker object. This function must be used on an object or entity which is not effective dated and has a user ID. For example, Employment Details.

Configuration Requirements

You can use this function when benefits management is enabled in Provisioning.

> → Remember
>
> As a customer, you don't have access to Provisioning. To complete tasks in Provisioning, contact your implementation partner or Account Executive. For any non-implementation tasks, contact Technical Support.

Input Parameters

Input Parameter	Type	Required	User Entry
User ID	Text	Yes	Enter the user ID for the user whose master data changes have to be tracked.

Behavior in Case of Errors

If an incorrect user ID is provided as input, no records are created in the *Benefit Employee Master Data Change Tracker* object.

Use Case

The administrator wants to create a record with user ID in the *Benefit Employee Master Data Change Tracker* object when there is any change in the master data for non-effective dated objects such as Employment Details.

> ⓘ Note
>
> This procedure is used only if you use basic rules. For rule scenarios, please refer to the topic in the *Related Information* section.

For example, when a user updates the Seniority Start Date field in the Employment Details object, the rule attached to the Employment Details object is executed and a record is created in the *Benefit Employee Master Data Change Tracker* object.

Related Information

Creating Automatic Enrollment Job for Employee Master Data Changes

16.25 Create Benefit Tracker for Employment Info Changes with Effective Date

This function is used to create a record with effective date and user ID into the Benefits Employee Master Data Change Tracker object. This function must be used on an object or entity that is effective dated and has a user ID. For example, Job Information.

Configuration Requirements

You can use this function when benefits management is enabled in Provisioning.

> → Remember
>
> As a customer, you don't have access to Provisioning. To complete tasks in Provisioning, contact your implementation partner or Account Executive. For any non-implementation tasks, contact Technical Support.

Input Parameters

Input Parameter	Type	Required	User Entry
User ID	Text	Yes	Enter the user ID for the user whose master data changes have to be tracked.
Effective Date	Date	Yes	Enter the effective date of the master data change.

Behavior in Case of Errors

If incorrect user ID is provided as input, no records are created in the *Benefit Employee Master Data Change Tracker* object.

Use Case

The administrator wants to create a record with user ID and effective date in the Benefit Employee Master Data Change Tracker object when there is any change in the master data for effective-dated objects such as Job Information.

> ⓘ Note
>
> This procedure is used only if you use basic rules. For rule scenarios, please refer to the topic in the *Related Information* section.

For example, a user updates the job location in Job Information, the rule attached to the Job Information object is executed and a record is created in the *Benefit Employee Master Data Change Tracker* object with the user ID and effective date.

Related Information

Creating Automatic Enrollment Job for Employee Master Data Changes

16.26 Create Benefit Tracker for Personal Info Changes

This function is used to create a record with person ID in the Benefits Employee Master Data Change Tracker object. This function must be used on an object or entity that is not effective dated and has a person ID. For example: Biographical Information.

Configuration Requirements

You can use this function when benefits management is enabled in Provisioning.

> → Remember
>
> As a customer, you don't have access to Provisioning. To complete tasks in Provisioning, contact your implementation partner or Account Executive. For any non-implementation tasks, contact Technical Support.

Input Parameters

Input Parameter	Type	Required	User Entry
Person ID	Text	Yes	Enter the person ID for the person for whom the master data changes have to be tracked.

Behavior in Case of Errors

If incorrect person ID is provided as input, no records are created in the *Benefit Employee Master Data Change Tracker* object.

Use Case

The administrator wants to create a record with person ID in the *Benefit Employee Master Data Change Tracker* object when there is any change in the master data for non-effective dated objects such as Biographical Information.

> **Note**
>
> This procedure is used only if you use basic rules. For rule scenarios, please refer to the topic in the *Related Information* section.

For example, when a user updates the *Country of Birth* field in *Biographical Information* object, the rule attached to the object is executed and a record is created in the *Benefit Employee Master Data Change Tracker* with a person ID.:

Related Information

Creating Automatic Enrollment Job for Employee Master Data Changes

16.27 Create Benefit Tracker for Personal Info Changes with Effective Date

This function is used to create a record with an effective date and a person ID in the *Benefits Employee Master Data Change Tracker* object. This function must be used on an object or entity that is effective dated and has a person ID. For example: Personal Information, Dependent Information.

Configuration Requirements

You can use this function when benefits management is enabled in Provisioning.

> → Remember
>
> As a customer, you don't have access to Provisioning. To complete tasks in Provisioning, contact your implementation partner or Account Executive. For any non-implementation tasks, contact Technical Support.

Input Parameters

Input Parameter	Type	Required	User Entry
Person ID	Text	Yes	Enter the person ID for the person for which the master data changes have to be tracked.
Effective Date	Date	Yes	Enter the effective date of the master data change.

Behavior in Case of Errors

If incorrect person ID is provided as input, no records are created in the *Benefit Employee Master Data Change Tracker* object.

Use Case

The administrator wants to create a record with person ID and effective date in the *Benefit Employee Master Data Change Tracker* object when there is any change in the master data for effective dated objects such as Dependent Information.

> **Note**
>
> This procedure is used only if you use basic rules. For rule scenarios, please refer to the topic in the *Related Information* section.

For example, when a user adds a new dependent in the Dependent Information, the rule attached to the Dependent Information object executed and a record is created in the *Benefit Employee Master Data Change Tracker* object with person ID and effective date.

Related Information

Creating Automatic Enrollment Job for Employee Master Data Changes

16.28 Create Date

This function creates a date for a given day, month, and year.

Example

Day: 10

Month: 6

Year: 2014

Result: June 10, 2014

16.29 Create DateTime()

This rule function returns a UTC timestamp created with a date, time, and time zone offset.

Input Parameters

Input Parameter	Type	Required	User Entry
Local Date	Date	Yes	Enter the local date.

Input Parameter	Type	Required	User Entry
Local Time	Time	Yes	Enter the local time.
Offset of Local Time Zone from UTC in Hours	Decimal	No	Enter the offset of the local time zone from UTC in hours. Any decimal value is allowed. If you leave the field empty or it returns a "null" value, the system uses the UTC time zone.

Behavior in Case of Errors

If local date or local time are "null", the rule stops with status FAIL.

> **Note**
>
> Please note that "null" does not mean the number zero in the context of business rules. Instead, it represents a missing value. A field with a null value is a field with no value.

Example

- Local date: *12/31/2016*
- Local time: *14:00:00*
- Offset of local time zone from UTC in hours: *1.0*

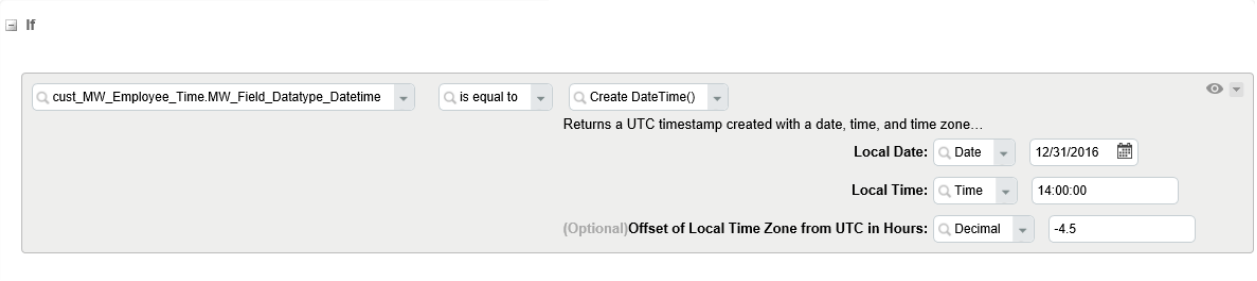
Result: **12/31/2016 13:00:00 UTC**

Example

- Local date: *12/31/2016*
- Local time: *14:00:00*
- Offset of local time zone from UTC in hours: *-4.5*

Result:**12/31/2016 18:30:00 UTC**

This is an example of what this could look like in the *If* section of a rule:



16.30 Create Time

This function returns a time created with a given hour, minute, and second.

Parameters

Name	Type	Mandatory	Description
Hour	Integer	Yes	The hour of the return time.
Minute	Integer	Yes	The minute of the return time.
Second	Integer	Yes	The second of the return time.

Example

Length	Result
createTime(11,11,11)	11:11:11
createTime(26,11,11)	Invalid
createTime(11,null,11)	Invalid

16.31 Currency from Pay Scale Structure

This function gets the currency from the pay scale structure based on the pay scale level, pay component, and effective date. The pay scale structure is configured in the MDF object *Pay Scale Level*.

Limitations

Use only if the *Pay Scale Group* and *Pay Scale Level* fields are included in the job information.

For more information about how to set up pay scale structures, see the SuccessFactors Employee Central Payroll Integration Guide.

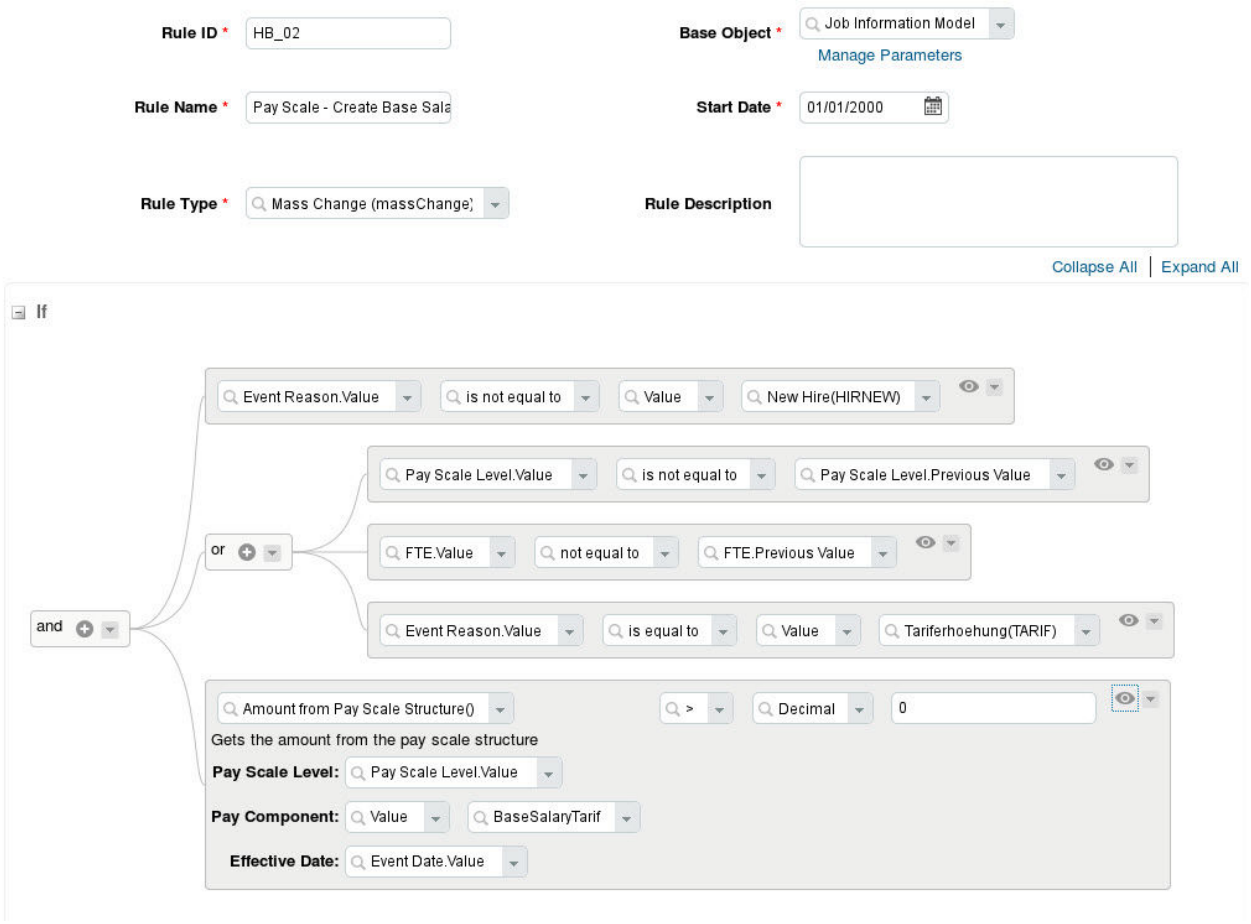
Input Parameters

For this parameter...	Of type ...	Which is...	Make this entry:
Pay Scale Level	Text	Required	Select the pay scale level you want to use. This is usually the value which is entered in the job information.
Pay Component	Value	Required	Select the *Value* field type, and select the pay component you want to search for in the pay scale structure. Usually, this is the pay component you want to create in the compensation model.
Effective Date	Date	Required	Enter or select the effective date you want to use to read the data from the pay scale structure. Usually, this is the event date of the job information.

Use Case

You can create rules that automatically assign the correct pay components to an employee based on the pay scale level and pay scale group you define for that employee.

Here's an example of what such a rule could look like – for the complete process of how to set up such rules, please refer to the SuccessFactors Employee Central Payroll Integration Guide:



Related Information

Amount from Pay Scale Structure [page 110]
Frequency from Pay Scale Structure [page 177]

16.32 Date Plus

With this function, you add or subtract a specified number of days or months to or from a given date.

Input Parameters

Input Parameter	Type	Required	User Entry
Base Date	Date	Yes	Select or enter a date that is the basis for the calculation. The number of days or months you enter in the following fields will be added to or subtracted from this date.
Number of Months	Number	No	Enter the number of months that you want to add to or subtract from the base date. You can enter: • A positive value for adding months • A negative value for subtracting months • 0 if the field is not applicable for calculation If you leave this field empty or select *Null*, the function uses the number zero.
Number of Days	Number	No	Enter the number of days that you want to add to or subtract from the base date. You can enter: • A positive value for adding days • A negative value for subtracting days • 0 if the field is not applicable for calculation If you leave this field empty or select *Null*, the function uses the number zero.

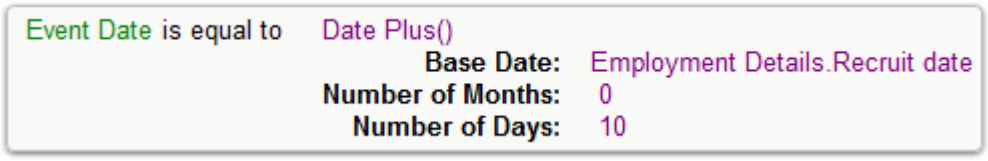
Behavior in Case of Errors

If *Base Date* is "null", the result of the function is "null".

> **Note**
>
> Please note that "null" doesn't mean the number zero in the context of business rules. Instead, it represents a missing value. A field with a null value is a field with no value.

Example

In this example, you add 10 days to the recruitment date:



In this example, you subtract 10 days from the recruitment date:

Use Case

You want the manager and employee to be reminded of the approaching end of a global assignment. To achieve this, you create a rule that uses the *Date Plus* function, which calculates the reminder date to be one month before the end of the global assignment. The end of the global assignment is represented by the end date of the Job Information record. You use approval workflows in Employee Central to define who gets the alerts and notifications.

You can find an example for creating alert triggering rules in the Employee Central Workflows guide.

Related Information

Creating an Alert Triggering Rule for Employee Central Objects

16.33 Day Of Month

This function determines the day of the month for a given date.

The default is for the current date to be used if NULL is passed as a parameter.

Example

Day of Month: June 10, 2021

Result: 10.

16.34 Day Of Week

This function determines the number of the weekday for a given date. 1 stands for Monday, 7 for Sunday.

Example

Day of Week: Jan 10, 2014

Result: 5 (=Friday)

The default is for the current date to be used if NULL is passed as a parameter.

16.35 Difference In Calendar Years

This function calculates the difference between two dates in calendar years. The system only takes account of the calendar year in the start and end dates, not the days or months.

Example

In the following two examples, the exact time difference is 1.5 years for each. However, the result shown is different as the system only takes into account the calendar years.

From Date	To Date	Result (in Calendar Years)	Details
12/22/2010	01/01/2012	2	The function calculates 2012-2010=2.
01/01/2010	09/15/2011	1	The function calculates 2011-2010=1.

Use Case

You can use this function to calculate seniority. You can find an example rule in the Time Off Guide, under *Accruals Based on Seniority*.

16.36 Difference In Years Round Down

This function calculates the time difference between two dates in calendar years. The system also takes into account the days and month of the date for the calculation. The result is a number that is rounded down to calendar years.

Example

Start Date: 12/31/2010

End Date: 1/9/2012

Result: 1 (calendar year)

Explanation: The exact time difference is 1 year, 9 days. The function rounds down to 1.

Use Case

You can calculate an employee's age based on the employee's birth date, rounding down the result.

For an example rule, see the examples in the Employee Central Implementation Guide, under *Setting up configurable rules*.

16.37 Difference In Years Round Up

This function calculates the time difference between two dates in calendar years. The system also takes into account the days and months of the date for the calculation. The result is a number that is rounded up to calendar years.

Example

From Date: 12/31/2010

To Date: 01/09/2012

Result: 2 (calendar years)

Explanation: The exact time difference is 1 year, 9 days. The function rounds up to 2.

If the time difference is, for example, 3 months, the function rounds up to 1.

16.38 Digitsum

Returns the sum of digits in the number. Each digit can be assigned a weight.

The syntax looks like this:

digitsum(long number [, int[] weight])

Limitations

- Only non-negative numbers and weights are supported.
- If number of weights (m) specified is less than number of digits (n) in the number, take 1 as weight for the remaining digits.
- If number of weights (m) specified is greater than the number of digits (n) in the number, take first n weights for calculation.

Input Parameters

Name	Type	Mandatory	Description
number	Long	Yes	The number the sum of whose digits you want.
weight	int[]	No	Weight for each digit in the number, by default 1.

Example

Length	Result
digitsum(123, 2, 5, 4)	24
digitsum(123, 2, 5)	15
digitsum(123, 2, 5, 4, 8)	24
digitsum(123)	6

16.39 Divide

This function divides values.

Limitations

You can only use the US format for decimals (using dots, not commas).

Input Parameters

Input Parameter	Type	Required	User Entry
Dividend	Decimal	Yes	Enter or select the number being divided.
Divisor	Decimal	Yes	Enter or select the number by which the dividend is divided. **Note** The divisor can't be zero.

Behavior in Case of Errors

- If dividend or divisor are "null", the function returns "null".
- If the divisor is or returns zero, the rule stops with status FAIL.

Note

Please note that "null" does not mean the number zero in the context of business rules. Instead, it represents a missing value. A field with a null value is a field with no value.

16.40 Format

You use this function to combine different objects into a meaningful text. For example, you can use this function to combine the country code (for example, DE), business unit name (for example, SMB), and a number (for example, 111) to one text: DESMB111.

In comparison with the *Format Number* function, you can select more field types and more arguments as input. In both functions, you can use Java string format specifiers that are replaced during runtime with the corresponding value.

Input Parameters

Input Parameter	Type	Required	User Entry
Template	Text	Yes	You can either: • Select any field of the field type *Text*, for example: *custom_field1*, where you use the custom field to store a template or pattern. • Select the field type *Text* and enter a string. You can use Java string format specifiers that are replaced during runtime with the corresponding value. For example: **`Company%s`**, where **`%s`** is replaced by what you have selected in the *Argument* field. **Note** Note the following: • For fields of type *Date*, use **`%t`** as the Java string format specifier. See examples below. • For fields of type *Decimal*, use **`%f`** as the Java string format specifier. If you use **`%.2f`**, for example, keep in mind that the number will be implicitly rounded dependent on the actual type of the number field (for example,

Input Parameter	Type	Required	User Entry
			1.9999 will be converted to 2.00 instead of 1.99). See also the function Format Number [page 175]. For all other field types, you can use **%s**. You can find more information on Java string format specifiers and Java string format examples on the internet.
Argument	Boolean Date Decimal Number Text	Yes	You can do one of the following: • Enter or select any field of the supported field types, for example: *Country*. • Select a function. For example, you can select the *Get Next Value* function and choose a Sequence Metadata Framework object you've already created. See *Get Next Value* for more information on how to create a Sequence Metadata Framework object. *Argument* is a multi-value parameter: For each format specifier defined in the template, you need to define an argument. If several specifiers refer to the same argument, you need to add a numbered dollar sign (1$, 2$, 3$ and so on) to the corresponding specifier in the *Template* field to refer to the first, second, third argument (for example, %$1s, %$2d, and so on).

Behavior in Case of Errors

If *Template* is "null", the function returns "null".

If *Argument* is "null", the Java specifier is replaced with the text *null* (for example, *DESMBnull*).

> **Note**
>
> Please note that "null" does not mean the number zero in the context of business rules. Instead, it represents a missing value. A field with a null value is a field with no value.

Example

You want to define a specific date format to get back 2021-06/1 as result. You can use Java formatters to achieve this goal.

> **Note**
>
> Java formatting is defined as follows:
>
> %[argument_index$][flags][width][.precision]conversion
>
> You can find more information about formatters in the Java documentation on the internet.

This is what you define in the function:

Template: %1$tY-%1$tm/%1$te

Argument: 01/06/2021 (or the object selected that returns this date)

The Java formatters used have the following meaning:

- 1$ stands for the first argument, which is a date in our example

  > **Note**
  >
  > You only need one argument as all specifiers refer to the same argument, which is identified by the numbered dollar sign 1$.

- tY stands for year
- tm stands for month
- te stands for day formatted as two digits without leading zero, that is, month days 1-31

Use Case

You can generate a unique ID for an employee to exchange data with an external payroll system. The unique ID is a combination of the country code and the employee ID, and this unique ID is then stored in a custom field, for example, custom_string5. Here's an example of what the corresponding rule can look like:

If

(Always true)

Then

Set custom_string5 *to be equal to Format* (*Template: Text* '%s-%s', *Argument*: country_code, *Argument*: employee_id)

16.41 Format Date for Position to Job Requisition Mapping

This function helps you convert date format to the format recognized by Requisition that is yyyy-MM-dd HH:mm:ss.

Configuration Requirements

- You can use this function when Position Management is enabled in Provisioning.
- You can use this function when Use Recruiting Management Integration is enabled from Position Management settings in Admin Center.
- You can use this function when SuccessFactors API integration is OFF in Provisioning.

> → Remember
>
> As a customer, you don't have access to Provisioning. To complete tasks in Provisioning, contact your implementation partner or Account Executive. For any non-implementation tasks, contact Technical Support.

Input Parameters

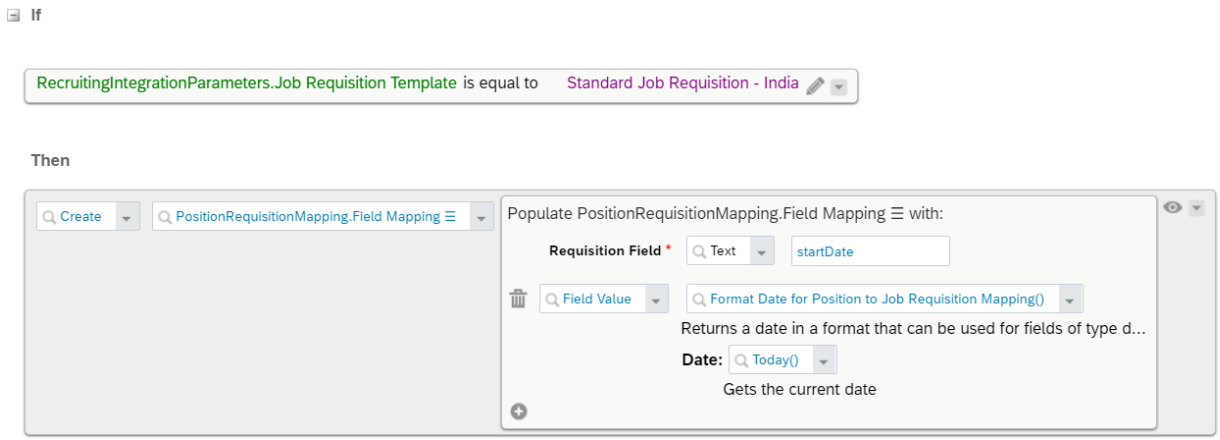
Input Parameter	Type	Required	User Entry
Date	Date	Yes	Select a date field or a date function, for example, Today().

Example

For example, if you input date as dd-mmm-yyyy HH:mm:ss (24 Nov 2020 12:00:00), the date is converted to yyyy-MM-dd HH:mm:ss (2020-11-24 12:00:00).

Use Case

For example, if the input to startDate field is driven from Today () function that has the format dd-mmm-yyyy HH:mm:ss, you can convert to yyyy-MM-dd HH:mm:ss using Format Date for Position to Job Requisition Mapping rule function.



16.42 Format Number

With this function, you can define a string template that the system uses to generate a text. For example, you can define that the system automatically assigns external codes following the format you define here.

Input Parameters

Input Parameter	Type	Required	User Entry
Template	Text	Yes	You define a string with the Java format specifier "%d" at any position in this string.
Number	Number	Yes	You can enter or select a number.

Behavior in Case of Errors

- If *Template* is “null”, the function returns “null”.

> **Note**
>
> Please note that "null" does not mean the number zero in the context of business rules. Instead, it represents a missing value. A field with a null value is a field with no value.

- If *Number* is "null", the Java specifier %d is replaced with the text *null* (for example, *POSnull*).
- If you use decimal numbers with the Java specifier %d (for example, %50.2d), you get an error.
- Precision for the Java format specifier isn't supported (for example, %.2d).

> **Note**
>
> Java formatting is defined as follows:
>
> %[argument_index$][flags][width][.precision]conversion
>
> You can find more information about formatters in the Java documentation on the internet.

Example

You can define that job titles or positions follow a simple sequence:

POS1, POS2, POS3...

You do this by creating a template for the text that defines how the system should define new positions. Here's an example where you use a Java variable (%d):

Template: Text **POS%d**

%d will be replaced with a number that follows the sequence you have defined in the Sequence object you have selected for the *Get Next Value* function in the *Number* field. For more information, see Get Next Value [page 256].

If you want to add leading zeros, follow this format:

```
POS%05d
```

The result is POS00001, with 4 leading zeros before the number "1". So the number is represented as a 5-digit number. If the number has more digits than given in the format, no leading zeros are displayed.

Use Case

You can define that the system automatically creates external codes, for example, for foundation objects, or position codes. You can find more information in the corresponding application-specific documentation.

Related Information

Generate Position Code Automatically

Example: Automatic Generation of Legacy Foundation Object Codes

16.43 Frequency from Pay Scale Structure

This function gets the frequency from the pay scale structure based on the pay scale level, pay component, and effective date. The pay scale structure is configured in the Metadata Framework object *Pay Scale Level*.

Configuration Requirements

The *Pay Scale Group* and *Pay Scale Level* fields need to be included in the job information.

For more information about how to set up pay scale structures, please refer to the Implementing Employee Compensation Data in Employee Central guide.

Input Parameters

Input Parameter	Type	Required	User Entry
Pay Scale Level	Text	Yes	Select the pay scale level you want to use. This is usually the value that is entered in the job information.
Pay Component	Value	Yes	Select the *Value* field type, and select the pay component you want to search for in the pay scale structure. Usually, this is the pay component you want to create in the compensation model.
Effective Date	Date	Yes	Enter or select the effective date you want to use to read the data from the pay scale structure. Usually, this is the event date of the job information.

Use Case

You can create rules for indirect valuations. You can find rule examples in the Implementing Employee Compensation Data in Employee Central guide.

Related Information

Indirect Valuation for New Hire with Multiple Rules

Amount from Pay Scale Structure [page 110]

Currency from Pay Scale Structure [page 162]

16.44 Generate External Code For Time Off()

This function generates a unique UUID which can be used as external code. The UUID is a 32-character long combination of numbers and letters.

Note: The system does **not** check whether this UUID already exists in the system, which could be the case when a user has manually created an external code.

Limitations

Use only with Employee Central Time Off.

16.45 Generate Unique Identifier

This function generates a unique identifier (ID), generally for integration purposes. The identifier is a 32-character long combination of numbers and letters.

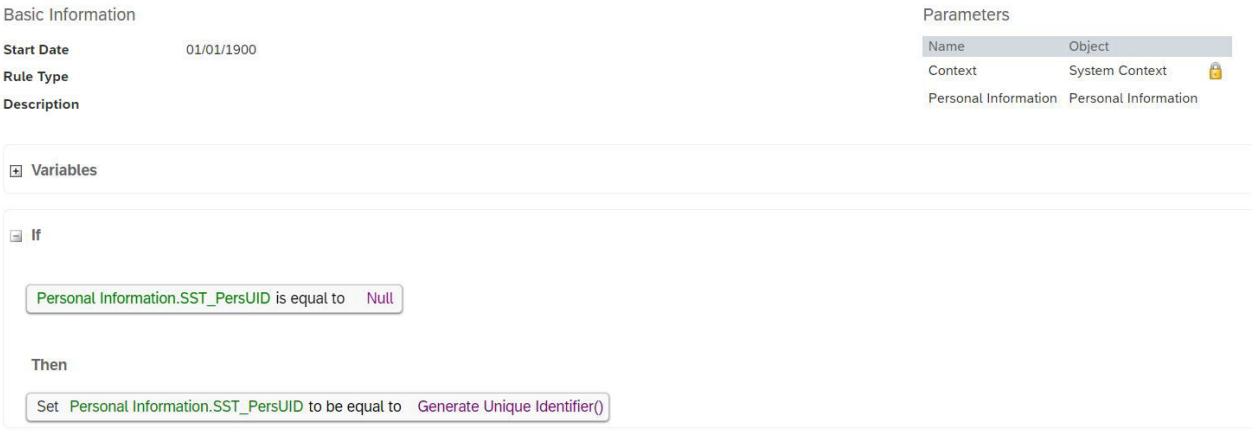
> ⓘ Note
>
> The system doesn't check whether this ID already exists in the system, which could be the case when a user has manually created an ID.

Input Parameters

There are no input parameters for this function.

Use Case

You want to add a unique ID for each employee when the employee's personal information is created or changed. You create a custom field in the Personal Information object and add an onSave rule to it. The rule automatically creates a unique ID if the custom field in the Personal Information is empty.



16.46 Get Absence Days()

With this function, you can calculate how long an employee has been absent within one year for a specific time type. For example, if an employee has been sick for more than 30 days in a year, you can use this function to raise an alert to the manager. The result of this function is the accumulated absence for that employee for a specific time type – in this example, 30 days for the time type Sickness.

The calculation considers work schedule and holidays if *Use Working Days Instead of Calendar Days* parameter is set to "Yes". This rule function considers the *Approved*, *Pending*, and *Pending Cancellation* Employee Time statuses for calculating absence days or hours. For cross-midnight absences, the absence is considered based on the working day to which the absence belongs.

Limitations

Use this function only with base object *Employee Time*.

Input Parameters

For this parameter...	Make this entry:
User	Select the *User* field. This is the user that creates the absence request in the Time Off calendar.
Time Type External Code	Select *Time Type.External Code*. This defines which time type has been used for the absence request (for example, Sickness).

For this parameter...	Make this entry:
Use Working Days Instead of Calendar Days	Select *Boolean*. You can then enter the following: • *Yes* to define that working days as defined in the working schedule should be used for calculating the absence. • *No* to use calendar days for calculating the absence.
Start Date	Select *Start Date*. This is the start date the user has entered in the Time Off calendar.
First Day of Year	Select *Number*, and enter a number to define the beginning of the calendar year you want to use to sum up an employee's absence days. For example, **1** to start with the first day of a month.
First Month of Year	Select *Number*, and enter a number to define the beginning of the calendar year you want to use to sum up an employee's absence days. For example, **1** to start with January, or **4** to start with April. The end date of the calendar year is then defined by the system accordingly (for example, December 31 when the start of the calendar year has been defined as January 01).

Use Case

Because this is an application-specific rule function, we recommend that you stick to the use case of Employee Central Time Off.

Please refer to the *Implementing Employee Central Time Off* guide for more information.

16.47 Get Absence In Days For Period()

This function calculates cumulated absence days for the given date period and a given time type.

If the time type calculation is based on calendar days, the rule function returns the number of calendar days. If the time type is based on work schedule days, the rule function returns the number of work schedule days, considering holidays. If it is a flexible requesting time type based on unit hours, the rule function returns 0 days. Absences with undetermined end dates are also counted as 0 days.

The calculation considers the work schedule as well as the holiday calendar assigned to the employee. This rule function considers the *Approved*, *Pending*, and *Pending Cancellation* Employee Time statuses for calculating absence days or hours. For cross-midnight absences, the absence is considered based on the working day to which the absence belongs.

> **Note**
>
> • If you use this rule function in a take rule and the time type of the employee time triggering the take rule is the same as the rule function time type, the employee time that triggers the rule is not considered because it is not yet saved.

- If an employee time is edited, the absence in days from the existing employee time is covered. You can subtract that existing employee time quantity in days later and add the result of the current employee time.
- If no start date or end date is specified, the return is 0.

Input Parameters

For This Field	Make This Entry
User ID	Select the *User* field. This is the user for which the absence days should be calculated.
Time Type External Code	Enter the time type external code the rule should be executed for.
Start Date	Period start date
End Date	Period end date

Examples

Let's look at some examples.

Example 1

Here's the data for A.N.Other's absence calculation.

User ID: A.N. Other

Time Type External Code: Enter text "SPECIAL LEAVE". For this time type, the time type calculation is based on work schedule days.

Start Date = October 1, 2017

End Date = October 31, 2017

Prerequisites: User A.N. Other has a work schedule: Monday - Friday 8 hours, Saturday and Sunday 0 hours. He also has a work schedule with October 3, 2017, marked as a full holiday.

Result

Taken Leave of Type "SPECIAL LEAVE"

- October 1, 2017 – October 4, 2017 => 2 days because October 1, 2017 is a Sunday with 0 working hours and October 3, 2017 is a full holiday with 0 working hours
- October 31, 2017 – November 4, 2017 =>8 hours within October
- The rule function returns 24 hours.

Example 2

Here's the data for A.N.Other's absence calculation.

User ID: A.N. Other

Time Type External Code: Enter text "SPECIAL LEAVE". For Time Type "SPECIAL LEAVE", the time type calculation is based on calendar days.

Start Date = October 1, 2017

End Date = October 31, 2017

Prerequisites: User A.N. Other has a work schedule: Monday - Friday 8 hours, Saturday and Sunday 0 hours. He also has a work schedule with October 3, 2017, marked as a full holiday.

Result

Taken Leave of Type "SPECIAL LEAVE"

- October 1, 2017 – October 4, 2017 => 4 calendar days.
- October 31, 2017 – November 4, 2017 => 1 calendar day within October.
- The rule function returns 5 days.

Example 3: Cross Midnight

Here's the data for A.N.Other's absence calculation.

- User ID: A.N.Other
- Time Type External Code: Enter text "SPECIAL LEAVE". For Time Type Special Leave, the time type calculation is based on work schedule days.
- Start Date = October 1, 2020
- End Date = October 20, 2020

Prerequisites: User A.N. Other has a cross-midnight work schedule: Monday - Friday 8 hours from 10pm to 6am (next day), Saturday and Sunday 0 hours. He also has a holiday calendar with October 3, 2017 marked as full holiday.

Result

Taken Leave of Type "SPECIAL LEAVE"

- September 30, 2020 – October 1, 2020 (in detail September 30, 2020 10pm until October 2, 2020 6am, which belongs to the working day October 1, 2020) => 1 day. Only the working day started on October 1 is considered. October 20, 2020 – October 22, 2020 (in detail October 20, 2020 10pm until October 23, 2020 6am) => 1 days within selection period (October 20, 2020 10pm until October 21, 2020 6am). The hours physically lying on October 21, 2020 are considered as well, because they belong to the working day starting on October 20, 2020 as full absence.
- The rule function returns 2 days.

16.48 Get Absence In Days For Period Based On Calendar Days For Time Types()

This rule function calculates the total number of absence days in given duration for the time types provided.

This calculation is based on calendar days and does not take account of the work schedule or holiday calendar in the employee's job info.

For cross-midnight absences, the absence is considered based on the working day to which the absence belongs.

This rule function considers the *Approved*, *Pending*, and *Pending Cancellation* Employee Time statuses for calculating absence days or hours.

Input Parameters

For This Field	Make This Entry
User ID	Select the *User* field. This is the user for which the total number of absence days should be calculated.
Start Date	First day of duration in which absences are to be considered for calculation. You can use the start date of accrualRuleParameters. Take a look at use case 1 below.
End Date	Last day of duration in which absences are to be considered for calculation. You can use the end date of accrualRuleParameters. Take a look at use case 1 below.
Employee Time (optional)	The employee time is only necessary if you use the rule function in a take rule. Take a look at use case 2 below.
Time Types	Select time types. These are used to select employee times for calculation. **Note** Please only use time types with the time type category *Absence* for this rule function. Otherwise it can lead to errors.

Use Case 1: Accrual Rule

This rule function can be used in the accrual rule for a time account type where the accrual to be posted needs to consider the impact of other absences (like leave without pay) that the employee has taken in a specified time frame.

For example, this rule function can be used to calculate total number of absences that employee has taken from January 1, 2015, to December 31, 2015, which were of type leave without pay and unauthorized leave.

Examples

Here is data to calculate the total number of days of leave without pay (time type: LWOP) and unauthorized leave (time type: UNAL) during the year 2015 during accrual rule processing:

User ID: A.N. Other

Start Date = January 1, 2015

End Date = December 31, 2015

Time Types: “LWOP”, “UNAL”

Example Details 1: Existing Employee Times

October 1, 2015 – October 31, 2015

Result 1: The rule function will return 31 days.

Example Details 2: Existing Employee Times

No employee times.

Result 2: The rule function will return 0 days.

Example Details 3: Existing Employee Times

- February 15, 2015 - February 20, 2015 (6 days)
- December 20, 2015 - January 10, 2016 (12 days)

Result 3: The rule function returns 18 days.

Use Case 2: Take Rule

This rule function can be used in a take rule to check whether someone is allowed to take the corresponding leave.

For example, an employee is this rule function can be used to calculate total number of absences that employee has taken from January 1, 2015, to December 31, 2015, which were of type leave without pay and unauthorized leave.

The processing action (Create, Edit, Cancel) is part of the take rule parameters. Depending on this action, you can use the rule function to get the current number of calendar days, taken by the time type of the current employee time. If the action is Create or Edit, forward the employee time to the rule function. In the case of Cancel, it does not make sense to call the rule function at all.

Examples

Here is data to calculate total number of days taken of Time Type: Vacation during the year 2015:

User ID: A.N. Other

Start Date = January 1, 2015

End Date = December 31, 2015

Time Types: “Vacation”

Created Employee Time: November 2, 2015 - November 1, 2015

Results

Example Details 1:

- Existing Employee Time: October 1, 2015 – October 31, 2015

Result 1: The rule function returns 33 days. Depending on this result, an error can be thrown to avoid the creation of the employee time.

Example Details 2:

- Existing Employee Time: October 1, 2015 – October 31, 2015
- Existing Employee Time will be updated: November 1, 2015 - November 2, 2015

Result 2: The rule function returns 2 days. Depending on this result, an error can be thrown to avoid the creation of the employee time.

Example Details 3:

- Existing Employee Time: February 15, 2015 - February 20, 2015 (6 days)
- Existing Employee Time: December 20, 2015 - January 10, 2016 (12 days in 2015)
- Created Employee Time: November 1, 2015 - November 2, 2015 (2 days)

Result 3: The rule function returns 20 days. Depending on this result, an error can be thrown to avoid the creation of the employee time.

Cross-Midnight Example

Here's the data for A.N.Other's absence calculation.

User ID: A.N. Other

Start Date = October 1, 2020

End Date = October 20, 2020

Time Types: "SPECIAL LEAVE"

Result

Taken Leave of Type "SPECIAL LEAVE"

- September 30, 2020 – September 30, 2020 (in detail September 30, 2020 10pm until October 1, 2020 6am, which belongs to the working day September 30, 2020)=> 0 days, because the working hours laying on October 1st belong to the working day of September 30, 2020 and therefore are not considered.
- October 5, 2020 - October 10, 2020 (in detail October 5, 2020 10pm until October 11, 2020 6am) => 6 calendar days; although the leave ends physically on October 11, 2020 6am, this day belongs to October 10, 2020 and is not considered.
  October 20, 2020 – October 22, 2020 (in detail October 20, 2020 10pm until October 23, 2020 6am)=> 1 day within selection period (October 20, 2020 10pm until October 21, 2020 6am). The hours physically lying on October 21, 2020 are considered as well, because they belong to the working day starting on October 20, 2020 and mark the working day starting on October 20, 2020 as full absence.

The rule function returns 7 days.

16.49 Get Absence In Days For Period Based On Working Days For Leave Of Absence Time Types()

This rule function calculates the number of absence days during a given duration for the given time types.

The calculation considers the work schedule as well as the holiday calendar assigned to the employee. This rule function considers the *Approved*, *Pending*, and *Pending Cancellation* Employee Time statuses for calculating absence days or hours. For cross-midnight absences, the absence is considered based on the working day to which the absence belongs.

- You can use this rule function for the time type setting *Duration display according to calendar days* as well as for work schedule days.
- You can use it to prorate accruals in an accrual rule scenario, or to raise validation errors in Take rules.
- You can specify whether the holiday calendar should be considered or not.

> **Note**
>
> This rule function returns the result in working days. If you want to see the result in calendar days, you should use the *Get Absence In Days For Period Based On Calendar Days()* rule function instead.

Reduced Scope

- This rule function can only be used for leave for absence time types.
- It cannot get absence days for a period of time longer than one year.
- The start date of the period cannot fall after the end date.

Input Parameters

For this field	Make this entry
User ID	Select the user for which the absence days should be calculated.
Start Date	Enter the start date of the period for which the absence days should be calculated.
End Date	Enter the end date of the period for which the absence days should be calculated.
Time Types	Select time types. These are used to select employee times for calculation. **Note** Please only use time types with the time type category *Absence* for this rule function. Otherwise it can lead to errors.
Employee Time (optional)	This is only necessary if you want to use the rule function in a Take rule.
Consider Holidays	Specify whether holidays should be taken into account when calculating absence days.

Behavior in Case of Errors

If the *Start Date* is greater than the *End Date*, the rule execution is stopped and an error message is shown.

If the value between *Start Date* and the *End Date* is greater than 12, the rule execution is stopped and an error message is shown.

If the *Start Date*, *End Date*, *User ID*, *Consider Holiday*, or *Time Type* is *null*, the rule execution is stopped and an error message is shown.

If you select a time type that is not a Leave of Absence time type, the rule execution is stopped and an error message is shown.

Example #1: Using the Rule Function in an Accrual Rule

In this example, let's assume we have the following scenario:

- **Employee's work schedule:** Monday to Friday, 8 hours each day
- **Existing absences:**
  - October 1st 2019 - October 6th 2019 (Tuesday to Sunday); Time Type = Leave of Absence 1
  - October 31st 2019 - November 5th 2019 (Thursday to Tuesday); Time Type = Leave of Absence 2
- Public holiday (full day) on Friday October 3rd 2019
- Employee does not accrue leave in the accrual period if they take more than a specified number of leave of absence days within that period.

You run the rule function with these parameters:

- *User ID*: Employee's ID
- *Start Date*: October 1st 2019
- *End Date*: October 31st 2019
- *Time Type External Codes*: Leave of Absence 1; Leave of Absence 2

And you get the following results:

- If you selected *Consider Holidays*, the result is 5 days.
  The first leave request is counted in its entirety, but only October 1st 2019 to October 4th 2019 are working days (4 days). From the second leave request, only October 31st 2019 falls within the selection period, and it's a working day (1 day).
- If you didn't select *Consider Holidays*, the result is 4 days.
  The first leave request is counted in its entirety, but only October 1st, 2nd, and 4th 2019 are working days (3 days). From the second leave request, only October 31, 2019 is within the selection period, and it's a working day (1 day).

Example #2: Using the Rule Function in a Take Rule

In this example, we have the following scenario:

- **Employee's work schedule:** Monday to Friday, 8 hours each day
- **Existing absences:**
  - October 1st 2019 - October 6th 2019 (Tuesday to Sunday); Time Type = Leave of Absence 1
  - October 31st 2019 - November 5th 2019 (Thursday to Tuesday); Time Type = Leave of Absence 2
- **Employee Time in creation:** December 1st 2019 to December 5th 2019 (Tuesday to Saturday); Time Type = Leave of Absence 1
- Public holiday (full day) on Friday October 3rd 2019
- Employee is allowed to take a maximum of 25 days leave in calendar year 2019.

You run the rule function with these parameters:

- *User ID*: Employee's ID
- *Start Date*: January 1st 2019
- *End Date*: December 31st 2019
- *Time Type External Codes*: Leave of Absence 1; Leave of Absence 2
- *Employee Time*: EmployeeTime

And you get the following results:

- If you selected *Consider Holidays*, the result is 12 days.
  The first leave request is counted in its entirety, but only October 1st to October 4th 2019 are working days (4 days). From the second leave request, 4 working days are counted. And from the leave request, 4 days are counted (Tuesday to Friday, since Saturday and Sunday are non-working days).
- If you didn't select *Consider Holidays*, the result is 11 days.
  The first leave request is counted in its entirety, but only October 1st, 2nd, and 4th 2019 are working days (3 days). From the second leave request, there are 4 working days. And the employee time in creation has 4 working days as well.
- The total number of leave days is less than 25. As such, if you specified that an error message should appear if the number of leave days in a year exceeds 25, no error message will be produced in this case.

16.50 Get Absence In Days For Period Based On Working Days For Time Types()

This function calculates the cumulated absence days for the given duration and given time types.

The calculation considers the work schedule as well as the holiday calendar assigned to the employee. This rule function considers the *Approved*, *Pending*, and *Pending Cancellation* Employee Time statuses for calculating absence days or hours. For cross-midnight absences, the absence is considered based on the working day to which the absence belongs.

Regardless of the unit used in the time type (hours or days), the rule function returns the absence in working days.

> **Note**
>
> This rule function doesn't work properly with leave of absence time types and absences with the undetermined end date indicator set. These absences are taken into account with quantity 0.

Input Parameters

For This Field	Make This Entry
User ID	Select the *User* field. This is the user for which the start time of the working day should be calculated.
Start Date	First day of duration in which absences are to be considered for calculation. You can use the start date of accrualRuleParameters. Take a look at use case 1 below..
End Date	Last day of duration in which absences are to be considered for calculation. You can use the end date of accrualRuleParameters. Take a look at use case 1 below.
Employee Time (optional)	The employee time is only necessary if you use the rule function in a take rule. Take a look at use case 2 below.
Time Types	Select Time Types. These are used to select Employee Times for calculation.

Use Case 1: Accrual Rule

This rule function can be used in the accrual rule for a time account type where the accrual to be posted needs to consider the impact of other absences (like leave without pay) that the employee has taken in a specified time frame.

For example, this rule function can be used to calculate total number of absences that employee has taken from January 1, 2016, to December 31, 2016, which were of type leave without pay and unauthorized leave.

Examples

Here is data to calculate the total number of days of leave without pay (time type: LWOP) and unauthorized leave (time type: UNAL) during the year 2016 during accrual rule processing:

User ID: A.N. Other

Start Date = January 1, 2016

End Date = December 31, 2016

Time Types: “LWOP”, “UNAL”

Work schedule pattern : Monday to Friday are working days.

Example Details 1: Existing Employee Times

October 1, 2016 – October 31, 2016

Result 1: The rule function returns 21 days.

Example Details 2: Existing Employee Times

No employee times.

Result 2: The rule function returns 0 days.

Example Details 3: Existing Employee Times

- February 15, 2016 - February 20, 2016 (6 days)
- December 20, 2016 - January 10, 2017 (9 days)

Result 3: The rule function returns 14 days.

Cross-Midnight Example

Here's the data for A.N.Other's absence calculation.

User ID: A.N. Other

Start Date = October 1, 2020

End Date = October 20, 2020

Time Types: "SPECIAL LEAVE"

Prerequisites:

- User A.N. Other has a cross-midnight work schedule: Monday - Friday 8 hours from 10pm to 6am (next day), Saturday and Sunday 0 hours.
- No holidays in selection period October 1, 2020 - October 20, 2020.

Result

Taken Leave of Type "SPECIAL LEAVE"

- September 30, 2020 – September 30, 2020 (in detail September 30, 2020 10pm until October 1, 2020 6am, which belongs to the working day September 30, 2020)=> 0 days, because the working hours laying on October 1st belong to the working day of September 30, 2020 and therefore are not considered.
- October 5, 2020 - October 10, 2020 (in detail October 5, 2020 10pm until October 11, 2020 6am) => 5 work schedule days
- October 20, 2020 – October 22, 2020 (in detail October 20, 2020 10pm until October 23, 2020 6am)=> 1 day within selection period (October 20, 2020 10pm until October 21, 2020 6am). The hours physically lying on October 21, 2020 are considered as well because they belong to the working day starting on October 20, 2020 and mark the working day starting on October 20, 2020 as full absence.

The rule function returns 6 days.

Use Case 2: Take Rule

This rule function can be used in a take rule to check whether someone is allowed to apply for or modify the absence.

For example, this rule function can be used to calculate absence in terms of total number of working days that the employee has taken from January 1, 2016, to December 31, 2016, which were of type leave without pay and unauthorized leave. If these leaves exceeds a specific number, for example 20 days, then the employee cannot create the absence record for future leave.

The processing action (Create, Edit, Cancel) is part of the take rule parameters. Depending on this action, you can use the rule function to get the current number of calendar days, taken by the time type of the current employee

time. If the action is Create or Edit, forward the employee time to the rule function. In the case of Cancel, it does not make sense to call the rule function at all.

Examples

Here is data to calculate total number of days taken of Time Type: Vacation during the year 2016:

User ID: A.N. Other

Start Date = January 1, 2016

End Date = December 31, 2016

Time Types: "Vacation"

Results

Example Details 1:

- Existing Employee Time: October 1, 2016 – October 31, 2016
- Created Employee Time: November 1, 2016 - November 2, 2016

Result 1: The rule function returns 23 days. Depending on this result, an error can be thrown to avoid the creation of the employee time.

Example Details 2:

- Existing Employee Time: October 1, 2016 – October 31, 2016
- Existing Employee Time will be updated: November 1, 2016 - November 2, 2015

Result 2: The rule function returns 2 days. Depending on this result, an error can be thrown to avoid the creation of the employee time.

Example Details 3:

- Existing Employee Time: February 15, 2016 - February 20, 2016 (5 days)
- Existing Employee Time: December 12, 2016 - January 10, 2017 (15 days in 2015)
- Created Employee Time: November 1, 2016 - November 2, 2016 (2 days)

Result 3: The rule function returns 22 days. Depending on this result, an error can be thrown to avoid the creation of the employee time.

16.51 Get Absence In Days For Period Based On Working Days For Time Types Excluding Weekdays()

With this rule function, you can exclude the weekdays of your choice when getting the sum total of absence days.

The calculation considers the work schedule as well as the holiday calendar assigned to the employee. This rule function considers the *Approved*, *Pending*, and *Pending Cancellation* Employee Time statuses for calculating absence days or hours. For cross-midnight absences, the absence is considered based on the working day to which the absence belongs.

Here's an example of a situation where you might want to do that.

As an employer, you want to provide your employees with meal vouchers for lunch using Employee Central Global Benefits. At the beginning of each month, employees receive one meal voucher for each day worked during the previous month. As such, in order to calculate the individual number of meal vouchers for each employee, the employee's absences need to be deducted from the number of planned working days according to the employee's work schedule.

However, on one weekday (say Friday), employees leave work prior to lunch, so Fridays are excluded from the sum of working days. If, in addition to that, an employee records an absence on Friday, the rule function avoids deducting this day twice from the final number by excluding it from the calculation of absence days.

> **Note**
>
> This rule function doesn't work properly with leave of absence time types and absences with the undetermined end date indicator set. These absences are taken into account with quantity 0.

Input Parameters

For this field	Make this entry
User ID	Select the user for which the start time of the working day should be calculated.
Start Date	The start date of the time period for which you want to calculate.
End Date	The end date of the time period for which you want to calculate.
Weekdays to be Excluded	Select the *Get Weekdays()* rule function, and choose one or more days you want to exclude from calculation.
Employee Time (optional)	**Note** This is only relevant if you use the rule function in a Take rule. Enter the employee time that will be modified or created when the Take rule is triggered.
Time Types	Select time types. These are used to select employee times for calculation. **Note** Please only use time types with the time type category *Absence* for this rule function. Otherwise it can lead to errors.

Examples

Let's look at some examples.

Here's the data for A.N.Other's absence calculation.

User ID: A.N. Other

Start Date = July 1, 2019

End Date = July 31, 2019

Time Types: "Vacation", 'Sick Leave'

Weekdays: "Friday"

Prerequisites: User A.N. Other has a work schedule: Monday - Friday 8 hours, Saturday and Sunday 0 hours.

Example Details 1: Existing Employee Times

Employee Time: MON, July 8, 2019 - FRI, July 12, 2019

Result: The rule function returns 4 days.

Example Details 2: Existing Employee Times

Employee Time: MON, July 8, 2019 - THU, July 11, 2019

Result: The rule function returns 4 days.

Example Details 3: Existing Employee Times

Employee Time: FRI, July 12, 2019

Result: The rule function returns 0 days.

Cross-Midnight Example:

Here's the data for A.N.Other's absence calculation.

User ID: A.N. Other

Start Date = October 1, 2020

End Date = October 20, 2020

Time Types: "SPECIAL LEAVE"

Weekdays: "Friday"

Prerequisites: User A.N. Other has a cross-midnight work schedule: Monday - Friday 8 hours from 10pm to 6am (next day), Saturday and Sunday 0 hours.

Prerequisites: No holidays in selection period October 1, 2020 - October 20, 2020.

Result

Taken Leave of Type "SPECIAL LEAVE"

September 30, 2020 – September 30, 2020 (in detail September 30, 2020 10pm until October 1, 2020 6am, which belongs to the working day September 30, 2020)=> 0 days, because the working hours laying on October 1st belong to the working day of September 30, 2020 and so are not considered.

October 5, 2020 - October 10, 2020 (in detail October 5, 2020 10pm until October 11, 2020 6am) => 4 work schedule days, cause Friday (October 9, 2020) is excluded.

October 20, 2020 – October 22, 2020 (in detail October 20, 2020 10pm until October 23, 2020 6am)=> 1 day within selection period (October 20, 2020 10pm until October 21, 2020 6am). The hours physically lying on October 21, 2020 are considered as well, because they belong to the working day started on October 20, 2020 and mark the working day starting on October 20, 2020 as full absence.

The rule function returns 5 days.

16.52 Get Absence in Days For Period with Threshold()

This rule function calculates the sum of absence days in a given period for the time types provided where individual Employee Time duration exceeds a given threshold value.

This rule function considers the *Approved*, *Pending*, and *Pending Cancellation* Employee Time statuses for calculating absence days or hours.

If an Employee Time does not fall into the rule function period completely, but overlaps, only the days within the period are considered. Please refer to the examples for more details.

For cross-midnight absences, the absence is considered based on the working day to which the absence belongs.

This rule function can carry out this calculation in two different ways based on the value of indicator parameter:

- Count All Days In Range
  Identify all Employee Times for which absence duration individually exceeds threshold. For all these Employee Times, calculate sum of total absence days in each Employee Time.
- Count Only Days Above Threshold
  Identify all Employee Times for which absence duration individually exceed threshold. For all these Employee Times, calculate sum of those absence days in each Employee Time which exceed threshold value.

This calculation is based on calendar days and does not take into account the work schedule or public holiday calendar as mentioned in employee's job info. If a one day Employee Time is considered, the number of days is taken from the quantity in days field, meaning 0.5 days for a half day absence.

The rule function can be used in the accrual rule for time account types where any accrual to be posted needs to consider the impact of other absences that the employee has taken in the specified time frame.

Input Parameters

For This Field	Make This Entry
User ID	Select the User field. This is the user for which the absence days should be calculated where each Employee Time has a duration that exceeds the Threshold value.
Start Date	Start date of duration in which absences are to be considered for calculation. You can use the start date of accrualRuleParameters.

For This Field	Make This Entry
End Date	End date of duration in which absences are to be considered for calculation. You can use the end date of accrualRuleParameters.
Threshold	This is used to specify the limit that is compared with duration of each Employee Time to identify whether that Employee Time needs to be considered for calculation or not.
Counting Method	Specifies whether for each relevant Employee Time whether: total days or only the part that exceeds the threshold value are used for the calculation of total absence days. Possible values are: • Count All Days In Range • Count Only Days Above Threshold
Time Types	Select time types. These are used to select employee times for calculation. **Note** Please only use time types with the time type category *Absence* for this rule function. Otherwise it can lead to errors.

Examples

Example 1

Here is data to calculate the total number of absences that an employee has taken from January 1 to December 31 2015 which were of type leave without pay (LWOP) and unauthorized leave (UNAL) with a duration of more than 5 days for each absence:

- User ID: A.N. Other
- Start Date: January 1, 2015
- End Date: December 31, 2015
- Threshold: 5
- Counting Method: Count All Days in Range
- "LWOP", "UNAL".

The employee has taken the following absences during 2015:

- Leave without pay from March 2, 2015, to March 10, 2015 (exceeds threshold of 5 days)
- Leave without pay from May 15, 2015, to May, 16, 2015 (does not exceed threshold of 5 days)
- Unauthorized leave from October 1, 2015, to October 15, 2015 (exceeds threshold of 5 days)
- Result 1: Rule function returns 24 because:
  - For the first Employee Time: Threshold of 5 days is exceeded = 9 days

- For the second Employee Time: Threshold of 5 days is not exceeded = 0 days
- For third Employee Time: Threshold of 5 days is exceeded = 15 days

Example 2

Here is data to calculate the total number of absences that an employee has taken from January 1 to December 31 2015 which were of type leave without pay (LWOP) and unauthorized leave (UNAL) with a duration of more than 5 days for each absence:

- User ID: A.N. Other
- Start Date: January 1, 2015
- End Date: December 31, 2015
- Threshold: 5
- Counting Method: Count Only Days Above Threshold
- "LWOP", "UNAL"

The employee has taken the following absences during 2015:

- Leave without pay from March 2, 2015, to March 10, 2015 (exceeds threshold of 5 days)
- Leave without pay from May 15, 2015, to May, 16, 2015 (does not exceed threshold of 5 days)
- Unauthorized leave from October 1, 2015, to October 15, 2015 (exceeds threshold of 5 days)
- Result 2: Rule function returns 14 because:
  - For the first Employee Time: Threshold of 5 days is exceeded by 4 days = 4 days
  - For the second Employee Time: Threshold of 5 days is not exceeded = 0 days
  - For third Employee Time: Threshold of 5 days is exceeded by 10 days = 10 days

Example 3

Here is data to calculate the total number of absences that an employee has taken from January 1 to December 31 2015 which were of type leave without pay (LWOP) and unauthorized leave (UNAL) with a duration of more than 5 days for each absence:

- User ID: A.N. Other
- Start Date: January 1, 2015
- End Date: December 31, 2015
- Threshold: 5
- Counting Method: Count Only Days Above Threshold
- "LWOP", "UNAL"

Employee has taken following absences during year 2015:

- Leave without pay from December 28, 2014, to January 10, 2015 (exceeds threshold of 5 days)
- Leave without pay from May 15, 2015, to May, 16, 2015 (does not exceed threshold of 5 days)
- Unauthorized leave from December 28, 2015, to January 10, 2016 (exceeds threshold of 5 days)
- Result 3: Rule function returns 9 because:
  - For the first Employee Time: Threshold of 5 days is exceeded by 5 days within selection period. Days in December 2014 are not considered = 10 days.
  - For the second Employee Time: Threshold of 5 days is not exceeded = 0 days.
  - For third Employee Time: Threshold of 5 days is not exceeded within selection period. Days in January 2016 are not considered = 0 days.

Example 4

Here is data to calculate the total number of absences that an employee has taken from January 1 to December 31 2015 which were of type leave without pay (LWOP) and unauthorized leave (UNAL) with a duration of more than 5 days for each absence:

- User ID: A.N. Other
- Start Date: January 1, 2015
- End Date: December 31, 2015
- Threshold: 5
- Counting Method: Count All Days In Range
- "LWOP", "UNAL"

Employee has taken following absences during year 2015:

- Leave without pay from December 28, 2014, to January 10, 2015 (exceeds threshold of 5 days)
- Leave without pay from May 15, 2015, to May, 16, 2015 (does not exceed threshold of 5 days)
- Unauthorized leave from December 28, 2015, to January 10, 2016 (exceeds threshold of 5 days)
- Result 4: Rule function returns 10 because:
  - For the first Employee Time: Threshold of 5 days is exceeded by 5 days within selection period. Days in December 2014 are not considered at all = 5 days
  - For the second Employee Time: Threshold of 5 days is not exceeded = 0 days
  - For third Employee Time: Threshold of 5 days is exceeded within selection period. Days in January 2016 are not considered at all = 0 days

Example 5: Cross Midnight

Here is data to calculate the total number of absences that an employee has taken from July 1 to July 31 2020 which were of type leave without pay (LWOP) and unauthorized leave (UNAL) with a duration of more than 5 days for each absence. The user has a cross-midnight work schedule Monday to Friday 10pm to 6am, Saturday and Sunday are not working days.

- User ID: A.N. Other
- Start Date: July 1, 2020
- End Date: July 31, 2020
- Threshold: 5
- Counting Method: Count All Days in Range
- "LWOP" (Leave without pay), "UNAL" (Unauthorized leave)

Employee has taken following absences during July 2020:

- Leave without pay from July 3, 2020 to July 7, 2020, which means the shift starting on July 7, 2020 is included in the absence and the absence ends physically on July 8, 2020 6am (does not exceed threshold of 5 days)
- Leave without pay from July 10, 2020 to July 20, 2020 (exceeds threshold of 5 days)
- Unauthorized leave from July 28, 2020 to August 10, 2020 (does not exceed threshold of 5 days)
- Result 5: Rule function returns 11 because:
  - For the first Employee Time: Threshold of 5 days is not exceeded; July 8, 2020 is not considered because it belongs to July 7th where the shift starts = 0 days
  - For the second Employee Time: Threshold of 5 days is exceeded by 6 days = 11 days
  - For third Employee Time: Threshold of 5 days is not exceeded within selection period = 0 days

16.53 Get Absence In Days Or Hours For Period Based On Deduction Quantity For Time Types()

This rule function calculates the total deduction quantity, in days or hours, for absences in a specified period and with the specified time types.

The rule function returns the absence hours or days by summing the deduction quantity. It considers the *Approved*, *Pending*, and *Pending Cancellation* Employee Time statuses for calculating absence days or hours.

Deduction quantity is the quantity calculated after the absence counting rule is applied. The rule function returns the deduction quantity in the unit in which the time type is defined.

The rule function also has an optional parameter to return only approved absence hours or days.

If there is a Leave of Absence (LOA) time type in the list of time types (input parameter), the rule returns 0.

If there is an absence with an undetermined end date for the selected user within the selected time, then rule function returns 0.

You can use the rule function in both take rules and accrual rules.

Input Parameters

For This Field	Make This Entry
User ID	Select the User field. This is the user for which the total number of absence days should be calculated.
Start Date	First day of duration in which absences are to be considered for calculation.
End Date	Last day of duration in which absences are to be considered for calculation.
Employee Time (Optional)	The employee time is only necessary if you use the rule function in a take rule. This should be the employee time being modified or created when the take rule is triggered.
considerOnlyApprovedAbsences (Optional)	Select this option if you only want to consider approved absences.
Time Types	Select Time Types. These are used to select Employee Times for the calculation.

Use Cases

Use Case 1: Accrual Rule

This rule function can be used in the accrual rule for a time account type where the accrual to be posted needs to consider the impact of other absences (like leave without pay) that the employee has taken in a specified time frame.

For example, this rule function can be used to calculate total number of absence hours or days that employee has taken from January 1, 2022, to December 31, 2022, which were of type "Sickness".

You must use the **Accrual** rule scenario to create this rule.

Examples

Here is data to calculate the total number of deduction days for Sickness leave in the year 2022 during accrual rule processing:

- User ID: A.N. Other
- Start Date = January 1, 2022
- End Date = December 31, 2022
- Time Types: “SICKNESS"
- Work schedule pattern : Monday to Friday are working days.

Example Details 1: Existing Employee Times

November 1, 2022 – November 14, 2022

Result 1: The rule function will return 10 days in line with the counting rule.

Example Details 2: Existing Employee Times

No employee times.

Result 2: The rule function will return 0 days.

Example Details 3: Existing Employee Times

- November 1, 2022 – November 14, 2022 (10 days)
- December 20, 2022 - December 31st 2022 (9 days)

Result 3: The rule function will return 19 days.

Result : The rule function should return 18 days.

Use Case 2: Take Rule

This rule function can be used in a take rule to raise a validation error if the absence hours or days exceed a certain limit in a period.

You must use the **Absence Validation** scenario to create this rule.

For example you can use this rule function to calculate absence in terms of total number of absence hours ordays by deduction quantity that the employee has taken from January 1, 2022, to December 31, 2022. For example, sick leave can count holiday as a regular absence day and should be considered in a take rule.

In this case, you can define an Absence Counting Rule with "Consider Holidays" = No, so that holidays are counted as an absence day.

The processing action (Create, Edit, Cancel) is part of the take rule parameters. Depending on this action, you can use the rule function to get the current number of deduction quantity, taken by the time type of the current employee time.

Examples

Here is data to calculate the total number of days taken of Time Type: Sickness during the year 2022.

Raise an error if the absence days exceeds 20 days.

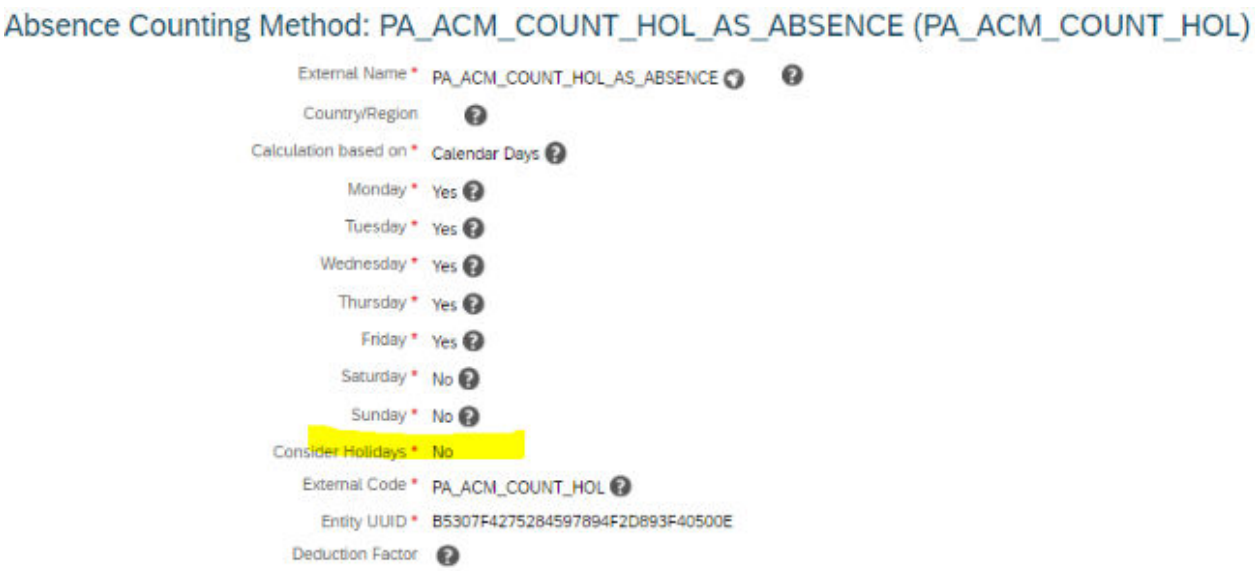
- User ID: A.N. Other
- Start Date = January 1, 2022
- End Date = December 31, 2022
- Time Types: “Sick Leave”

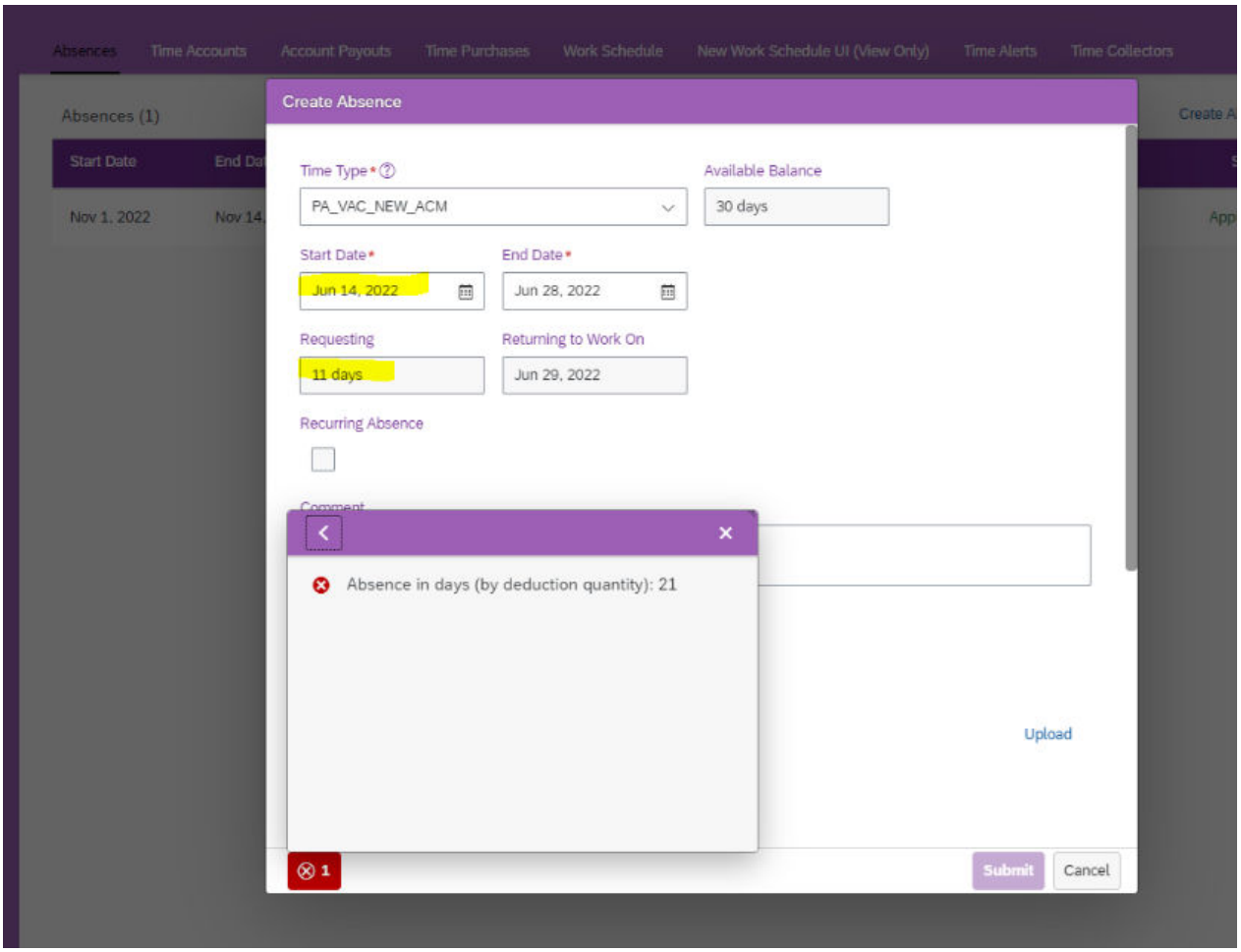
Results

Example Details :

- Existing Employee Time: November 1, 2022 – November 14, 2022 (10 days)
- Created Employee Time: June 14, 2022 - June 28, 2022 (11 days)

June 14th is a public holiday. The absence counting rule is defined so that holiday is counted as an absence day.





The rule function will return 21 days. Depending on this result, an error can be raised to avoid the creation of the employee time.

16.54 Get Absence In Hours For Period()

This function calculates cumulated absence hours for the given date period and a given time type.

The calculation considers the work schedule as well as the holiday calendar assigned to the employee. This rule function considers the *Approved*, *Pending*, and *Pending Cancellation* Employee Time statuses for calculating absence days or hours. For cross midnight absences, the absence is considered based on the working day to which the absence belongs.

If the time type is relevant for leave of absence, the rule function returns 0 hours. If it is a flexible requesting time type based on unit days, the rule function returns 0 hours. Absences where the undetermined end date indicator is set are taken into account with quantity 0,

> **Note**
>
> - If you use this rule function in a take rule and the time type of the employee time triggering the take rule is the same as the rule function time type, the employee time that triggers the rule is not considered because it is not yet saved.

- If an employee time is edited, the absence in hours from the existing employee time is covered. You can subtract that existing employee time quantity in hours later and add the result of the current employee time.
- If no start date or end date is specified, the return is 0.

Input Parameters

For This Field	Make This Entry
User ID	Select the *User* field. This is the user for which the absence hours should be calculated.
Time Type External Code	Enter the time type external code the rule should be executed for.
Start Date	Period start date
End Date	Period end date

Examples

Let's look at some examples.

Example 1

Here's the data for A.N.Other's absence calculation.

User ID: A.N. Other

Time Type External Code: Enter text "SPECIAL LEAVE".

Start Date = October 1, 2017

End Date = October 31, 2017

Prerequisites: User A.N. Other has a work schedule: Monday - Friday 8 hours, Saturday and Sunday 0 hours. He also has a holiday calendar with October 3, 2017 marked as a full holiday.

Result

Taken Leave of Type "SPECIAL LEAVE"

- October 1, 2017 – October 4, 2017 => 16 hours because October 1st, 2017 is a Sunday with 0 working hours and October 3rd, 2017 is a full holiday with 0 working hours.
- October 31, 2017 – November 4, 2017 => 8 hours within October.
- The rule function returns 24 hours.

Example 2 Cross Midnight

Here's the data for A.N.Other's absence calculation.

User ID: A.N. Other

Time Type External Code: Enter text "SPECIAL LEAVE".

Start Date = October 1, 2020

End Date = October 20, 2020

Prerequisites: User A.N. Other has a work schedule: Monday - Friday 8 hours from 10pm to 6am (next day), Saturday and Sunday 0 hours. He also has a holiday calendar with October 3, 2020 marked as a full holiday.

Result

Taken Leave of Type "SPECIAL LEAVE"

- September 30, 2020 – October 1, 2020 (in detail September 30, 2020 10pm until October 2, 2020 6am, which belongs to the working day October 1, 2020)=> 8 hours, because only the working day starting on October 1st is considered, which is October 1, 2020 10pm until October 2, 2020 6am. October 1 midnight until 6am belongs to the working day starting on September 30, 2020 and is not considered here.
- October 20, 2020 – October 22, 2020 (in detail October 20, 2020 10pm until October 23, 2020 6am) => 8 hours within selection period (October 20, 2020 10pm until October 21, 2020 6am). The hours physically lying on October 21, 2020 are considered as well because they belong to the working day starting on October 20, 2020.
- The rule function returns 16 hours.

16.55 Get Absence In Hours For Period For Time Types()

This function calculates the cumulated absence hours for the given duration and given time types.

This rule function considers the *Approved*, *Pending*, and *Pending Cancellation* Employee Time statuses for calculating absence days or hours.

Regardless of the unit used in the time type (hours or days), the rule function returns the absence in hours. Absences for a leave of absence (LOA) time type and absences where the undetermined end date indicator is set are taken into account with quantity 0.

For the calculation, the quantity in hours field on the Employee Time and Employee Time Calendar is considered. The calculation is based on the employee's work schedule and holiday calendar.

For cross-midnight absences, the absence is considered based on the working day to which the absence belongs.

Input Parameters

For This Field	Make This Entry
User ID	Select the *User* field. This is the user for which the start time of the working day should be calculated.
Start Date	Duration start date.
End Date	Duration end date.

For This Field	Make This Entry
Employee Time	The employee time is only necessary if you use the rule function in a take rule. This should be the employee time being modified or created when the take rule is triggered.
Time Types	Select time types. These are used to select employee times for calculation. **Note** Please only use time types with the time type category *Absence* for this rule function. Otherwise it can lead to errors.
Consider Only Approved Absences (optional)	Select "Yes" if you want to consider **only** absences in status Approved. By default, all absences in the statuses Approved, Pending, and Pending Cancellation are considered.

Use Case 1: Accrual Rule

This rule function can be used in the accrual rule for a time account type where the accrual to be posted needs to consider the impact of other absences (like leave without pay) the employee has taken in a specified time frame.

For example, this rule function can be used to calculate the total number of absences that employee has taken from January 1, 2016, to December 31, 2016, which were of type leave without pay and unauthorized leave.

Examples

Here is data to calculate the total number of days of leave without pay (time type: LWOP) and unauthorized leave (time type: UNAL) during the year 2016 during accrual rule processing:

User ID: A.N. Other

Start Date = January 1, 2016

End Date = December 31, 2016

Time Types: “LWOP”, “UNAL”

Work schedule pattern : Monday to Friday are working days of 8 hours each.

Example Details 1: Existing Employee Times

October 3, 2016 – October 5, 2016

Result 1: The rule function returns 24 hours.

Example Details 2: Existing Employee Times

No employee times.

Result 2: The rule function returns 0 hours.

Example Details 3: Existing Employee Times

- February 15, 2016 - February 16, 2016 (16 hours)
- December 30, 2016 - January 10, 2017 (8 hours)

Result 3: The rule function returns 24 hours.

Use Case 2: Take Rule

This rule function can be used in a take rule to check whether someone is allowed to apply for or modify the absence.

For example, the rule function can be used to calculate absence in terms of total number of working hours that the employee has taken from January 1, 2016, to December 31, 2016, which were of type leave without pay and unauthorized leave. If these leaves exceed a specific number, for example 20 hours, then the employee cannot create the absence record for future leave.

The processing action (Create, Edit, Cancel) is part of the take rule parameters. Depending on this action, you can use the rule function to get the current number of calendar days, taken by the time type of the current employee time. If the action is Create or Edit, forward the employee time to the rule function. In the case of Cancel, it does not make sense to call the rule function at all.

Examples

Here's the data to calculate total number of days taken of Time Type: Vacation during the year 2016:

User ID: A.N. Other

Start Date = January 1, 2016

End Date = December 31, 2016

Time Types: “Vacation”

Results

Example Details 1:

- Existing Employee Time: October 1, 2016 – October 31, 2016
- Created Employee Time: November 1, 2016 - November 2, 2016

Result 1: The rule function returns 32 hours. Depending on this result, an error can be thrown to avoid the creation of the employee time.

Example Details 2:

- Existing Employee Time: October 1, 2016 – October 31, 2016
- Existing Employee Time will be updated: November 1, 2016 - November 2, 2015

Result 2: The rule function returns 16 days. Depending on this result, an error can be thrown to avoid the creation of the employee time.

Example Details 3:

- Existing Employee Time: February 15, 2016 - February 17, 2016 (27 days)

- Existing Employee Time: December 30, 2016 - January 10, 2017 (8 hours in 2016)
- Created Employee Time: November 1, 2016 - November 2, 2016 (16 hours)

Result 3: The rule function returns 51 hours. Depending on this result, an error can be thrown to avoid the creation of the employee time.

Example 4: Cross Midnight:

Here's the data for A.N.Other's absence calculation.

User ID: A.N. Other

Time Type External Code: Enter text "SPECIAL LEAVE".

Start Date = October 1, 2020

End Date = October 20, 2020

Prerequisites: User A.N. Other has a cross-midnight work schedule: Monday - Friday 8 hours from 10pm to 6am (next day), Saturday and Sunday 0 hours. He also has a holiday calendar with October 3, 2020 marked as a full holiday.

Result

Taken Leave of Type "SPECIAL LEAVE"

- September 30, 2020 – October 1, 2020 (in detail September 30, 2020 10pm until October 2, 2020 6am, which belongs to the working day October 1, 2020)=> 8 hours, because only the working day starting on October 1 is considered, which is October 1, 2020 10pm until October 2, 2020 6am. October 1 midnight until 6am belongs to the working day starting on September 30, 2020 and is not considered here.
- October 20, 2020 – October 22, 2020 (in detail October 20, 2020 10pm until October 23, 2020 6am)=> 8 hours within selection period (October 20, 2020 10pm until October 21, 2020 6am). The hours physically lying on October 21, 2020 are considered as well because they belong to the working day starting on October 20, 2020.

The rule function returns 16 hours.

Use Case 3: Give me all approved absence hours in the year 2022

Here's the data for A.N.Other's calculation.

User ID: A.N. Other

Start Date = January 1, 2022

End Date = December 31, 2022

Time Type: Vacation

Consider only approved absences = Yes

February 15, 2022 - February 16, 2022 (16 hours) - Approved

June 27, 2022 - June 30, 2022 (32 hours) - Pending

Result: The rule function returns 16 hours.

16.56 Get Age of Dependent in Months()

This rule function gets the age of the dependent for a given list of relationship types and ranked as of a given date.

The rule function returns the age of the dependent in completed months and days. It is required for the use case where an employee can be granted maternity, child care, or paternal leave if the dependent is of a particular age. Dependents are sorted in ascending order of date of birth. If there are two or more dependents with the same date of birth, such as twins or triplets, sorting is done based on last name followed by first name.

Input Parameters

For this parameter	Make this entry
Person ID External	Person ID external, accessed from the biographical information on the rule UI.
User ID	User ID for which youngest dependent is calculated.
Date	The date on which youngest dependent is fetched.
Relationship Types	List of relationship types for which dependents will be fetched. The values shown are the external codes of the options available in the relationship type picklist.
Rank	Rank of dependent whose age is to be calculated

Examples

Let's look at an example.

Employee is hired on January 1, 2010, so the dependent card has following data for the employee's dependents:

Relationship Type	Date of Birth	Last Name	First Name
Child	September 1, 2009	Klarc	Adam
Child	September 1, 2009	Calley	Bob
Child		Eethen	Alen
Child		Eethen	Gable
Child	January 27, 2016	Alaric	Anna

Sorting results in the following ranking:

Sorted Dependents

Rank	Dependent
1	Klarc, Adam
2	Calley, Bob
3	Alaric, Anna
4	Eethen, Alen
5	Eethen, Gable

Result: To get the age of dependent with rank '3', when this rule function is called on July 7, 2016, the age of dependent "Alaric, Anna" is returned at 5.6 months, representing 5 months and 6 days.

16.57 Get Amount from Previous Recurring Pay Component Record()

The function returns the value from the previous effective dated record that has the pay component recurring present. If the previous effective dated record does not have the pay component recurring present, then the function returns the value from the last record that had the pay component recurring.

Example

In a recurring pay compent, there is a field that displays the change percentage on the amount field.

For example, a user was hired with basic pay of 10,000 USD on 01/01/2018.

User ID	Pay Component ID	Amount	% of Change	Start Date	End Date	Currency
101	P1	10,000	-	01/01/2018	12/31/9999	USD

After a probationary period, the user receives a pay raise of 10% as of 03/01/2018. So admin inserts a new record with new amount.

User ID	Pay Component ID	Amount	% of Change	Start Date	End Date	Currency
101	P1	10,000	-	01/01/2018	12/31/9999	USD
101	P1	11,000	10	03/01/2018	12/31/9999	USD

However, the admin realizes that the user should have received a 20% pay raise. So admin corrects the record and changes the amount to 12k.

Here if you were to use the previous value function, then the system would have calculated the previous value as 11k, and hence % of change is calculated as (12k-11k)/11k ~= 9.01 %.

User ID	Pay Component ID	Amount	% of Change	Start Date	End Date	Currency
101	P1	10,000	-	01/01/2018	12/31/9999	USD
101	P1	12,000	9.01	03/01/2018	12/31/9999	USD

But the admin would really expect the system to show the correction rather than the difference.

So to calculate the correct percent, the system need to compare it with value from previous effective-dated record, which is 10k. (12k-10k)/10 = 20 %

User ID	Pay Component ID	Amount	% of Change	Start Date	End Date	Currency
101	P1	10,000	-	01/01/2018	12/31/9999	USD
101	P1	12,000	20	03/01/2018	12/31/9999	USD

16.58 Get Balance For Posting Types In Period()

The rule function returns the balance of postings in a given period.

You can decide which postings you are interested in based on the list of posting types. Examples include period-end processing postings and interim account update postings.

Input Parameters

For this parameter:	Make this entry:
Start Date	Enter the date the relevant period should start with.
End Date	Enter the date the relevant period should end with.
Time Account	Enter the time account parameter from the rule.
Posting Types	Select a list of relevant posting types.

Examples

Let's look at some examples.

We have a time account with these postings:

January 1, 2017	Accrual	+5 days
February 1, 2017	Accrual	+5 days
February 15, 2017	Manual Adjustment	+10 days
March 1, 2017	Accrual	+5 days
March 1, 2017	Interim Account Update	-1 day
April 1, 2017	Accrual	+5 days

Example 1: Run the rule function with these parameters:

- Start Date: January 1, 2017
- End Date: December 31, 2017
- Time Account: Time account with postings as listed in the table.
- Posting Types: Accrual, Manual Adjustment

The result would be 30 within this period - 20 accrual postings and 10 manual adjustment postings.

Example 2: Run the rule function with these parameters:

- Start Date: March 1, 2017
- End Date: March 1, 2017
- Time Account: Time account with postings as listed in the table.
- Posting Types: Interim Account Update

The result would be -1 postings within this period.

Use Cases

Especially for period-end processing rules or interim account update rules, it might be useful to know whether there is already such a posting on the corresponding time account. By using this rule function in combination with the rule function Get Number Of Postings For Posting Types in Period() [page 269], you can consider the existing posting during the creation of the new one.

Example with interim account update: The current interim account should ensure that 20 hours are deducted from each time account. The first time you run this rule, the corresponding posting will be created. If you run this interim rule again for the same time account, again 20 hours are deducted. However, you can use this rule function to determine whether there is already an interim account update posting for this date on the time account and, if there is, skip this account based on this information or you can compare the result of this rule function with the 20 hours that should be deducted. If there is, for example, an interim account update that deducts 15 hours, you can add a second interim account update that deducts 5 hours to get to the total of 20 hours to be deducted.

Restrictions

⚠ Restriction

If you want to use the rule function when determining the *Amount posted* in a *Create Time Account Detail* statement of a rule (for example, an accrual rule, accrual transfer rule, period-end processing rule and so forth) you shouldn't use the rule function directly on that field. Instead, use the rule function only indirectly, by using a variable:

1. Create a variable that stores the result of the rule function.
2. In the *Create Time Account Details* statement, under *Amount posted*, refer to the **variable**, which has been defined in the previous step (instead of referring to the rule function).

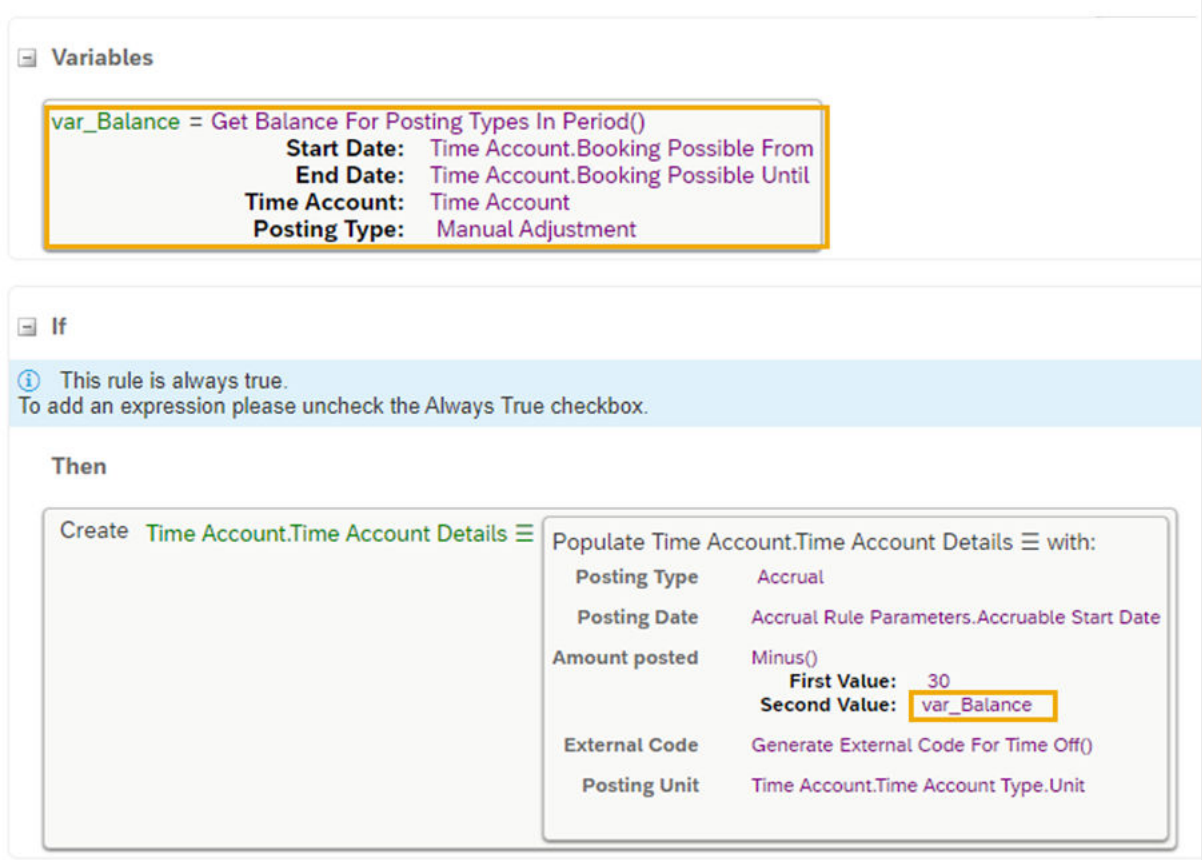
The reason for applying this method is that if the rule function is called during the transaction of creating a time account detail, that time account detail has already been created. However, the properties of the time account detail (including *Amount posted*) are still empty. This can lead the rule function to fail. By following these steps, if the rule function is called before the time account detail has been created, that time account detail isn't considered at this point. The rule function then works as expected.

→ Tip

We recommend that you don't use this rule function in any termination rules. When a termination is carried out in an accrual period, the balance for the accrual period is recalculated and set to zero, and the prorated accrual is calculated. This rule function would then return the balance for the current accrual period as zero.

→ Remember

This is how the rule function should be used within the rule. You must create a **variable** that stores the result of the rule function, and refer to the **variable** under *Amount posted*.



If you use the rule function in an accrual rule scenario and query the balance based on the accrual or recalculation posting type, please keep the following in mind. During recalculation scenarios, existing accrual and recalculation postings on the time account are removed before the rule is called. For that reason, the rule can falsify the result as those postings won't be considered by the rule function.

16.59 Get Completed Calendar Weeks Between Dates (ISO Standard)()

This function calculates the number of weeks the employee is going to work in the accrual period.

Overview

This function assumes that the week starts on a Monday and that the year has 52 weeks (as defined by ISO standard). If the accruable start date is not on a Monday, that week does not count. If the accruable end date is not a Sunday, that week does not count. The function is used for fields that show data in numeric form.

The accrual calculation is based on the number of weeks the employee is going to work, This rule function can used for the hire accrual rule, termination accrual rule, and normal accrual rule.

You can only use this function in relation to time account types where you select *Annual* as the accrual frequency.

Manage Parameters

When you click *Manage Parameters*, make the following entries:

Code	Name	Object
accrualRuleParameters	accrualRuleParameters	Accrual Rule Parameters

Input Parameters

Here's a table showing the entries you need to make in the input parameters.

For this parameter:	Make this entry
Start Date	Please use Manage Parameters accrualRuleParameters and select Start Date.
End Date	Please use Manage Parameters accrualRuleParameters and select End Date.
Accruable Start Date	Please use Manage Parameters accrualRuleParameters and select Accruable Start Date.
Accruable End Date	Please use Manage Parameters accrualRuleParameters and select Accruable End Date.

Examples

Let's take a look at some examples:

Example 1: Hire and termination within accrual period

Start Date: January 1, 2013

End Date: December 31, 2013

Accruable Start Date: February 1, 2013

Accruable End Date = October 31, 2013

Result 1

- February 1 is in calendar week 5 and it is a Friday, which means the week does not count. Counting therefore starts at week 6.

- October 31 is in calendar week 44 and it is a Thursday, which means the week does not count.
- The calculated number of weeks will be (43–5=) 38.

Example 2: Termination within accrual period

Start Date: January 1, 2013

End Date: December 31, 2013

Accruable Start Date: January 1, 2013

Accruable End Date = October 31, 2013 (termination date – 1)

Result 2

- October 31 is in calendar week 44 and it is a Thursday, which means the week does not count.
- The calculated number of weeks will be (44–1=) 43.

16.60 Get Completed Remaining Calendar Weeks (ISO Standard)()

This function calculates the number of weeks the employee is going to work in his hire period.

Overview

This function assumes that the week starts on a Monday, the year has 52 weeks (as defined by ISO standard), and the weeks until the hire week are subtracted from those 52 weeks. If the hire date is not on a Monday, that week is also subtracted during calculation The function is used for fields that show data in numeric form.

You can only use this function in relation to time account types where you select *Annual* as the accrual frequency.

Manage Parameters

When you click *Manage Parameters*, make the following entries:

Code	Name	Object
accrualRuleParameters	accrualRuleParameters	Accrual Rule Parameters

Input Parameters

Here's a table showing the entries you need to make in the input parameters.

For this parameter:	Make this entry
Recruit Date	Select the User Field – Employment Details – Recruit Date.
Start Date	Please use Manage Parameters accrualRuleParameters and select Start Date.
End Date	Please use Manage Parameters accrualRuleParameters and select End Date.

Examples

Let's take a look at some examples:

Example 1

Recruit Date = 01.05.2013 Start Date = 01.01.2013 End Date = 31.12.2013 – 01.05.2013 is calendar week 18, but it is a Wednesday, therefore the week does not count – Number of weeks, which will be returned = 52 – 18 => 34 weeks

Recruit Date: May 1, 2013 (calendar week 18)

Start Date: January 1, 2013

End Date = December 31, 2013

Result 1: May 1 is in week 18, but it is a Wednesday. The calculated number of remaining weeks is therefore (52–34=) 18 weeks.

Example 2

Recruit Date = 01.04.2013, Start Date = 01.01.2013, End Date = 31.12.2013 – 01.04.2013 is calendar week 14, which is a Monday, therefore the week counts completely – Number of weeks which will be returned = 52 – 13 => 39 weeks

Recruit Date: April 1, 2013 (calendar week 14)

Start Date: January 1, 2013

End Date = December 31, 2013

Result 2: April 1 is in week 14, and that's a Monday, so the week is considered in the calculation. The calculated number of remaining weeks is therefore (52–13=) 39 weeks.

16.61 Get Completed Weeks Between Dates

This function calculates the number of completed weeks within a period. The count starts on the first week start day in the period and ends on the last week start day minus 1 day in the period.

Input Parameters

For this parameter...	Make this entry:
Start Date	Enter or select the start date to calculate the completed weeks. The count starts on the week start day within that week that you have defined in the *Week Start Day* field.
End Date	Enter or select the start date to calculate the completed weeks. The count ends one day before the week start day that you have defined in the *Week Start Day* field.
Week Start Day	Enter a number between 1-7 to define on which day the week starts: **1** = Monday, **7** = Sunday This defines when the count starts and ends.

Example

Start Date: 05/01/2013

End Date: 12/31/2013

Week Start Day: 2 (this means the week end day is 1=Monday)

In this example, 05/01/2013 is a Wednesday (=3), therefore the count starts on the next Tuesday (05/07/2013) as defined in the *Week Start Day* field. 12/31/2013 is a Tuesday (=2), therefore the count ends on the last Monday (12/30/2013). The number of weeks which will be returned is calculated as follows:

Number of days starting at 05/07/2013 – 12/30/2013 divided by 7 = 34 weeks

16.62 Get Completed Months Of Time Types In Period()

This rule function calculates the number of complete months for which the employee was absent in a given period for the time types provided. The calculation is based on **full** calendar months covered during the absence and does not include partial months.

The rule function can be used in accrual rules for time account types where the accrual to be posted needs to consider the impact of other absences including full months that an employee has taken in the specified time frame. Each absence is considered separately and a month is only counted if it is fully covered by one absence.

Please refer to the second example for more details. For cross-midnight absences, the absence is considered based on the working day to which the absence belongs to. Please refer to the third example for more details.

Input Parameters

For This Field	Make This Entry
User ID	Select the *User* field. This is the user for which the total number of absence days should be calculated.
Start Date	First day of duration in which absences are to be considered for calculation. You can use the start date of accrualRuleParameters.
End Date	Last day of duration in which absences are to be considered for calculation. You can use the end date of accrualRuleParameters.
Time Types	Select time types. These are used to select employee times for calculation. **Note** Please only use time types with the time type category *Absence* for this rule function. Otherwise it can lead to errors.

Examples

Example 1

Here is data to calculate number of complete months covered by leave without pay leave (Time Type: LWOP) and unauthorized leave (Time Type: UNAL) during year 2015 during accrual rule processing:

User ID: A.N. Other

Start Date = January 1, 2015

End Date = December 31, 2015

Time Types: “LWOP”, “UNAL”

Example Details 1

The employee has taken the following absences during 2015:

- Leave without pay from February 1, 2015, to March 16, 2015.
- Leave without pay from April 3, 2015, to May 4, 2015.
- Unauthorized leave from September 28, 2015, to December 12, 2015.

Result 1: The rule function returns 3 because February, October and November months are fully covered by absences of time types LWOP and UNAL. March, April, May, September and December months have absences of time types LWOP and UNAL, but these absences do not cover these months fully.

Example Details 2

The employee has taken the following absences during 2015:

- Leave without pay from February 1, 2015, to March 16, 2015.
- Leave without pay from March 17, 2015, to March 31, 2015.

Result 2: The rule function returns 1 because each Employee Time is considered separately. This means that March is not covered fully.

Example 3: Cross Midnight

Here is data to calculate the number of complete months covered by leave without pay leave (Time Type: LWOP) and unauthorized leave (Time Type: UNAL) during year 2020 during accrual rule processing:

User ID: A.N. Other

Start Date = January 1, 2020

End Date = December 31, 2020

Time Types: “LWOP”, “UNAL”

Prerequisites: User A.N. Other has a cross-midnight work schedule. Monday to Friday he is working from 10 p.m. to 6 a.m. Saturday and Sunday are non working days.

Example Details 3:

The employee has taken the following absences during 2020:

- Leave without pay from February 20, 2020 (10 p.m.), to March 30, 2020 (this absence includes the shift starting on March 30, means physically the absence ends on March 31 at 6 a.m.)
- Unauthorized leave from April 1, 2020 (10 p.m.), to May 4, 2020.

Result 3: The rule function returns 1 because only April is fully covered by time type UNAL. Although there is a part of the working day for April for midnight until 6am, which is not covered by the absence, the month is completely counted because this part belongs to the day the shift starts on - March 31, 2020). March is not fully covered because March 31, 2020 is not part of the first absence request.

16.63 Get Count of Dependents()

This rule function gets the count of dependents for a given list of relationship types on a given date.

The rule function returns the count of the dependents. Dependents with null date of birth are included in the count.

Input Parameters

For this parameter	Make this entry
Person ID External	Person ID external, accessed from the biographical information on the rule UI.
User ID	User ID for which youngest dependent is calculated.
Date	The date on which youngest dependent is fetched.
Relationship Types	List of relationship types for which dependents will be fetched. The values shown are the external codes of the options available in the relationship type picklist.

Examples

Let's look at an example.

Employee is hired on January 1, 2010, so the dependent card has following data for the employee's dependents:

Relationship Type	Date of Birth	Last Name	First Name
Child	September 1, 2009	Klarc	Adam
Child	September 1, 2009	Calley	Bob
Child		Eethen	Alen
Child		Eethen	Gable
Child	January 27, 2016	Alaric	Anna

Result: When this rule function is called, the count returned is 5.

16.64 Get Current DateTime()

This function produces a timestamp using the current server date and time.

Example

- **Current server date:** 01/31/2018
- **Current server time:** 14:39:42
- **Result:** 2018-01-31 14:39:42

Example

Since the timestamp is automatically based on the server date and time, there are no input parameters required for this rule function.

16.65 Get Date of Birth Of Youngest Dependent()

This rule function returns the date of birth of the youngest dependent for a particular relationship type.

The rule function takes account only of those dependents whose date of birth exists in the system. Dependents are sorted in ascending order of date of birth. If there are two or more dependents with the same date of birth, such as twins or triplets, sorting is done based on last name followed by first name. The dependent at the highest index in the sorted list of dependent is the youngest dependent.

The rule function is required for the use case where employees can be granted maternity, child care, or paternal leave if the dependent is born in a certain time period.

Input Parameters

For this parameter	Make this entry
Person ID External	Person ID external, accessed from the biographical information on the rule UI.
User ID	User ID for which youngest dependent is calculated.
Date	The date on which youngest dependent is fetched. **Note** This parameter is optional, but we **strongly** recommend that you specify a valid date here.

For this parameter	Make this entry
Relationship Types	List of relationship types for which dependents will be fetched. The values shown are the external codes of the options available in the relationship type picklist.

Validation of Input Parameters

- You must enter either an Employee Person ID or a User ID or both. If you don't, an exception is raised.
- Date is an optional parameter. If no date is entered, the current date is used for the calculation. We strongly recommend that you specify a valid date here, even if that date is the current date.
- Relationship Types is a required parameter. If none is entered, an exception is raised.

Examples

Let's look at an example.

Employee is hired on January 1, 2010, so the dependent card has following data for the employee's dependents:

Relationship Type	Date of Birth	Last Name	First Name
Child	September 1, 2009	Klarc	Adam
Child	September 1, 2009	Calley	Bob
Child		Eethen	Alen
Child		Eethen	Gable
Child	January 27, 2016	Alaric	Anna

Sorting results in the following ranking:

Sorted Dependents

Rank	Dependent
1	Klarc, Adam
2	Calley, Bob
3	Alaric, Anna
4	Eethen, Alen
5	Eethen, Gable

Result: When this rule function is called, it returns the youngest dependent's date of birth - that is January 27, 2016, for Anna Alaric.

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

Determine the date on which the maximum total FTE across multiple employments occurs.

> **Note**
>
> This rule function is only applicable for concurrent employment scenarios. Do not use it within the context of global assignment or adding a new hire.

Example

Let's say you have the following employees with multiple employments. Here's what you'll get when you run this rule function:

Example of "Get Maximum Total FTE for Time Period()"

-	Employee A	Employee B
Employment 1	0.5 FTE active from 2012-12-22	1 FTE active from 2016-01-01
Employment 2	0.25 FTE active between 2015-01-01 and 2016-01-01	1 FTE active from 2018-02-01
Employment 3	0.25 FTE active from 2016-02-02	-
Person ID	Person ID of the employee from the rule context	Person ID of the employee from the rule context
From Date	2018-01-01	2018-01-01
To Date	2018-03-01	2018-03-01
Job Information	Job Information of the employee from the rule context	Job Information of the employee from the rule context
Result when you run "Get Maximum Total FTE for Time Period()"	2018-01-01	2018-02-01

Input Parameters

Input Parameters for "Get Maximum Total FTE for Time Period()"

For this parameter...	Of this type...	Which is...	Make this entry
Person ID	Text	Required	The Person ID of the employee for whom the FTE is being calculated.

For this parameter...	Of this type...	Which is...	Make this entry
From Date	Date	Required	Start of the time period for which the total FTE will be calculated.
To Date	Date	Required	End of the time period for which the total FTE will be calculated.
Job Information	Job Information	Required	The Job Information of the employee for whom the FTE is being calculated.

Use Case

This rule function will be most relevant to use in conjunction with "Get Maximum Total FTE for Time Period()". If you determine that FTE exceeds the threshold, you might be interested to know the exact date on which this happened.

You can also use either of the following to determine the date on which max FTE occurred:

- An "onSave" rule on JobInformation
- An "on Change" rule on the field FTE

Related Information

Get Maximum Total FTE for Time Period() [page 244]

16.67 Get Employee Status()

This rule function returns the external code of employee status for the input parameters User Id (mandatory) and date (optional). By default, the current date is considered. The rule function can be consumed as a take rule.

The return value is true for **Active** and false for **Inactive** employee status. The rule function can be consumed as a take rule.

This rule function is required for the use case where the employee is terminated and should not be allowed to raise further employee time. It generates an appropriate warning or error message.

Input Parameters

Here's a table showing the entries you need to make in the input parameters.

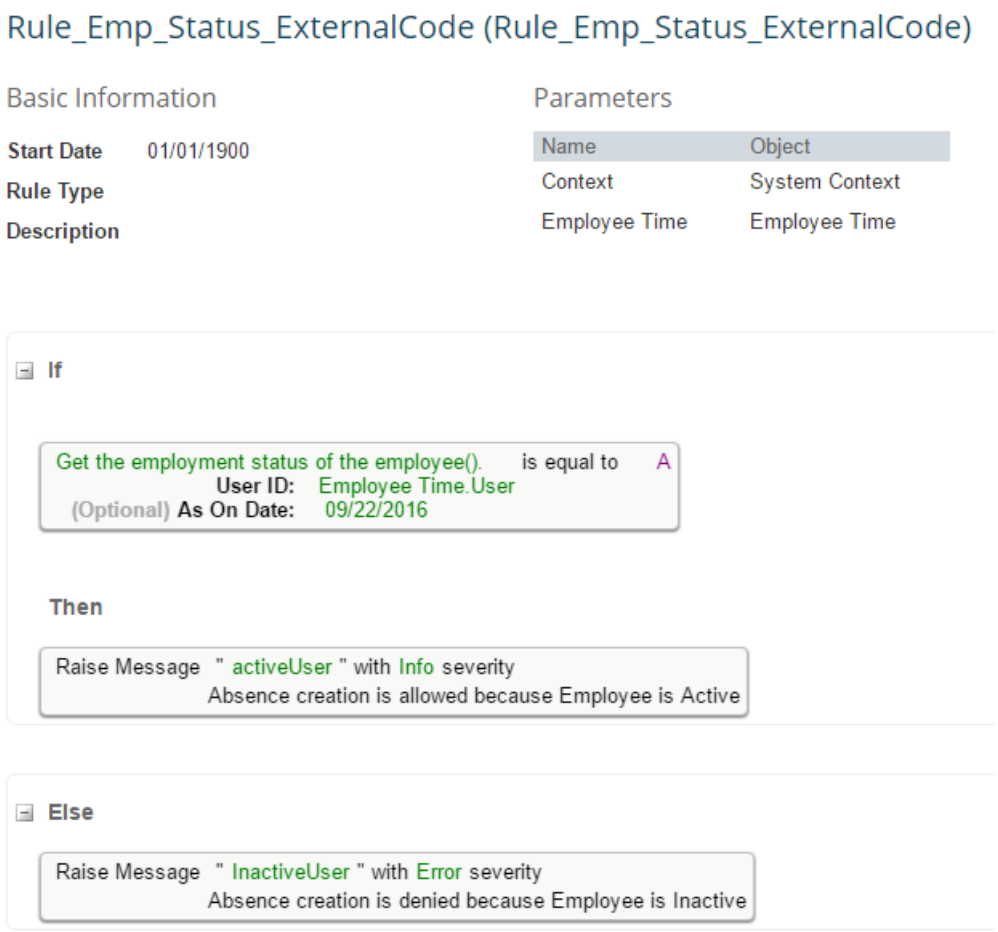
For this parameter:	Make this entry
User Id	User Id for employee whose status is to be fetched.
Date (optional)	Date at which employee status is to be fetched.

Validation of Input Parameters

- The User Id is a required parameter. If it is not provided, an exception is raised and an appropriate error is logged.
- The Date is an optional parameter. If it is not passed, the current date is used to fetch the employee status.

Example

Let's look at an example.



The employee is terminated on September 20, 2016. So the external code of employee status in the picklist will be equal to T.

The rule function as shown in the screenshot (configured as a take rule) will be executed when the user attempts to create an employee time for this employee. The IF condition in the above rule function will be false and the ELSE part will get executed.

Therefore, the error message "Absence creation is denied because Employee is Inactive" will be displayed and employee time creation will not be allowed.

16.68 Get End Time Of Working Day ()

This function calculates the end time of the working day on a given date.

It reads the work schedule associated with the user to calculate the end time of the date provided to the rule function. The rule function returns a value in 24 hour clock format.

If the user applied for time off using clock times, you can use this rule function for specific validation against the end of the day. This is typically done in a take rule (absence validation rule) that you can set to be triggered when absence is created or modified.

For cross-midnight working days, the returned end time is on the next calendar day. Please refer to the cross-midnight example below.

The rule function returns null if:

- A duration-based work schedule is assigned to the user on the requested date.
- The requested date is a non-working day.

Input Parameters

For this parameter:	Make this entry:
User	Select the *User* field. This is the user for which the end time of working day should be calculated.
Date	Date for which the end time is required

Example 1 - Non-Cross-Midnight Work Schedule

Let's look at an example.

Here's the data for A.N.Other's end time calculation:

- User ID: A.N. Other
- Date: Wednesday, July 1, 2020

Prerequisites: User A.N.Other should have the Clock Times time recording variant. The work schedule is configured with Monday to Friday as working days, from 08:00 to 17:00. Saturday and Sunday are non-working days.

Result: The rule function returns an end time of 17:00:00.

Example 2: Cross-Midnight Work Schedule

This time, A.N Other's information looks like this:

- User ID: A.N.Other
- Date: Monday, July 6, 2020

Prerequisites: User A.N.Other should have the Clock Times time recording variant. The work schedule is configured with Monday to Friday as working days, from 22:00 to 06:00. Saturday and Sunday are non-working days.

Result: The rule function returns an end time of 06:00:00, which is physically Saturday morning, but belongs to the shift that starts on Friday.

16.69 Get Expiration Date For Work Permit

This function determines the expiration date of a work permit.

In case an employee has more than one work permit of the same document type, the function returns the furthest expiration date. This rule function can be used for data changes to validate their compliance against the validity of a work permit.

> ⓘ Note
>
> Three parameters, Personal ID External, Country, and Document Type are passed. Invalid values of these parameters will cause this function to throw an exception, though users won't have a chance to notice it. The business rule that uses the function won't get executed as a result.

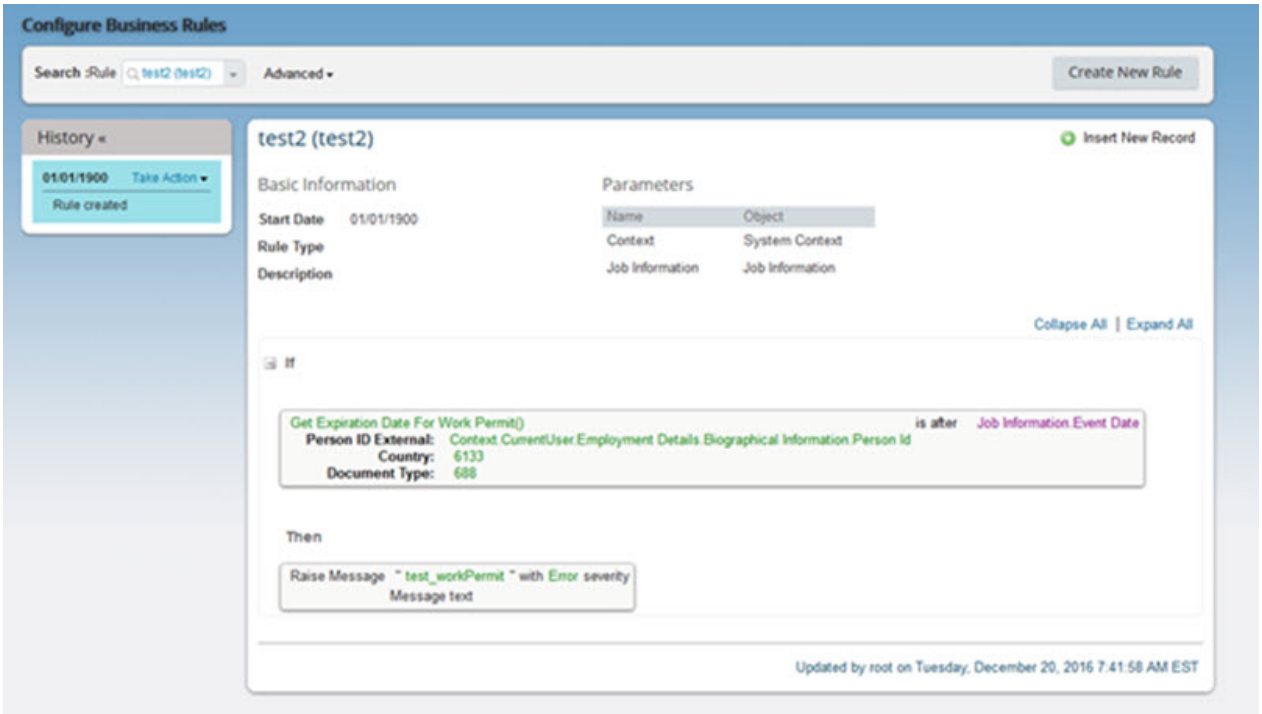
Input Parameters

For This Field	Of Type	Which Is	Make This Entry
Person ID External	Text	Mandatory	Select the Person ID (External) from the Biographical Information.

For This Field	Of Type	Which Is	Make This Entry
Country	Text	Mandatory	Check in hris-element workPermitInfo whether there is a picklist assigned to the country field. If yes, go to the *Admin Center* > *Picklists Management* and export all picklists. Enter the picklist option ID of the country (for example, “6133” for USA). - If not, enter the ISO Country Code (for example, “USA”).
Document Type	Text	Mandatory	Check in hris-element workPermitInfo to see which picklist is assigned to the document-type field. Go to the *Admin Center* > *Picklists Management* and export all picklists. Enter the picklist option ID of the document type (for example, “688” for Passport).

Use Case

In this use case, an employee has a work permit with country “USA” (6133) and of document type “Passport” (688). If a job info record of this employee is created or changed and the work permit expiration date returned by the rule function is after the job info event date, then an error message is shown.



16.70 Get First Day Of Month

This function finds the first day of the same month as the month in the specified date.

Input Parameters

For this parameter...	Make this entry:
Date	Select a field of the field type *Date* (for example: *Event Date*). The first day of the same month as in this date is returned as result. If "Null" is passed to the date, then current date is considered as default.

16.71 Get Fiscal Year Start or End Date

This function returns the fiscal year start or end date based on a given date and country/region.

Configuration Requirements

You can use this function when fiscal year is enabled in Provisioning.

> → Remember
>
> As a customer, you don't have access to Provisioning. To complete tasks in Provisioning, contact your implementation partner or Account Executive. For any non-implementation tasks, contact Technical Support.

Input Parameters

Input Parameter	Type	Required	User Entry
Country/Region	Text	Yes	Select the country or region for which you want to get fiscal year details.
Fiscal Year Details for Date	Date	Yes	Select the date for which you want to get the fiscal year start date or end date.
Date Type (Start Date or End Date)	Enum	Yes	Select either start date or end date as date type: • *Start Date* returns the start date of the fiscal year of the selected country/region and date. • *End Date* returns the end date of the fiscal year of the selected country/region and date.

Behavior in Case of Errors

The rule function returns "null" in the following scenarios:

- Invalid country/region code for *Country/Region* parameter

- Mapping is missing for the given country/region in the *Fiscal Year to Country/Region Map* object
- Incorrect configuration in the *Fiscal Year Variant* object attached to the given country/region

Example

Country/Region	Fiscal Year Details for Date	Date Type (Start Date or End Date)	Result	Details
Australia	March 23, 2020	Start Date	April 1, 2019	The start or end date returned by the rule function depends on the fiscal year configuration in the system.
India	October 19, 2020	End Date	March 31, 2019	The start or end date returned by the rule function depends on the fiscal year configuration in the system.

Use Case

An admin wants to check if the start date and the end date of the leave for which the advance leave payment is being requested falls in the same fiscal year or across two fiscal years. For more information about achieving this use case using the Get Fiscal Year Start or End Date rule function, refer to **Create a Business Rule for Fiscal Year Validation** in the Employee Central Time Management guide.

Related Information

Create a Business Rule for Fiscal Year Validation

16.72 Get Foundation Object Default Value

This function retrieves the default value of a foundation object at runtime, based on the defaulting configurations and input parameters given to the function.

Configuration Requirements

You can only use this function for conditional groups and defaults in Employee Central.

It is recommended that this function be used with *Set* action while creating a business rule.

Input Parameters

Input Parameter	Type	Required	User Entry
Default Field	DefaultingField	Yes	Select the foundation object default field that the rule function must return.
Job Information	jobinfo	No	Select the user Job Information to read the conditional fields for defaulting.
Position	Position	No	Select position of the user to read the conditional fields for defaulting.

Behavior in Case of Errors

If the **Default Field** input parameter isn't selected, the function returns a null value.

If both **Job Information** and **Postion** input parameters are selected, the function returns a null value. If neither of the input parameters are selected, the function returns a null value.

Use Case

You want to default a foundation object field in Position or HRIS elements, by creating a rule that uses this function to retrieve the configured default value of the foundation object field based on conditional groups and defaults. The rule can be configured for OnInit, OnSave, and OnChange events.

Example

If you want to default the **Location** field in Position, you can create a business rule which uses the function **Get Foundation Object Default Value ()** which calculates and returns the default value for Location based on the values for condition fields and data configured in these objects: Default, Employer, and Employee Groups.

Related Information

Creating Business Rules for Conditional Defaults

16.73 Get Generic Object Default Value

This function retrieves the default value of a generic object at runtime, based on the defaulting configurations and input parameters given to the function.

Configuration Requirements

You can only use this function for conditional groups and defaults in Employee Central.

It is recommended that this function be used with *Set* action while creating a business rule.

Input Parameters

Input Parameter	Type	Required	User Entry
Default Field	DefaultingField	Yes	Select the generic object default field that the rule function must return.
Job Information	jobinfo	No	Select the user Job Information to read the conditional fields for defaulting.
Position	Position	No	Select Position of the user to read the conditional fields for defaulting.

Behavior in Case of Errors

If the **Default Field** input parameter isn't selected, the function returns a null value.

If both **Job Information** and **Postion** input parameters are selected, the function returns a null value. If neither of the input parameters are selected, the function returns a null value.

Use Case

You want to default a generic object field in Position or HRIS elements, by creating a rule that uses this function to retrieve the configured default value of the generic object field based on conditional groups and defaults. The rule can be configured for OnInit, OnSave, and OnChange events.

Example

If you want to default the **Paygroup** field in Position, you can create a business rule which uses the function **Get Generic Object Default Value ()** which calculates and returns the default value for Paygroup based on the values for condition fields and data configured in these objects: Default, Employer, and Employee Groups.

Related Information

Creating Business Rules for Conditional Defaults

16.74 Get Incumbent By Position

This function returns the user ID of the incumbent of the position. If more than one incumbent is assigned, only one is returned.

Limitations

This function is only available if Position Management is activated.

Use Case

You can find a rule example in the Position Management Guide, under *Rule Functions in Position Management*.

16.75 Get Job Info Date Field Value On Key Date()

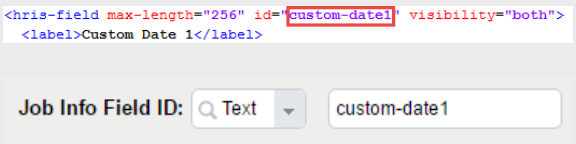
This rule function reads a job info date field for a specific key date.

The rule function is needed because, if you navigate directly to the job info date field to retrieve it without this rule function, you cannot specify the key date the field should be returned for. The job information is effective dated and might have multiple records within an accrual period. If you navigate, the first job info record within the accruable period will be returned. If, for example, you want to get the field “contract end date”, you would like to get the latest record in the accrual period, not the first one.

Limitations

- Use this rule function only in the Accrual scenario.
- You cannot read any job information outside the employee's accrual period.

Input Parameters

For this parameter	Make this entry
User	Select the *User* field. This is the user the job info date will be read for.
Job Info Field ID	**⚠ Caution** Please enter the field ID from your data model. Type it in manually - do **not** navigate to the corresponding job information field.  The field in your data model must be a date field to be returned. For example: custom-date1.
Date	Select a key date, such as Accrual End Date from accrualRule-Parameters.

Examples

Let's look at an example.

User: A.N. Other

Job Info Field ID: custom-date1

Date: December 31, 2017

Job Information History for this user:

- Effective start date January 1, 2010; custom-date1 = December 31, 2016
- Effective start date January 1, 2017; custom-date1 = May 31, 2017
- Effective start date June 1, 2017; custom-date1 = November 30, 2017

Result

The custom-date1 field from the job information valid on December 31, 2017 is November 30, 2017.

16.76 Get Job Info Numeric Field Value On Key Date()

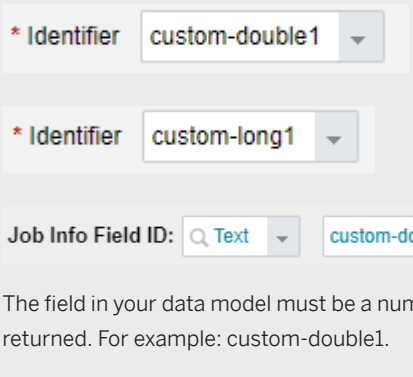
This rule function reads a job info numeric field for a specific key date.

The rule function is needed because, if you navigate directly to the job info numeric field to retrieve it without this rule function, you cannot specify the key date the field should be returned for. The job information is effective dated and might have multiple records within an accrual period. If you navigate, the first job info record within the accruable period will be returned. If, for example, you want to get the field “working days per week”, you would like to get the latest record in the accrual period, not the first one.

Limitations

- Use this rule function only in the Accrual scenario.
- You cannot read any job information outside the employee's accrual period.

Input Parameters

For this parameter	Make this entry
User	Enter a user ID.
Job Info Field ID	⚠ Caution Please enter the field ID from your data model. Type it in manually - do not navigate to the corresponding job information field.  The field in your data model must be a numeric field to be returned. For example: custom-double1.

For this parameter	Make this entry
Date	Select a key date, such as Accrual End Date, from accrualRule-Parameters.

Example

Let|s look at an example.

User: A.N. Other

Job Info Field ID: custom-double1

Date: December 31, 2017

Job Information History for this user:

- Effective start date January 1, 2010; custom-double1 = 12
- Effective start date January 1, 2017; custom-double1 = 14
- Effective start date June 1, 2017; custom-double1 = 16

Result

The custom-double1 field from the job information valid on December 31, 2017 is 16.

16.77 Get Job Info String Field Value On Key Date()

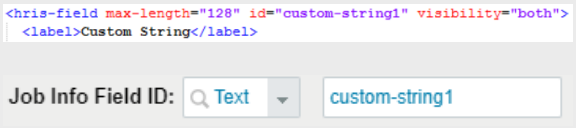
This rule function reads a job info string field for a specific key date.

The rule function is needed because, if you navigate directly to the job info string field to retrieve it without this rule function, you cannot specify the key date the field should be returned for. The job information is effective dated and might have multiple records within an accrual period. If you navigate, the first job info record within the accruable period will be returned. If, for example, you want to get the field “employee class”, you would like to get the latest record in the accrual period, not the first one.

Limitations

- Use this rule function only in the Accrual scenario.
- You cannot read any job information outside the employee's accrual period.
- Only fields directly under the Job Information heading can be used. Fields from the subsections cannot be used.

Input Parameters

For this parameter	Make this entry
User	Select the *User* field. This is the user the job info date will be read for.
Job Info Field ID	**⚠ Caution** Please enter the field ID from your data model. Type it in manually - do **not** navigate to the corresponding job information field.  The field in your data model must be a string field to be returned. For example: custom-string1. Only fields directly under the Job Information heading can be used here. Fields from the subsections cannot be used.
Date	Select a key date, such as Accrual End Date from accrualRule-Parameters.

Examples

Let's look at an example.

User: A.N. Other

Job Info Field ID: custom-string1

Date: December 31, 2017

Job Information History for this user:

- Effective start date January 1, 2010; custom-string1 = value1
- Effective start date January 1, 2017; custom-string1 = value2
- Effective start date June 1, 2017; custom-string1 = value3

Result

The custom-string1 field from the job information valid on December 31, 2017 is value3.

16.78 Get Job Location()

This function returns job location data from the Job Location generic object. You can use this function to set the value of the standard Job Location field on job requisitions, based on existing data in the requisition.

This function enables you to create business rules that help your organization adopt the Unified Data Model by ensuring that the standard Job Location field is set correctly and aligned with other location data in the requisition.

Configuration Requirements

This function is only available in the Job Requisition scenario for Recruiting, in business rules for a specified job requisition template. It isn't available if you choose "Job Requisition (All)" as the base object of the rule.

For more information about how to use this function, refer to Setting the Job Location Field on Job Requisitions with a Business Rule.

Input Parameters

Input Parameter	Type	Required	User Entry
Job Requisition	Object	Yes	User selects the "Job Requisition Template" object in a menu. No other selection is allowed.

Behavior in Case of Errors

The function returns a null value if there's no matching job location found in the Job Location object. The function tries to match existing location data in other fields to an existing job location in the Job Location object. If it can't find a match, it returns a null value.

Example

Here's an example of a Save rule that uses the function to set the value of the Job Location field if the field is null.

Use Case

Use this rule function to maintain location data on job requisitions in a standardized way, as required by the Unified Data Model for SAP SuccessFactors Recruiting.

In job requisitions that were created before adoption of the Unified Data Model, location information was captured in various ways, such as:

- Standard fields for Country, State, City, and Postal Code
  Location object in Employee Central (`location_obj`)
  Custom fields of different types (text, picklist, object, enum)

To support the Unified Data Model, we introduced a standard way to capture location information on a job requisition— the standard Job Location generic object and standard Job Location field (sfstd_jobLocation_obj). You can use this function to create a business rule that sets a value in the **new**, standard Job Location field, based on existing data in the other **preexisting** fields on each requisition.

> **Note**
>
> For more information about this process, see Migrating Job Location Data for the Unified Data Model

16.79 Get Level of Entity Within Company Structure

The rule function calculates the level of a certain entity used in company structure definitions.

It is important that you define the parameters for the IF statement as mentioned in the parameter table below.

If the conditions are met, the rule triggers the desired operation.

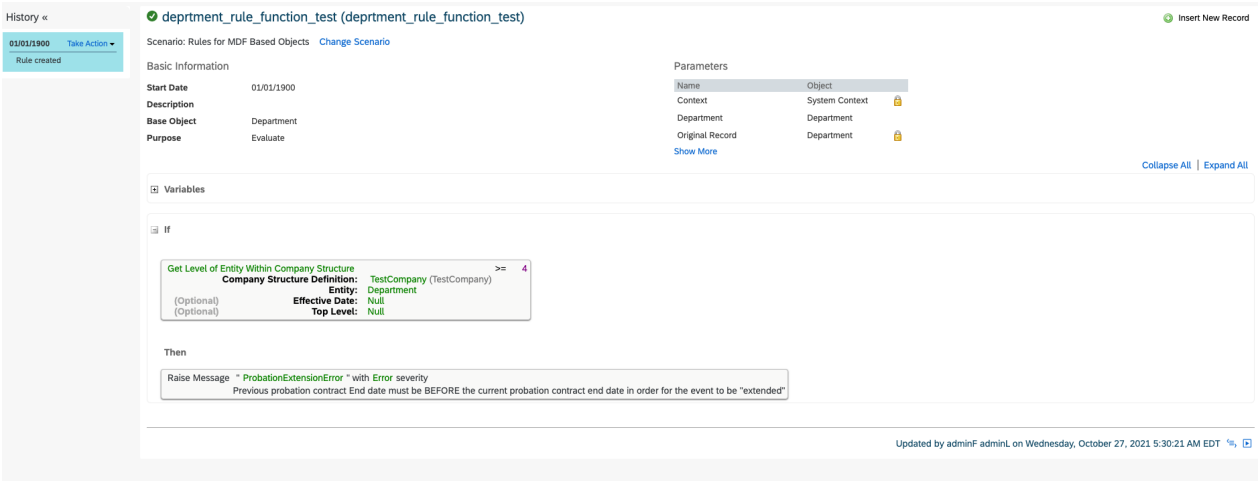
For example: If the rule function is used to verify that the company structure does not exceed a desired number of levels and you want to raise an error message if that number is exceeded, you must proceed as described in the *Make this entry:* column.

Input Parameters

For this parameter..	Of type...	Which is...	Make this entry:
Company Structure Definition	Text Value	Required	The company structure that is used to calculate the level.
Entity	Text Value	Required	The entity instance for which the level will be calculated. Example: Department.
Effective	Number	Optional	The date on which the level of the entity instance is calculated. If you don't enter anything, the current date is used.
Top Level	Number	Optional	The level of the root object in your company structure. If you don't enter anything, level 1 is used.

Additionally, you must define the THEN statement to raise the error message and add the rule as onSave rule, for example to the department object. The level is counted from the root object, such as a Business Unit, and depending on the Top Level number you have defined in the business rule.

Here's an example of how the rule could look:



16.80 Get Legacy Picklist Default Value

This function retrieves the default value of a picklist for HRIS entities at runtime based on the defaulting configurations and input parameters given to the function

Configuration Requirements

You can only use this function for conditional groups and defaults in Employee Central.

It is recommended that this function be used with *Set* action while creating a business rule.

Input Parameters

Input Parameter	Type	Required	User Entry
Default Field	DefaultingField	Yes	Select the picklist default field that the rule function must return.
Job Information	jobinfo	No	Select the user Job Information to read the conditional fields for defualting.
Position	Position	No	Select position of the user to read the conditional fields for defaulting.

Behavior in Case of Errors

If the **Default Field** input parameter isn't selected, the function returns a null value.

If both **Job Information** and **Postion** input parameters are selected, the function returns a null value. If neither of the input parameters are selected, the function returns a null value.

Use Case

You want to default a picklist field in an HRIS element such as Job Information, by creating a rule that uses this function to retrieve the configured default value of the picklist based on conditional groups and defaults. The rule can be configured for OnInit, OnSave, and OnChange events.

Example

If you want to default the **Employee Class** field in **Jobinfo**, you can create a business rule which uses the function **Get Legacy Picklist Default Value ()** which calculates and returns the default value for Employee Class based on the values for condition fields and data configured in these objects: Default, Employer, and Employee Groups.

Related Information

Creating Business Rules for Conditional Defaults

16.81 Get Local Date of DateTime()

This rule function gets the local date of a UTC timestamp. The date depends on the time zone offset you specify.

Input Parameters

Input Parameter	Type	Required	User Entry
DateTime	DateTime	Yes	Enter the timestamp.
Offset of Local Time Zone from UTC in Hours	Decimal	No	Enter the offset of the local time zone from UTC in hours. Any decimal value is allowed. If left empty, time zone UTC is assumed.

Behavior in Case of Errors

If DateTime is "null", the rule stops with status FAIL.

Example

- *DateTime*: 01/01/2017 07:00:00 UTC
- *Offset of Local Time Zone from UTC in Hours*: 1.0
- Result: 01/01/2017

Example

- *DateTime*: 01/01/2017 07:00:00 UTC
- *Offset of Local Time Zone from UTC in Hours*:-8.0
- Result: 12/31/2016

16.82 Get Local Time of DateTime()

This rule function gets the local time of a UTC timestamp.

Example

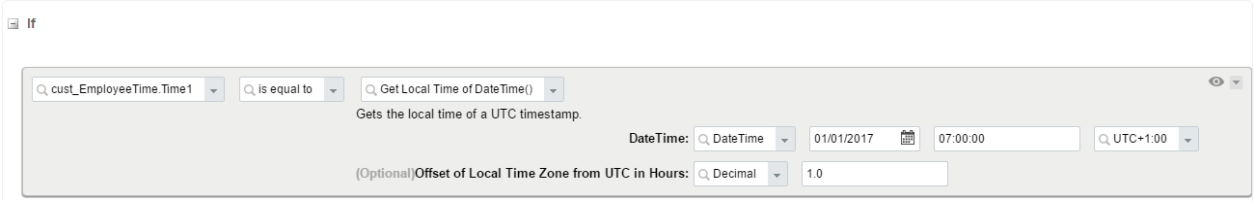
- *DateTime*: 01/01/2017 07:00:00 UTC
- *Offset of Local Time Zone from UTC in Hours*: 1.0
- Result: 08:00:00

Example

- *DateTime*: 01/01/2017 07:00:00 UTC
- *Offset of Local Time Zone from UTC in Hours*:-8.0
- Result: 23:00:00

Input Parameters

For this parameter...	Of type...	Which is...	Make this entry:
DateTime	DateTime	Required	Enter the timestamp.
Offset of Local Time Zone from UTC in Hours	Decimal	Optional	Enter the offset of the local time zone from UTC in hours. Any decimal value is allowed. If left empty, time zone UTC is assumed.



16.83 Get Matrix Position Code By Type

This function returns the code of the matrix position that is assigned by the specified type.

Limitations

This function is only available if Position Management is activated.

Use Case

You can find a rule example in the Position Management Guide, under *Rule Functions in Position Management*.

16.84 Get Maximum Total FTE for Time Period()

Determine the maximum total FTE that occurs across multiple employments during a given time period. If the start date and end date of the time period are the same, it returns the total FTE as of the given date. If the FTE value is not available in the employee's Job Information, then it will be derived from standard working hours.

> **Note**
>
> This rule function is only applicable for concurrent employment scenarios. Do not use it within the context of global assignment or adding a new hire.

Example

Let's say you have the following employees with multiple employments. Here's what you'll get when you run this rule function:

Example of "Get Maximum Total FTE for Time Period()"

-	Employee A	Employee B	Employee C
Employment 1	0.5 FTE active from 2012-12-22	1 FTE active from 2016-01-01	1 FTE active from 2016-01-01
Employment 2	0.25 FTE active between 2015-01-01 and 2016-01-01	1 FTE active from 2018-02-01	1 FTE active from 2018-02-01
Employment 3	0.25 FTE active from 2016-02-02	-	-

-	Employee A	Employee B	Employee C
Person ID	Person ID of the employee from the rule context	Person ID of the employee from the rule context	Person ID of the employee from the rule context
From Date	2018-01-01	2018-01-01	2018-01-01
To Date	2018-03-01	2018-03-01	2018-01-01
Job Information	Job Information of the employee from the rule context	Job Information of the employee from the rule context	Job Information of the employee from the rule context
Result when you run "Get Maximum Total FTE for Time Period()"	0.75	2	1

Input Parameters

Input Parameters for "Get Maximum Total FTE for Time Period()"

For this parameter...	Of this type...	Which is...	Make this entry
Person ID	Text	Required	The Person ID of the employee for whom the FTE is being calculated.
From Date	Date	Required	Start of the time period for which the total FTE will be calculated.
To Date	Date	Required	End of the time period for which the total FTE will be calculated.
Job Information	Job Information	Required	The Job Information of the employee for whom the FTE is being calculated.

Use Case

If you use Concurrent Employment features, you can create rules that determine how much FTE can be allocated to an employee across all assignments. It's possible that you don't want the total FTE to exceed 1 for all assignments, or you might want to specify a larger maximum. You can use an "onSave" rule on JobInformation to calculate the maxinum FTE, and then validate this value against your FTE limit. You can also set up an "onChange" rule on the field FTE that will verify the FTE limit each time this value is changed.

You can choose to determine the FTE from the employment hire date or can choose another time period that includes future employments.

Related Information

Get Maximum Total FTE for Time Period() [page 244]

16.85 Get MDF Picklist Default Value

This function retrieves the default value of the MDF picklist at runtime based on the defaulting configurations and input parameters given to the function.

Configuration Requirements

You can only use this function for conditional groups and defaults in Employee Central.

It is recommended that this function be used with *Set* action while creating a business rule.

Input Parameters

Input Parameter	Type	Required	User Entry
Default Field	DefaultingField	Yes	Select the picklist default field that the rule function must return.
Job Information	jobinfo	No	Select the user Job Information to read the conditional fields for defaulting.
Position	Position	No	Select position of the user to read the conditional fields for defaulting.

Behavior in Case of Errors

If the **Default Field** input parameter isn't selected, the function returns a null value.

If both **Job Information** and **Postion** input parameters are selected, the function returns a null value. If neither of the input parameters are selected, the function returns a null value.

Use Case

You want to default a picklist field in Position Information, by creating a rule that uses this function to retrieve the configured default value of the picklist based on conditional groups and defaults. The rule can be configured for OnInit, OnSave, and OnChange events.

Example

If you want to default the **Employee Class** field in **Position**, you can create a business rule which uses the function **Get MDF Picklist Default Value ()** which calculates and returns the default value for Employee Class based on the values for condition fields and data configured in these objects: Default, Employer, and Employee Groups.

Related Information

Creating Business Rules for Conditional Defaults

16.86 Get Month Name

Returns the full name of the current month.

Use Case

If a rule is executed on 08/28/2024, the rule function *Get Month Name()* returns "August".

16.87 Get Month Name Short

Returns the short name of the current month.

Use Case

If a rule is executed on 08/28/2024, the rule function *Get Month Name Short()* returns "Aug".

16.88 Get Months From Hire Date Taking Account Of Threshold

This function calculates the number of months the employee is going to work in the hire period. The threshold indicates whether the hire month is included in the calculation.

Overview

The function can be used for hire rules. The function is used for fields that show data in numeric form. Threshold means, for example, that the hire month counts if the employee's start day is on or before the 15th of the month. In this case, enter "15" as the threshold.

You can only use this function in relation to time account types where you select the **`Annual`** accrual frequency.

Manage Parameters

When you click *Manage Parameters*, make the following entries:

Code	Name	Object
accrualRuleParameters	accrualRuleParameters	Accrual Rule Parameters

Input Parameters

Here's a table showing the entries you need to make in the input parameters.

For this parameter:	Make this entry
Start Date	Please use Manage Parameters accrualRuleParameters and select Start Date.
End Date	Please use Manage Parameters accrualRuleParameters and select End Date.
Recruitment Date	Select The User Field – Employment Details – Recruit Date.
Threshold	This indicates the number of days for which the hire month counts. For example, a threshold of 15 means the hire month counts if the employee is hired in the first 15 days of the month.

Example

Let's take a look at an example:

Accrual Start Date: January 1, 2016

Accrual End Date = December 31, 2016

Threshold: 15

Every month starts on the first.

Case 1

Recruitment Date; June 15, 2016

Threshold date: June 15, 2016

Recruitment date June 15, 2016, is within the countable period.

Result

Number of months returned will be 7 (June 1, 2016 to December 31, 2016).

Case 2

Recruitment Date; June 16, 2016

Threshold date: June 15, 2016

Recruitment date June 16, 2013, is **not** within the countable period.

Result

Number of months returned will be 6 (July 1, 2016 to December 31, 2016).

16.89 Get Months Taking Account Of Threshold()

This function calculates the number of months the employee is going to work in the accrual period. It can be used for hire and termination rules. The function is used for fields that show data in numeric form.

Overview

The threshold values relates to the months, which are derived by the annual accrual period. This is not necessarily a calendar month.

Example 1: If your accrual period is from January 1 to December 31, typical calendar months are derived: January 1 to January 31, February 1 to February 28 or 29, and so on.

Example 2: If your accrual period is from March 15 to March 14, the following months are derived: March 15 to April 14, April 15 to May 14, and so on.

Threshold start relates to the first month the employee is eligible for accruals (accruable period). This month counts if the employee's accruable period starts on or after the threshold. If you enter 0, the month is never counted.

Hire related to example 1: If you use threshold start as "10", the hire month counts if the hire date is between the 1st and the 10th of the month.

Hire related to example 2: If you use threshold start as "10", the hire month counts if the hire date is between the 15th and the 25th.

Threshold end relates to the last month of the employee's accruable period. This month counts if the employee's accruable period ends after the threshold. If you enter 0, the month is always counted.

Termination related to example 1: If you use threshold end as "10", the termination month counts if the termination date is between the 11th and the end of the month.

Termination related to example 2: If you use threshold end as "10", the termination month counts if the termination date is between the 26th and the 14th of the next month.

You can only use this function in relation to time account types where you select the Annual accrual frequency.

Manage Parameters

When you click *Manage Parameters*, make the following entries:

Code	Name	Object
accrualRuleParameters	accrualRuleParameters	Accrual Rule Parameters

Input Parameters

Here's a table showing the entries you need to make in the input parameters.

For this parameter:	Make this entry
Accruable Start Date	Please use Manage Parameters accrualRuleParameters and select Accruable Start Date
Accruable End Date	Please use Manage Parameters accrualRuleParameters and select Accruable End Date
Start Date	Please use Manage Parameters accrualRuleParameters and select Accrual Start Date
End Date	Please use Manage Parameters accrualRuleParameters and select Accrual End Date

For this parameter:	Make this entry
Threshold Start	Indicates the number of days for which the accruable start month counts.
Threshold End	Indicates the number of days for which the accruable end month counts.

Examples

Here's the data for A. N. Other's number of months taking account of threshold. The employee is hired on July 15, 2014 and terminated on November 20, 2015. The vacation account has an annual frequency period starting on January 1.

Case 1: Accrual for 2014

Accruable Start Date: July 15, 2014

Accruable End Date: December 31, 2014

Start Date: January 1, 2014

End Date: December 31, 2014

Threshold start: 15 (accruable start month counts if the start date is on or before the 15th)

Threshold end: 15 (accruable end month counts if end date is after the 15th)

Result

A. N. Other's number of months would be 6 (July until December). If the threshold start were set to a number before 15, such as 10, July would not be counted.

Case 2: Accrual for 2015

Accruable Start Date: January 1, 2015

Accruable End Date: November 20, 2015 (the employee is no longer allowed to get accrual after his or her termination date)

Start Date: January 1, 2015

End Date: December 31, 2015

Threshold start: 15 (accruable start month counts if the start date is on or before the 15th)

Threshold end: 15 (accruable end month counts if end date is after the 15th)

Result

A. N. Other's number of months would be 11 (January until November). If the threshold end were set to 20, November would not be counted.

Case 3: Accrual for 2015 with annual frequency, starting in the middle of the month.

Accruable Start Date: April 1, 2015

Accruable End Date: March 14, 2016

Start Date: March 15, 2015

End Date: March 14, 2016

Threshold start: 20 (accruable start month counts if the start date is between the 15th and 20 days later)

Threshold end: 10 (accruable end month counts if end date is after the 25th)

Result

A. N. Other's number of months would be 12 (middle of March 2015 until middle of March 2016). If the threshold end were set to 10, March 15, 2015 until April 14, 2015 would not be counted.

16.90 Get Nature of Singapore Citizenship of Dependent()

This rule function is Singapore-specific. It is designed to get the nature of citizenship of dependents for a given list of relationship types and ranks on a given date.

The rule function returns the citizenship type. It is required for the use case where certain benefits are given to citizens of Singapore only. It also considers the dependent where date of birth is not in the system. Dependents are sorted based on date of birth. In the case of twins or triplets, if twin/triplet or date of birth is not maintained, sorting is based on last name followed by first name. Dependents whose date of birth is not maintained are added to the end of the sorted list.

Input Parameters

For this parameter	Make this entry
Person ID External	Person ID external, accessed from the biographical information on the rule UI.
User ID	User ID for which youngest dependent is calculated.
Date	The date on which youngest dependent is fetched.
Relationship Types	List of relationship types for which dependents will be fetched. The values shown are the external codes of the options available in the relationship type picklist.
Rank	Rank of dependent whose age is to be calculated. The rank of a dependent is determined after all the dependents are sorted. Here, sorting is based on the date of birth. If the date of birth is the same for two or more dependents, then sorting is based on last name followed by first name.

Examples

Let's look at an example.

Employee is hired on January 1, 2010, so the dependent card has following data for the employee's dependents:

Relationship Type	Date of Birth	Last Name	First Name		Date Citizenship Acquired
Child	September 1, 2009	Klarc	Adam	By Birth	September 5, 2009
Child	September 1, 2009	Calley	Bob	By Birth	September 5, 2009
Mother		Eethen	Alen	Not a Singapore citizen	
Child		Eethen	Gable	By Descent	January 1, 2015
Spouse	January 27, 1956	Alaric	Anna	By Descent	

Sorting results in the following ranking:

Sorted Dependents

Rank	Dependent
1	Klarc, Adam
2	Calley, Bob
3	Alaric, Anna
4	Eethen, Alen
5	Eethen, Gable

To get the nature of citizenship type of the dependent for the rank 3 for the relationship type child, when this rule function is called, the nature of citizenship type returned is "By descent".

16.91 Get Next Available Manager By Position

This function returns the user ID of the next available manager in the position hierarchy.

Limitations

This function is only available if Position Management is activated.

Use Case

You can find a rule example in the Position Management Guide, under *Rule Functions in Position Management*.

16.92 Get Next Possible Date for Changes

This function returns the next possible date for which payroll-specific data changes are allowed for an employee assigned to a pay group. Payroll-specific data is, for example, *Job Information*, *Payment Information*, *Compensation Information* in the People Profile.

Configuration Requirements

You can use this function when the following prerequisites are fulfilled:

- Settings that activate the Employee Central Payroll integration are enabled in Provisioning.

  > → Remember
  >
  > As a customer, you don't have access to Provisioning. To complete tasks in Provisioning, contact your implementation partner or Account Executive. For any non-implementation tasks, contact Technical Support.

- Payroll System Configuration is defined for each Country/Region.
- OAuth outbound is configured.

Input Parameters

Input Parameter	Type	Required	User Entry
Pay Group	Text	Yes	Enter or select the *Pay Group* you want to use. This is usually the pay group external value that is entered in the *Compensation Information*.
User ID	Text	Yes	Enter or select the *User ID* you want to use to read the data from the pay group code. This is usually the User ID specified in the *Job Information*.

Behavior in Case of Errors

The rule function returns 01/01/1900 in the following cases:

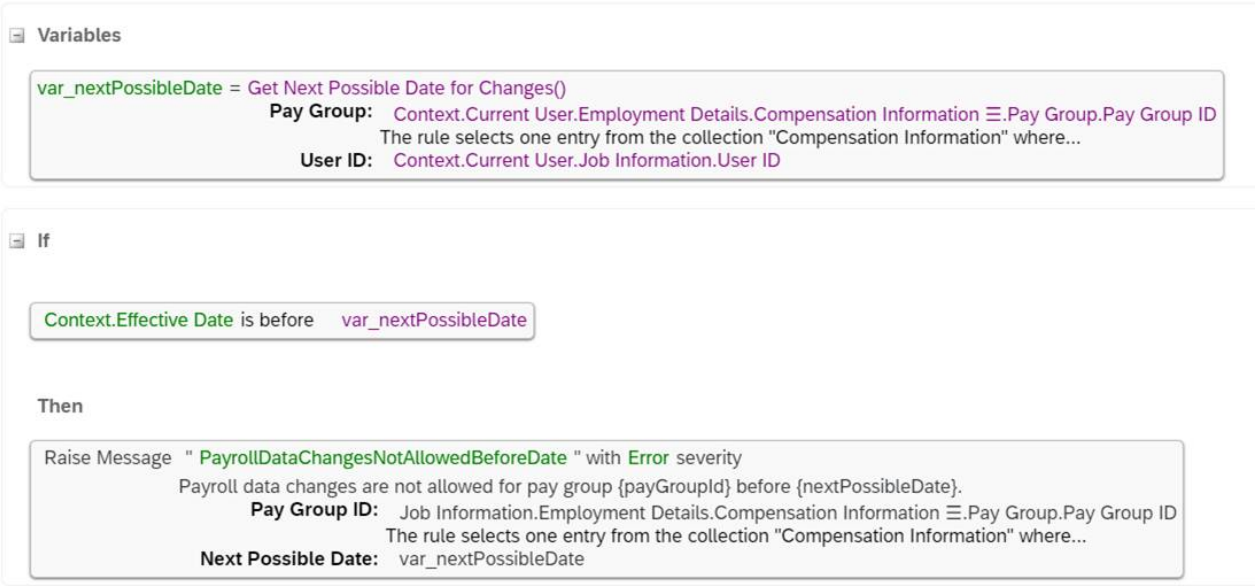
- The pay group or the user ID isn't valid.
- The Oauth configuration is missing or the payroll system URL/Client ID is missing in the *Payroll System Configuration* object.

Use Case

With this rule function, you can create a business rule to validate the changes done when employee data is included in the payroll control record information in the Employee Central Payroll system.

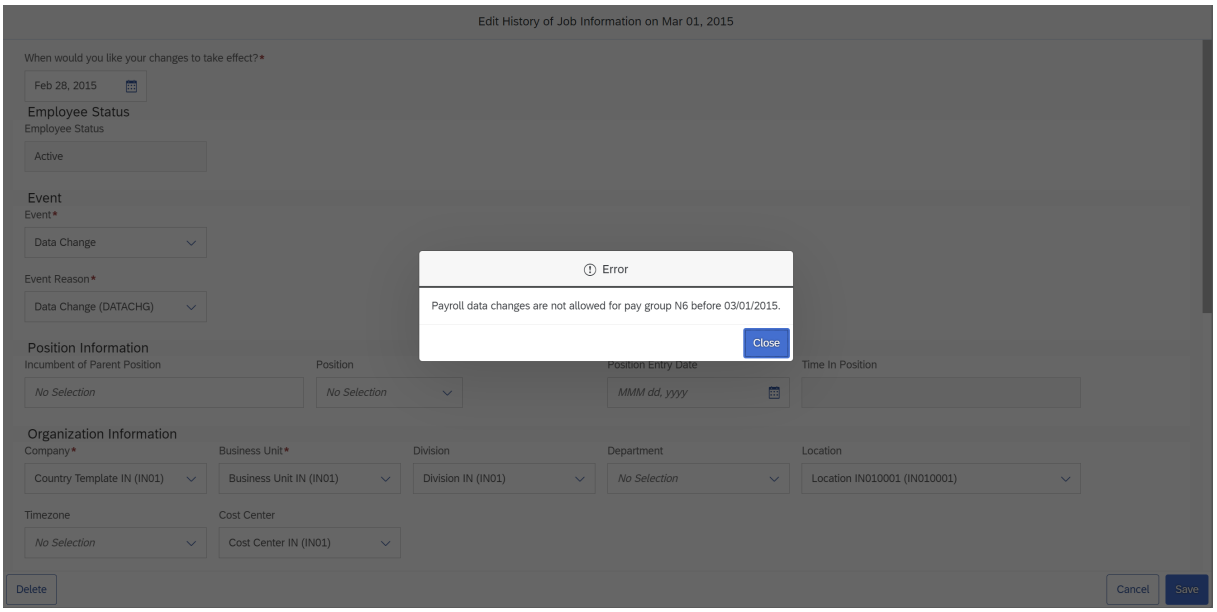
> ### Example
>
> In the Employee Central Payroll system, if the possible date for changes that apply to the payroll control record for a pay group is 01-01-2020, then no data changes should be allowed in the Employee Central cards *Job Information*, *Payment Information*, *Compensation Information* before this date. The rule function calls the payroll control record from the payroll system and returns the information so that you can raise an error message if the effective start date of the attempted change is before the payroll control record date.

The rule is associated to a *Job Information*. The *Pay Group ID* and the *User ID* are passed from the context fields. The effective start date is compared with the date resulting from the rule function. If the changes are before the date provided by the payroll system, an error message is raised.



You can also define conditions for the error message to be raised in Employee Central:

- If the return date is `01/01/1900 by default`, an error message is raised.
- Adding an `OR` condition specifying that the `var_nextPossibleDate` is equal to `01/01/1900` will raise an error message in all cases.



Related Information

Activating Employee Central Payroll Integration
Payroll System Configuration Per Country/Region
Using OAuth 2.0 to Integrate Employee Central and Employee Central Payroll

16.93 Get Next Value

This function gets the next value for a number, considering the options you've defined in the selected Sequence object.

This function is often used in combination with the *Format Number* or *Concatenate* function.

Configuration Requirements

You've created the Sequence object you want to use in this function. If you import a rule that uses this function from another system, make sure to also import the Sequence object referenced in the rule.

For more information on how to create a Sequence object, please refer to Creating a Sequence Object [page 49].

Input Parameters

Input Parameter	Type	Required	User Entry
Sequence	Value	Yes	Select a Sequence object.

Behavior in Case of Errors

In certain cases, the rule function might produce sequences with larger gaps in the numbering. Usually, there might only be a gap of 2-3 numbers, but this gap can be significantly wider if a batch runs into issues and has to be rolled back. For example, if the batch is rolled back when importing 500 new positions, you might end up with a sequence containing a gap of 500 numbers.

Use Case

You can define that the system automatically creates external codes, for example, for foundation objects, or position codes. You can find more information in the corresponding application-specific documentation.

Related Information

Generate Position Code Automatically
Example: Automatic Generation of Legacy Foundation Object Codes

16.94 Get Next Working Day()

This function determines the next working day for a user and a specific date.

Overview

The calculation is based either on work schedule only or on work schedule and holiday calendar. The function tries to find the next working day within the next 365 days starting from the input date. In the case of a cross-midnight work schedule, the day the shift starts is defined as a working day. If there is no next working day, the function returns “null”.

The following constellations lead to “null” as the return value:

- The employee has no work schedule assigned in the future.
- The employee has a work schedule assigned, but the work schedule has no working days.

Input Parameters

For this parameter	Make this entry
User ID	Select the *User* field. This is the user for which the next working day will be determined.
Date	Select a date, which is used as a basis to determine the next working day.
Consider Holidays	Choose whether holidays should be considered or not. If holidays are **not** considered, the next working date is determined based on the work schedule. If holidays **are** considered, the next working day is determined based on the work schedule and holiday calendar. If the day is a working day in the work schedule and there is no holiday, a none holiday, or a half-day holiday, the day is a working day. If the day is not a working day in the work schedule or there is a full holiday on that day, it is a non-working day.

Example 1: Non-Cross-Midnight Work Schedule and Holidays Not Considered

Let's look at an example.

A.N.Other was hired on January 1, 2015 with a duration-based work schedule based on 8 hours working time Monday - Friday. Saturday and Sunday are days off. There is a public full day holiday on May 4, 2015.

Here's the data used to determine A.N. Other's next working day:

- User ID: A.N.Other
- Date: May 1, 2015
- Consider Holidays: **No**

Result 1: May 1, 2015 is a Friday. The employee is not working on the next 2 days. so the function returns the next Monday as the next working day: May 4, 2015, because holidays are not considered.

Example 2: Non-Cross-Midnight Work Schedule and Holidays Considered

Here's another example.

A.N.Other was hired on January 1, 2015 with a duration-based work schedule based on 8 hours working time Monday till Friday and Saturday and Sunday as days off. There is a public full day holiday on May 4, 2015.

Here's the data used to determine A.N. Other's next working day:

- User ID: A.N.Other
- Date: May 1, 2015
- Consider Holidays: **Yes**

Result 2: May 1, 2015 is a Friday. The employee is not working the next 2 days and the next working day is actually a public holiday, so the function returns Tuesday as the next working day: May 5, 2015.

Example 3: Cross-Midnight Processing and Holidays Not Considered

Now we'll look at an example where cross-midnight processing is active.

A.N. Other was hired on January 1, 2015, with a cross-midnight work schedule with working time Monday-Friday 22:00 - 06:00 (next day). Saturday and Sunday are days off. There is a public full day holiday on May 4, 2015.

Here's the data used to determine A.N.Other's next working day:

- User: A.N.Other
- Date: May 1, 2015
- Consider Holidays: **No**.

Result 3: May 1, 2015, is a Friday. Saturday is not counted as a working day because the working hours 00:00 to 06:00 belong to the shift started on Friday. The employee is not working on the next 2 days (Saturday and Sunday), so the function returns Monday, May 4, 2015 as the next working day because holidays are not considered.

Example 4: Cross-Midnight Processing and Holidays Considered

Here's another example.

A.N. Other was hired on January 1, 2015, with a cross-midnight work schedule with working time Monday-Friday 22:00 - 06:00 (next day). Saturday and Sunday are days off. There is a public full day holiday on May 4, 2015.

Here's the data used to determine A.N.Other's next working day:

- User: A.N.Other
- Date: May 1, 2015
- Consider Holidays: **Yes**.

Result 4: May 1, 2015, is a Friday. Saturday is not counted as a working day because the working hours 00:00 to 06:00 belong to the shift started on Friday. The employee is not working on the next 2 days (Saturday and Sunday). The next working day would be Monday, but, as this is a public holiday, the function returns Tuesday, May 5, 2015.

16.95 Get Number of Absences for Period for Time Types()

This function calculates the number of absences of a list of time types within a given period.

An absence is also considered if it overlaps partially with the given period. For cross midnight absences, the absence is considered based on the working day to which the absence belongs.

Input Parameters

For This Field	Make This Entry
User	Select the *User* field. This is the user for which the number of absences is calculated.
Start Date	Select period start date.
End Date	Select period end date.
Time Types	Select time types. These are used to select employee times for calculation. **Note** Please only use time types with the time type category *Absence* for this rule function. Otherwise, errors might occur.

Examples

Let's look at some examples.

Example 1

User: A.N. Other

Start Date: January 1, 2017

End Date: December 31, 2017

Time Types: Vacation; Sickness

Existing Absences for this user:

- December 20, 2016 – January 1, 2017; Time Type = Sickness
- May 1, 2017 – May 5, 2017; Time Type = Special Leave
- June 1, 2017 – June 10, 2017; Time Type = Vacation

Result

The number of absences is 2.

The Sickness absence is counted because it overlaps with the period and the time type is part of the time types list.

The Vacation absence is counted because it is completely within the period and the time type is part of the time types list.

The Special Leave is not counted because it is not part of the time types list.

Example 2: Cross Midnight

User: A.N. Other

Start Date: July 1, 2020

End Date: July 31, 2020

Time Types: Vacation; Sickness

Prerequisites: User A.N. Other has a cross midnight work schedule. He is working Monday to Friday from 10pm to 6am. Saturday and Sunday are non working days.

Existing Absences for this user:

- June 15, 2020 - June 30, 2020 (in detail June 15, 2020 10pm to July 1, 2020 6am); Time Type = Sickness
- July 3, 2020 - July 3, 2020 (in detail July 3, 2020 10pm to July 4, 2020 6am); Time Type = Special Leave
- July 30, 2020 - August 30, 2020 (in detail July 30, 2020 10pm to August 31, 2020 6am); Time Type = Vacation

Result

The number of absences is 1.

The Sickness absence is not counted because it does not overlap with the period. Although the absence ends on July 1, 2020 6am, this belongs to the shift starting on June 30, 2020 and is not counted.

The Vacation absence is counted because it overlaps with the period and the time type is part of the time types list.

The Special Leave is not counted because it is not part of the time types list.

16.96 Get Number of Allowances in Period()

This rules function enables you to check how many allowances a user has within a given period of time.

To get the number of allowances, enter both the user and period of time. If you want, you can search for specific allowance types. The rule function will then check all active time sheets for allowances saved on the database, including time sheets that have not been approved yet.

Input Parameters

For this parameter	Make this entry
User (Required)	The user whose allowances should be checked.

For this parameter	Make this entry
Start Date (Required)	The start of the time period for which you want to check allowances.
End Date (Required)	The end of the time period for which you want to check allowances.
Amendment Scenario (Optional)	The allowances that you want to consider in case of an amendment scenario. If you don't enter a parameter value, all allowances are considered, except for those from inactive, rejected or cancelled time sheets. • *Only consider the amended time sheet.*: This value considers allowances from time sheets with the statuses *Pending* or *Pending Approval* if there is an amendment. This value also considers approvals if there is no amendment. • *Only consider the approved time sheet.*: This value considers only allowances from approved time sheets. • *Consider only allowances that have changed.*: This value lets you find amendments for monthly time sheets that are stored in the *Allowance Recording* object.
Allowance Types (Optional)	The type(s) of allowances you want to check for. If you leave this blank, the rule function will automatically search for all allowance types.

Use Case

You want to make sure that employees can't take a full day off if meal allowances are paid. To achieve this, you create a rule that uses the Get Number of Allowances in Period function. This rule determines how many allowances of a specific type are paid out to an employee within a given time. In this particular case, you want to find out if one or more meal allowances were paid during a requested absence (one or more days). If one or more meal allowances are detected during a requested absence, the absence request is rejected.

16.97 Get Number Of Calendar Days()

This rule function calculates the number of calendar days between a start date and end date, including the start date and the end date.

Parameters

For this parameter	Enter
Start Date	The start date of the period you want to calculate.
End Date	The end date of the period you want to calculate.

> **Note**
>
> Ensure that you enter an actual date for both parameters (you cannot leave one blank or enter "0"), and that the start date falls before the end date. Otherwise, the rule containing this rule function will not run correctly.

Example #1:

- Start date: April 1, 2019
- End date: April 1, 2019
- Result: 1 day

Example #2:

- Start date: April 1, 2019
- End date: April 10, 2019
- Result: 10 days

16.98 Get Number Of Child Positions

This function returns the number of child positions.

Limitations

This function is only available if Position Management is activated.

Use Case

You can find a rule example in the Position Management Guide, under *Rule Functions in Position Management*.

16.99 Get Number Of Days For Year Of Date()

This functions determines the number of days, 365 ot 366, in the year of the date under consideration. It has one parameter, where you enter the date in question.

16.100 Get Number Of Eligible Days()

This function calculates the number of eligible days for getting accruals. It is used for fields that show data in numeric form. You can define upfront in Time Off which employee status is included in eligibility.

Manage Parameters

When you click *Manage Parameters*, make the following entries:

Code	Name	Object
accrualRuleParameters	accrualRuleParameters	Accrual Rule Parameters

Input Parameters

For this parameter:	Make this entry
User ID	Select the *User* field. This is the user for which the accrual should be created.
Time Account Type	Select the time account type the rule is valid for.
Start Date	Please use Manage Parameters accrualRuleParameters and select Start Date.
End Date	Please use Manage Parameters accrualRuleParameters and select End Date.

For this parameter:	Make this entry
Accruable Start Date	Please use Manage Parameters accrualRuleParameters and select Accruable Start Date.
Accruable End Date	Please use Manage Parameters accrualRuleParameters and select Accruable End Date.

Example

Here's the data for A.N.Other's average eligible days calculation.

User ID: A.N. Other

Time Account Type: = Vacation

Accrual Start Date = January 1, 2017

Accrual End Date = December 31, 2017

Accruable Start Date = July 1, 2017

Accruable End Date = December 31, 2017

Result

Case 1: Employee is active in accruable period = 184 days would be the result

Case 2: Employee is not eligible (take a look at the information on the Time Off Configuration in the appendix to the Implementing Employee Central Time Off guide for more on this) in accruable period for August (31 days): 184 – 31 = 153 days would be the result.

.

16.101 Get Number Of Holidays For Period()

This function calculates the number of holidays for a user and a given time period.

Overview

The result depends on the holiday category and the holiday planned working time of the holidays:

- For *No Planned Working Time*, 1 day is added to the result.
- For Reduced Planned Working Tiime, the day is added to the result as follows:
  - When Day Model Variant Identifier is not entered, 0.5 days are added to the result.

- When Day Model Variant Identifier is entered, a value between 0 to 1 days is added to the result based on the Day Model Variant associated with this identifer.
- For *Scheduled Working Time*, 0 days are added to the result.

You can also define that only holidays on specific days are taken into account, for example holidays that fall on a weekend (Saturday and Sunday).

If an error occurs, the function returns *Null*.

Input Parameters

For this field	Make this entry
User	Select the *User* field. This is the user for which the number of holidays should be calculated.
Start Date	Period start date
End Date	Period end date
Weekdays	If you enter any days here, only holidays for these days are taken into account. If you want the rule to consider **all** days within the designated time period, enter Null here.

Example

Let's look at an example.

In the selection period, the following holidays have been defined for the employee's holiday calendar:

- *Full* holiday on October 5, 2015 (Monday)
- *Half* holiday on December 24, 2015 (Thursday)
- *None* holiday on December 19, 2015 (Saturday).

Result

If you call the rule function without weekdays, all holidays are considered. The result is 1.5 days (= 1 + 0.5 + 0).

If you call the rule function for weekdays Saturday and Sunday, the result is 0. Only the holiday on December 19, 2015, is considered as it´s a Saturday, but, as it´s a None holiday, the result is 0.

16.102 Get Number Of Months From Hire Date()

This function calculates the number of months the employee is going to work.

Overview

You can exclude the hire month from the calculation if the hire date is not on the first day of the month. The function is used for fields that show data in numeric form.

The accrual calculation is based on the number of months from the hire date, including or excluding the hire month.

> ⓘ Note
>
> - Use this rule function only in the hire rule scenario. Do **not** use it in any other rule scenario.
> - This rule function can only be used for the "Annually" accrual frequency.
> - If the *Account Creation Start Date* setting in the time account type is set to "Employee Hire Date", the rule function always returns 12 months.

Manage Parameters

When you click *Manage Parameters*, make the following entries:

Code	Name	Object
accrualRuleParameters	accrualRuleParameters	Accrual Rule Parameters

Input Parameters

Here's a table showing the entries you need to make in the input parameters.

For this parameter:	Make this entry
Start Date	Please use Manage Parameters accrualRuleParameters and select Start Date.
End Date	Please use Manage Parameters accrualRuleParameters and select End Date.
Recruit Date	Select the User Field – Employment Details – Recruit Date.

For this parameter:	Make this entry
Incl. Recruitment Month	Decide whether the recruitment month should be counted if the hire date is not the first day of the month. If the hire date *is* the first day of the month, it will be counted anyway.

Examples

Let's take a look at some examples:

Example 1

Start Date: January 1, 2016

End Date: December 31, 2016

Recruitment Date: June 1, 2016

Incl.Recruitment Month: true

Result 1

- Every month starts on the first.
- Incl. actual month has no effect because the employee is hired at the beginning of the month.
- The calculated number of months will be 7 (month 6 to month 12 inclusive).

Example 2: Termination within accrual period

Start Date: February 5, 2015

End Date: February 4, 2016

Recruitment Date: June 10, 2015

Incl.Actual Month: true

Result 2

- Every month starts on the 5th.
- Hire month should be counted.
- The calculated number of months will be 8.

16.103 Get Number Of Months From Start Date Until End Date

This function determines the number of months between two given dates. The end date is excluded from this calculation.

Examples

Example 1

January 1, 2014, until December 31, 2014

Result: 11

Example 2

January 1, 2014, until January 1, 2015

Result: 12

Example 3

January 15, 2014, until February 14, 2014

Result: 0

Example 4

January 15, 2014, until February 15, 2014

Result: 1

16.104 Get Number Of Postings For Posting Types in Period()

The rule function returns the number of postings in a given period.

You can decide which postings you are interested in based on the list of posting types. Examples include period-end processing postings and interim account update postings.

Input Parameters

For this parameter:	Make this entry:
Start Date	Enter the date the relevant period should start with.

For this parameter:	Make this entry:
End Date	Enter the date the relevant period should end with.
Time Account	Enter the time account parameter from the rule.
Posting Types	Select a list of relevant posting types.

Examples

Let's look at some examples.

We have a time account with these postings:

January 1, 2017	Accrual	+5 days
February 1, 2017	Accrual	+5 days
February 15, 2017	Manual Adjustment	+10 days
March 1, 2017	Accrual	+5 days
March 1, 2017	Interim Account Update	-1 day
April 1, 2017	Accrual	+5 days

Example 1: Execute the rule function with these parameters:

- Start Date: January 1, 2017
- End Date: December 31, 2017
- Time Account: Time account with postings as listed in the table.
- Posting Types: Accrual, Manual Adjustment

The result would be 5 postings within this period - 4 accrual postings and 1 manual adjustment posting.

Example 2: Execute the rule function with these parameters:

- Start Date: April 1, 2017
- End Date: December 31, 2017
- Time Account: Time account with postings as listed in the table.
- Posting Types: Interim Account Update

The result would be no (0) postings within this period. The interim account update posting is posted outside the given period, so it is not considered.

> **Note**
>
> If you specify the *Posting Type* as *Accrual* or *Recalculation*, you don't get any results if this rule function is used in the accrual rule. That's because we remove all existing accruals to be recalculated before calling accrual rules.

The reason for this is that the resulting accrual amount should be independent of whether the accrual is initially created or whether it's recalculated. For example, if you have a cap accrual scenario or you are using the balance rule function, that function shouldn't consider the amount of the accrual you are about to calculate or recalculate.

That's why, for accrual calculation, not getting any results if this rule function is used in the accrual rule is expected behavior. If you have an accrual rule scenario and you want to get the number of persisted accrual postings, you can't use the Get Number Of Postings For Posting Types in Period() rule function.

Use Cases

Especially for period-end processing rules or interim account update rules, it might be useful to know whether there is already such a posting on the corresponding time account. By using this rule function in combination with the rule function Get Balance For Posting Types In Period() [page 209], you can consider the existing posting during the creation of the new one:

Example with interim account update: The current interim account should ensure that 20 hours are deducted from each time account. The first time you run this rule, the corresponding posting will be created. If you run this interim rule again for the same time account, again 20 hours are deducted. However, you can use this rule function to determine whether there is already an interim account update posting for this date on the time account and, if there is, skip this account based on this information.

16.105 Get Number of Valuated Hours for Time Sheet()

This rule function collects time valuation results from the time sheet base object that satisfy the optional parameters

Use this rule function in conjunction with the Time Off rule function *Get Balance For Types()* and the Time Sheet rule function *Get Working Time Account on Key Date()* to get the working time account as of the key date.

> ⓘ Note
>
> Though you don't have to enter anything for the optional parameters detailed in the table of input parameters, if you don't, the result will be zero

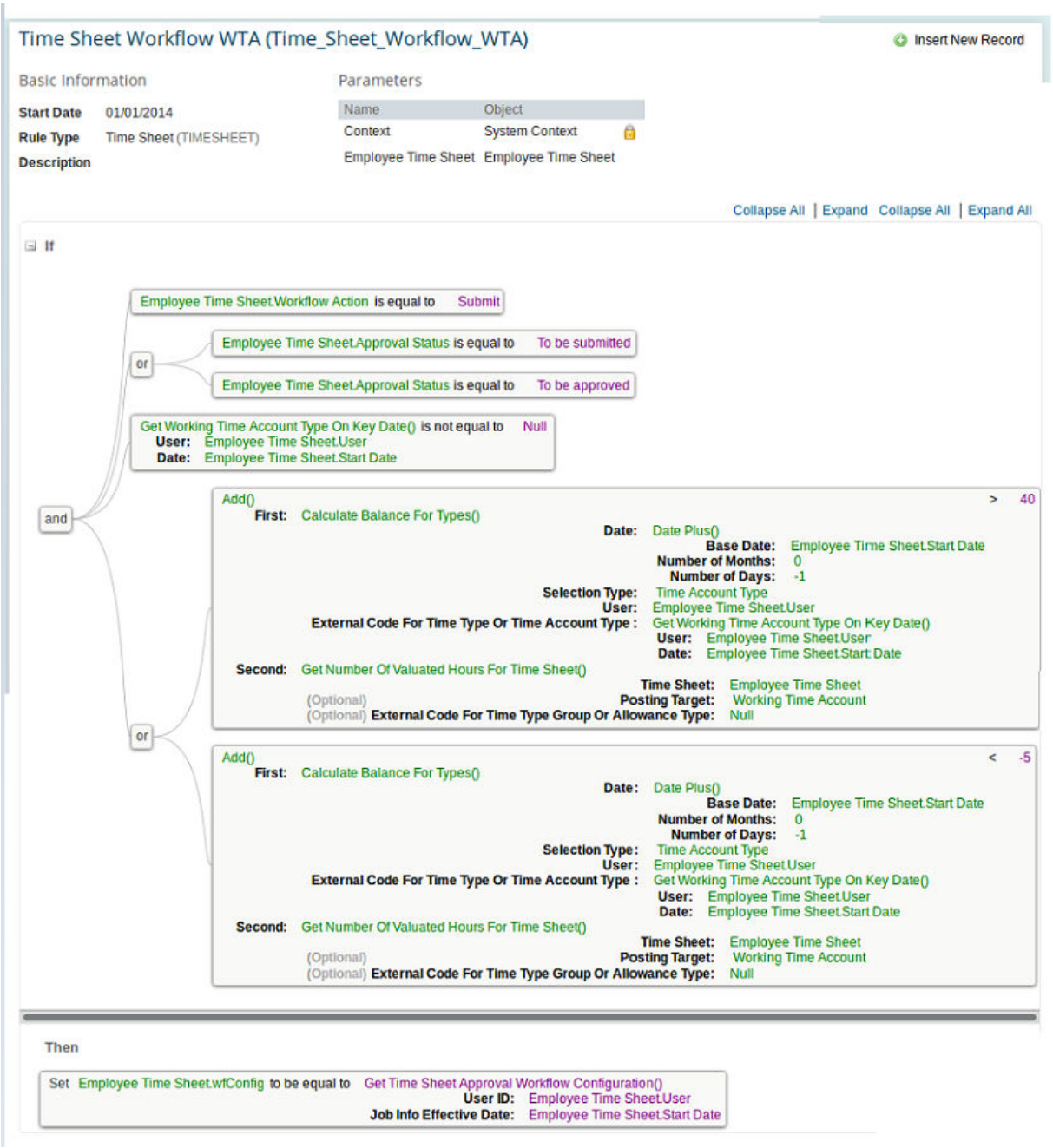
Input Parameters

Here's a table showing the entries you need to make in the input parameters:

For this parameter:	Make this entry
Time Sheet (Required)	The time sheet that forms the base object.
TimeValuationResult.postingTarget (Optional)	Enter **`EC_Payroll`**, **`Working_Time`**, or **`Working_Time_Account`**.
An arbitrary number of Strings interpreted as externalCode of a Time Type Group or an Allowance_Type. (Optional)	

Example

If you choose working time account as the posting target, the result is the sum of all time valuation results of this time sheet that will be posted to the working time account. In the example, this number is added to the working time account balance as of the date before the start date of the time sheet, which results in what the working time account balance would be if this time sheet was approved. The rule checks that this working time account balance does not exceed a limit of 40 hours, or falls below -5 hours. In both cases, the time sheet needs manager approval.



16.106 Get Number Of Working Days Or Hours For Period()

This rule function returns the number of days or hours for an employee in a given period.

To calculate this, the work schedule assigned to the employees job information is considered as well as temporary changes. You can choose whether holidays should be considered or not.

In the case of a cross-midnight work schedule, a working day is determined as the day on which the working time starts. If Hours is used as a parameter, all hours belonging to this working day are added together. Take a look at the cross-midnight examples,

Input Parameters

For this parameter	Make this entry
User	Enter the user ID.
Start Date	Enter the start date
End Date	Enter the end date.
Time Unit	Select whether the rule function should return the quantity in hours or days.
Consider Holidays	Select whether holidays are considered or not.

Examples with Non-Cross-Midnight Work Schedules

Prerequisites

- A.N.Other has a work schedule based on duration, with 8 hours working time configured Monday to Friday. Saturday and Sunday are non-working days.
- A.N.Other's holiday calendar includes a full day public holiday on Monday, May 4, 2020, and a half day holiday on Wednesday, June 3, 2020.

Example 1: Full day holiday is not considered.

- User: A.N.Other
- Start Date: Monday, May 4, 2020
- End Date: Sunday, May 10, 2020
- Time Unit: Hours
- Consider Holidays: No

Result 1: The rule function returns 40 hours. May 4 counts because holidays are not considered. Saturday and Sunday are not working days at all.

Example 2: Full day holiday is considered.

- User: A.N.Other
- Start Date: Monday, May 4, 2020
- End Date: Sunday, May 10, 2020
- Time Unit: Hours
- Consider Holidays: Yes

Result 2: The rule function returns 32 hours. May 4 doesn't count because it is a public holiday. Saturday and Sunday are not working days at all.

Example 3: Full day holiday is not considered.

- User: A.N.Other
- Start Date: Monday, May 4, 2020
- End Date: Sunday, May 10, 2020
- Time Unit: Days
- Consider Holidays: No

Result 3: The rule function returns 5 days. May 4 counts as a full day because holidays are not considered. Saturday and Sunday are not working days at all.

Example 4: Full day holiday is not considered.

- User: A.N.Other
- Start Date: Monday, May 4, 2020
- End Date: Sunday, May 10, 2020
- Time Unit: Days
- Consider Holidays: Yes

Result 4: The rule function returns 4 days. May 4 doesn't count because holidays are considered. Saturday and Sunday are not working days at all.

Example 5: Half day holiday is not considered.

- User: A.N.Other
- Start Date: Monday, June 1, 2020
- End Date: Sunday, June 7, 2020
- Time Unit: Hours
- Consider Holidays: No

Result 5: The rule function returns 40 hours. June 3 counts because holidays are not considered. Saturday and Sunday are not working days at all.

Example 6: Half day holiday is considered.

- User: A.N.Other
- Start Date: Monday, June 1, 2020
- End Date: Sunday, June 7, 2020
- Time Unit: Hours
- Consider Holidays: Yes

Result 6: The rule function returns 36 hours. June 3 counts as a half day because it is half day holiday and 4 hours are not considered. Saturday and Sunday are not working days at all.

Example 7: Half day holiday is not considered.

- User: A.N.Other
- Start Date: Monday, June 1, 2020
- End Date: Sunday, June 7, 2020
- Time Unit: Days

- Consider Holidays: No

Result 7: The rule function returns 5 days. June 3 counts because holidays are not considered. Saturday and Sunday are not working days at all.

Example 8: Full day holiday is not considered.

- User: A.N.Other
- Start Date: Monday, June 1, 2020
- End Date: Sunday, June 7, 2020
- Time Unit: Days
- Consider Holidays: Yes

Result 8: The rule function returns 4.5 days. June 3 counts as a half day because holidays are considered and 0.5 days are deducted. Saturday and Sunday are not working days at all.

Examples with Cross-Midnight Work Schedules

Prerequisites

- A.N.Other has a work schedule based on clock times, with 8 hours working time from 22:00 to 06:00 (next day) configured Monday to Friday. Saturday and Sunday are non-working days.
- A.N.Other's holiday calendar includes a full day public holiday on Monday, May 4, 2020. Half day holidays are not supported for employees with work schedules based on clock times.

Example 1: Full day holiday is not considered.

- User: A.N.Other
- Start Date: Wednesday, May 6, 2020
- End Date: Saturday, May 9, 2020
- Time Unit: Days
- Consider Holidays: No

Result 1: The rule function returns 3 working days (Wednesday, Thursday, Friday). Although the employee is physically working on Saturday (00:00 to 06:00), Saturday is not counted. A day is defined as a working day if the start time of the shift is on that day. Holidays are not considered, so May 6 is counted as well.

Example 2: Full day holiday is considered.

- User: A.N.Other
- Start Date: Wednesday, May 6, 2020
- End Date: Saturday, May 9, 2020
- Time Unit: Days
- Consider Holidays: Yes

Result 2: The rule function returns 2 working days (Thursday, Friday). Although the employee is physically working on Saturday (00:00 to 06:00), Saturday is not counted. A day is defined as a working day if the start time of the shift is on that day. Holidays are considered, so May 6 is not counted.

Example 3: Full day holiday is not considered.

- User: A.N.Other
- Start Date: Wednesday, May 6, 2020
- End Date: Friday, May 8, 2020
- Time Unit: Hours
- Consider Holidays: No

Result 3: The rule function returns 24 working hours (Wednesday 22:00-06:00, Thursday 22:00-06:00, Friday 22:00-06:00). Although the employee is physically working on Saturday (00:00 to 06:00), those hours are considered as being on Friday because the shift starts on that day. Holidays are not considered, so May 6 is counted as well.

Example 4: Full day holiday is considered.

- User: A.N.Other
- Start Date: Wednesday, May 6, 2020
- End Date: Saturday, May 9, 2020
- Time Unit: Days
- Consider Holidays: Yes

Result 4: The rule function returns 16 working hours (Thursday 22:00-06:00, Friday 22:00-06:00). Although the employee is physically working on Saturday (00:00 to 06:00), those hours are considered as being on Friday because the shift starts on that day. Holidays are considered, so May 6 is not counted.

16.107 Get Pay Calendar Begin or End or Check Date

This function retrieves the start, end, or check date of the pay period that contains the given input date.

Input Parameters

For this parameter...	Make this entry...
Pay Group	Enter the relevant pay group.
Date	Enter a date (most commonly effective start date).
Validation Type	Start, end, or check date.

16.108 Get Pay Calendar Info

This function returns the payroll calendar information that can be used to get the payroll period begin date, end date and check date.

Payroll calendar info includes the begin date, end date, check date, pay group.

Configuration Requirements

You can use this function when the following prerequisites are fulfilled:

- The *Pay Calendar* MDF object is enabled.
- Pay period details are available in the *Pay Calendar* MDF object.

Input Parameters

Input Parameter	Type	Required	User Entry
Pay Group	Value	Yes	Enter or select the *Pay Group* you want to use. This is usually the pay group external value that is entered in the *Compensation Information*.
Effective Date	Date	Yes	Enter or select the *Effective Date* you want to use to read the date for which the pay range should be read. **Note** If this field is empty or null, the system determines the pay calendar info from the current date.

Behavior in Case of Errors

The rule function returns an empty record for the payroll calendar information with the parameter *Is Changed* as false and with error messages for the following reasons:

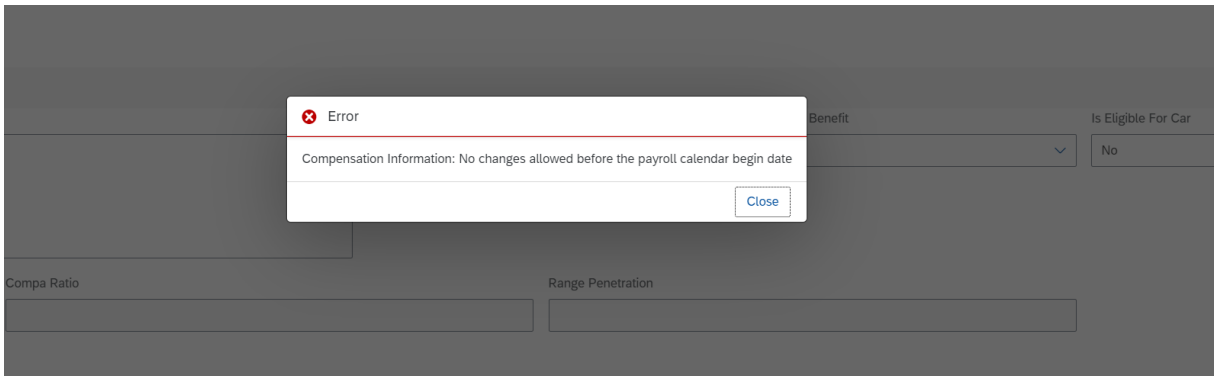
- The pay group is null.
- The pay group isn't valid.
- The pay calendar isn't found for the pay group.
- The pay period isn't found for the pay group.
- The pay period is before or after the effective date.
- The object definition Pay Calendar doesn't exist or is inactive.

Use Case

With this rule function, you can create a business rule to validate if a pay group fits to a pay period with a specific date (effective date).

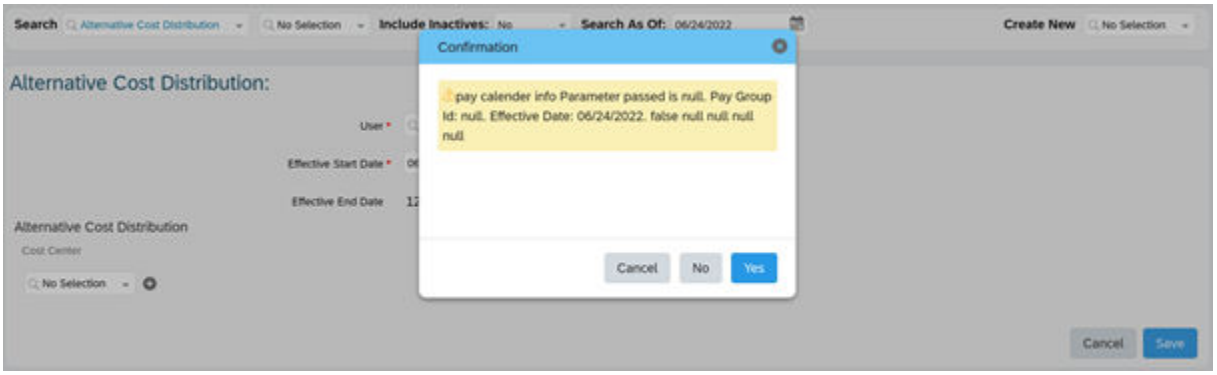
1. In this example, you can validate if the effective date is after the begin date of the pay calendar. If the effective date is before the begin date of the pay calendar, the rule raises the following error message:

```
Variables
var_get_pay_calendar_info = Get Pay Calendar Info()
                                Pay Group:  Compensation Information.Pay Group
                                Effective Date:  Compensation Information.Event Date

If
Context.Effective Date is before  var_get_pay_calendar_info.Begin Date

Then
Raise Message  " PayCalendarBeginDateMsg " with Error severity
                No changes allowed before the payroll calendar begin date
```



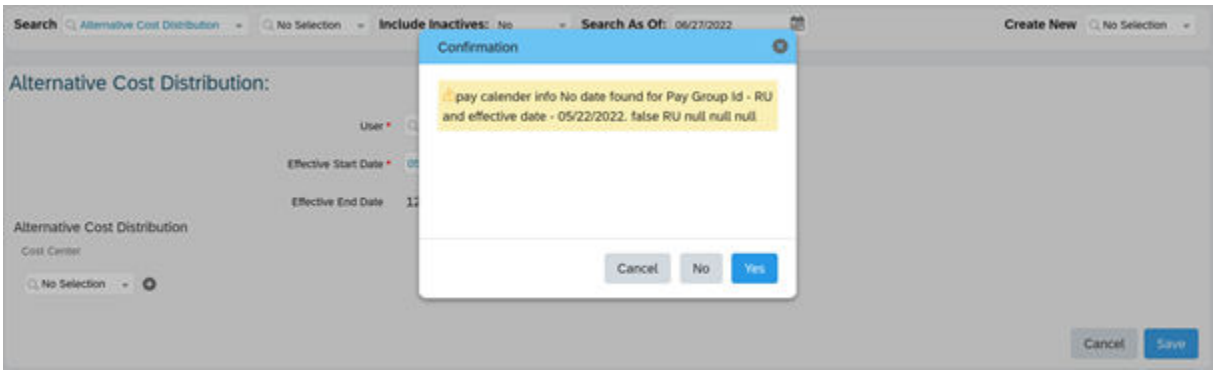
2. In this example, you can validate if the effective date is in range of the pay period. The rule function then returns the boolean flag *Is Changed*. You can use it in the rule to raise a message with the reason included in the result of rule.

```
Variables
var_pay_calendar_info = Get Pay Calendar Info()
                  Pay Group Id:  Alternative Cost Distribution.User.Employment Details.Compensation Information ☰.Pay Group
                                 The rule selects one entry from the collection "Compensation Information" where...
                  Effective Date:  Context.Effective Date

If
var_pay_calendar_info.Is Changed is equal to  No

Then
Raise Message  " pay calender info {reason} {changeInfo} {payGroup} {beginDate} {endDate} {checkDate} " with Warning severity
               pay calender info {reason} {changeInfo} {payGroup} {beginDate} {endDate} {checkDate}
                  reason:  var_pay_calendar_info.Reason
                  changeInfo:  var_pay_calendar_info.Is Changed
                  payGroup:  var_pay_calendar_info.Pay Group Id
                  beginDate:  var_pay_calendar_info.Begin Date
                  endDate:  var_pay_calendar_info.End Date
                  checkDate:  var_pay_calendar_info.Check Date
```



3. In this example, you can validate if an employee has no pay group. The rule function then returns the boolean flag *Is Changed*. You can use it in the rule to raise a message with the reason included in the result of rule.

```
Variables
var_pay_calendar_info = Get Pay Calendar Info()
                  Pay Group Id:  Alternative Cost Distribution.User.Employment Details.Compensation Information ☰.Pay Group
                                 The rule selects one entry from the collection "Compensation Information" where...
                  Effective Date:  Context.Effective Date

If
var_pay_calendar_info.Is Changed is equal to  No

Then
Raise Message  " pay calender info {reason} {changeInfo} {payGroup} {beginDate} {endDate} {checkDate} " with Warning severity
               pay calender info {reason} {changeInfo} {payGroup} {beginDate} {endDate} {checkDate}
                  reason:  var_pay_calendar_info.Reason
                  changeInfo:  var_pay_calendar_info.Is Changed
                  payGroup:  var_pay_calendar_info.Pay Group Id
                  beginDate:  var_pay_calendar_info.Begin Date
                  endDate:  var_pay_calendar_info.End Date
                  checkDate:  var_pay_calendar_info.Check Date
```



16.109 Get Pay Range Attributes

Use this rule function to determine the attributes of a pay range such as Minimum Pay, Maximum Pay, Mid Point, Currency, and Frequency.

You need to enter a pay range, the pay range field, and a date.

Configuration Requirements

Use only if pay ranges are set up. You can find full details of how to set up and use pay ranges in the Showing Pay Range on Position section of the Implementing Employee Central Position Management guide.

Input Parameters

Input Parameter	Type	Required	User Entry
Pay Range	Value	Yes	Select the pay range whose attributes you want to determine.
Pay Range Field	Value	Yes	Select the attribute you're interested in. Options are: • Minimum Pay • Maximum Pay • Mid Point • Currency • Frequency
As Of Date	Date	Yes	Select the date as of which you want to determine the specified attribute.

Behavior in Case of Errors

If "null" values are provided for the *Get Pay Range Attributes* function, it returns empty.

> **Note**
>
> Please note that "null" doesn't mean the number zero in the context of business rules. Instead, it represents a missing value. A field with a null value is a field with no value.

Use Case

Use this rule function to determine the attributes of a pay range as of a particular date. You can find a rule example in the Employee Central Position Management guide.

Related Information

Showing Pay Range on Position

16.110 Get Pay Range By Position

This function gets the associated pay range from the position.

Limitations

Use only if pay ranges are set up.

Calculating the pay range for a position depends on the Job Information configuration. This means that all fields for Job Information need to be configured with the same data types as for the position. If the pay range depends on a custom MDF object, for example, a field with this data type must exist in both Position and Job Information.

Input Parameters

For this parameter...	Of type...	Which is...	Make this entry:
Position	Generic Object	required	a position object

For this parameter...	Of type...	Which is...	Make this entry:
Effective Date	Date	optional	the effective date for which the pay range should be read. If this field is null, the effective start date of the position will be used.

Use Case

You can create a rule that displays the associated pay range when the relevant fields of the position are changed.

The pay range is only calculated in *Manage Data* or *Manage Positions* in the position organization chart. It is not calculated on other pages such as the workflow approval page.

Related Information

Showing Pay Range on Position

16.111 Get Payroll Area Control Record

This function returns the payroll control record information that can be used to validate if the payroll-specific data changes are to be allowed. Payroll-specific data is, for example, *Job Information*, *Payment Information*, *Compensation Information* in the People Profile.

Payroll control record includes the status, earliest retroactive date, next possible date for changes, current period start date, next period start date as well as the reason for not allowing changes.

Configuration Requirements

You can use this function when the following prerequisites are fulfilled:

- Settings that activate the Employee Central Payroll integration are enabled in Provisioning.

  > → Remember
  >
  > As a customer, you don't have access to Provisioning. To complete tasks in Provisioning, contact your implementation partner or Account Executive. For any non-implementation tasks, contact Technical Support.

- Payroll System Configuration is defined for each Country/Region.
- OAuth outbound is configured.

> **Note**
>
> For Employee Central Payroll, Support Package SP99 must be installed in your system. Please note that you can't use the feature as a whole until the support package mentioned has been released.

Input Parameters

Input Parameter	Type	Required	User Entry
Pay Group	Text	Yes	Enter or select the *Pay Group* you want to use. This is usually the pay group external value that is entered in the *Compensation Information*.
User ID	Text	Yes	Enter or select the *User ID* you want to use to read the data from the pay group code. This is usually the User ID specified in the *Job Information*.
Start Date	Date	Yes	Enter or select the *Start Date* the data is to be updated on. This is usually the *Effective Start Date* specified in the *Job Information*.

Behavior in Case of Errors

The rule function returns empty payroll control information record with `is allowed` as false and with error messages for following reasons:

- The pay group, or the user ID, or start date is null.
- The pay group isn't valid.
- The user's job information isn't valid.
- The start date is before the earliest retroactive date.
- The OAuth configuration is missing or the payroll system URL/Client ID is missing in the *Payroll System Configuration* object.
- The payroll control record is, for example, in *RELEASED_FOR_PAYROLL* status. Check the table below for errors related to the status of the payroll control record.

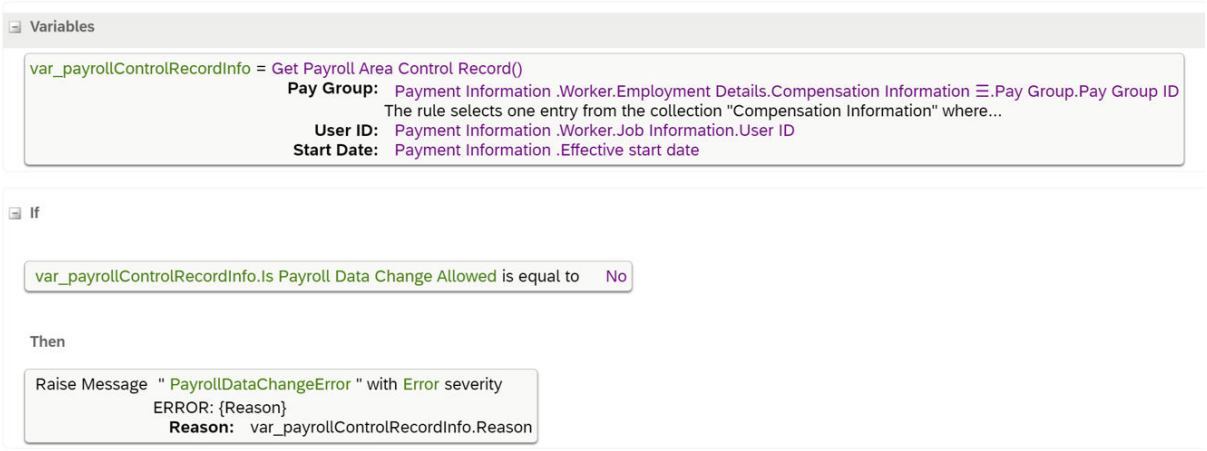
The following table lists the behavior of this rule for different status values of the Payroll Control Record. In particular it shows when data changes are allowed, and the reasons when data changes are not allowed, and an error is returned

Payroll Area Status	Data Change Allowed?	Reason
New [0]	No	The payroll area status is new. It can't be used until the payroll status is set up correctly.
Released for Payroll [1]	No	The payroll area is locked for master data maintenance.
Payroll Correction [2]	Yes	
Exit Payroll [3]	Yes	
Check Payroll Results [4]	Yes	
Deleted [9]	No	The payroll are status is deleted. You can't use it any longer.

Use Case

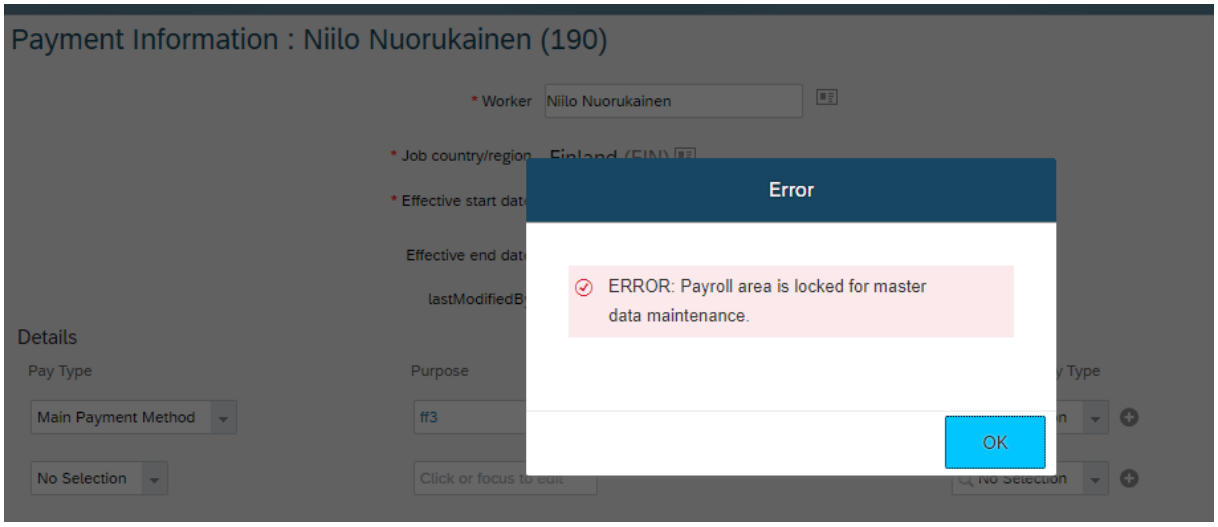
With this rule function, you can create a business rule to validate the changes done when employee data is included in the payroll control record in Employee Central Payroll.

1. In the Employee Central Payroll system, if the payroll control record status is *RELEASED_FOR_PAYROLL*, you aren't allowed to change data from Employee Central cards *Job Information*, *Payment Information*, *Compensation Information*.



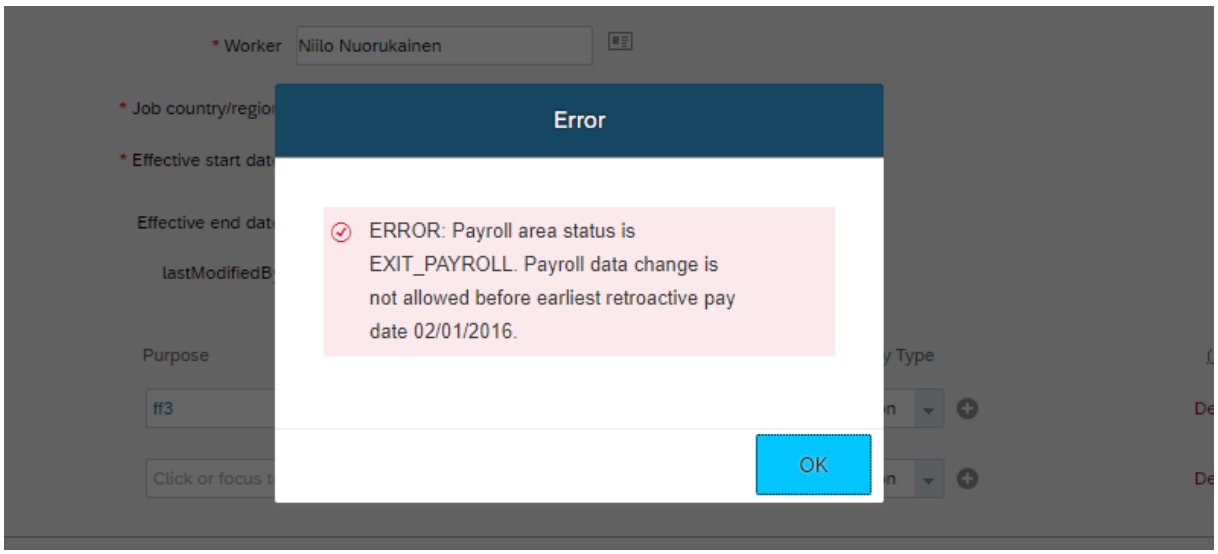
   The rule is associated to *Payment Information*. The *Pay Group ID*, the *User ID*, and the *Start Date* are passed from the payment information fields. The result of the rule function is saved to a variable. You can add a

condition to check the payroll area status. If the status is *Release for Payroll*, an error is raised.



2. In this example, you aren't allowed to change data if the changes being made have effective date before the earliest retroactive pay date of payroll control record in the Employee Central Payroll system. This rule function returns the payroll control record information from the Employee Central Payroll system. You can then use this information in a business rule to raise error message for all such use cases.

```
Variables
var_payrollControlRecord = Get Payroll Area Control Record()
                    Pay Group:  Context.Current User.Employment Details.Compensation Information ☰.Pay Group.Pay Group ID
                                The rule selects one entry from the collection "Compensation Information" where...
                      User ID:  Context.Current User.Job Information.User ID
                   Start Date:  Context.Effective Date

If
Context.Effective Date is before  var_payrollControlRecord.Earliest Retroactive Pay Date

Then
Raise Message  " PayrollDataChangeError " with Error severity
                    ERROR: {reason}
                      Reason:  var_payrollControlRecord.Reason
```

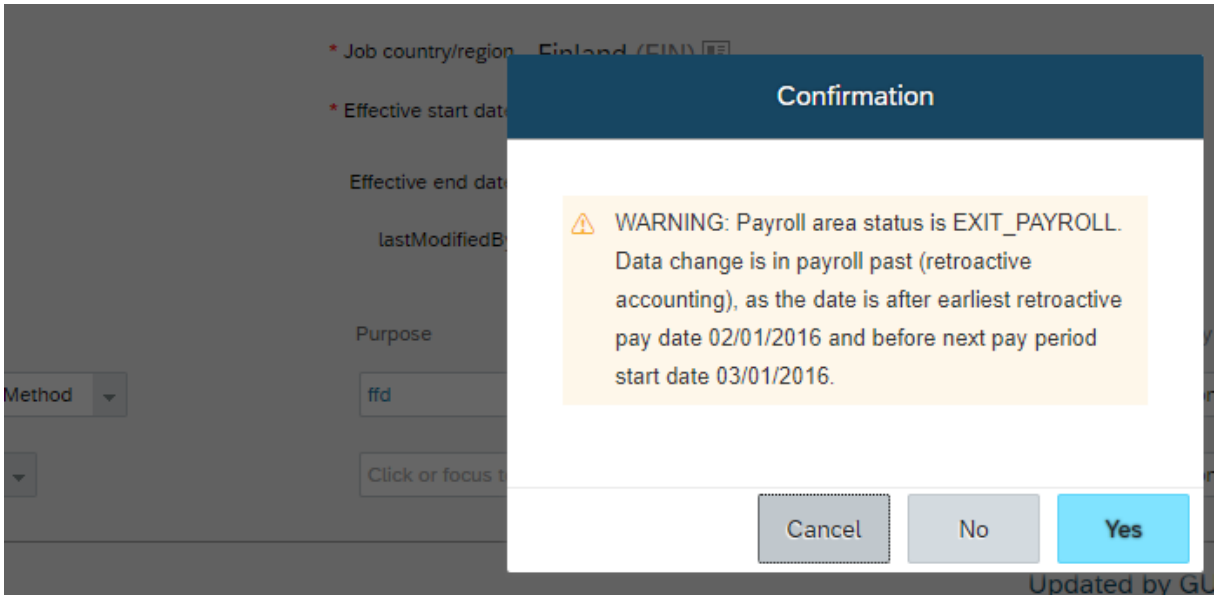


3. In this example, you can validate the effective date with the earliest retroactive date and the status. The rule function then returns a Boolean flag *Is Payroll Data Change Allowed* which can be used in the rule to raise a message with the reason that is included in the result of the rule.

```
Variables
var_payrollControlRecord = Get Payroll Area Control Record()
                    Pay Group:  Context.Current User.Employment Details.Compensation Information ☰.Pay Group.Pay Group ID
                                The rule selects one entry from the collection "Compensation Information" where...
                      User ID:  Context.Current User.Job Information.User ID
                   Start Date:  Context.Effective Date

If
var_payrollControlRecord.Is Payroll Data Change Allowed is equal to  No

Then
Raise Message  " PayrollDataChangeError " with Error severity
                    ERROR: {reason}
                      Reason:  var_payrollControlRecord.Reason
```

4. This is an example of conditions set up for the warning message to be raised in Employee Central. If the reason isn't null, a warning message is raised.

```
Variables
var_payrollControlRecordInfo = Get Payroll Area Control Record()
                    Pay Group:  Context.Current User.Payment Information .Worker.Employment Details.Compensation Information ☰.Pay Group.Pay Group ID
                                The rule selects one entry from the collection "Compensation Information" where...
                      User ID:  Context.Current User.Payment Information .Worker.Job Information.User ID
                   Start Date:  Context.Current User.Payment Information .Effective start date

If
  var_payrollControlRecordInfo.Is Payroll Data Change Allowed is equal to  No
  Then
    Raise Message  " PayrollDataChangeError " with Error severity
                   ERROR: {reason}
                     Reason:  var_payrollControlRecordInfo.Reason

Else If
  var_payrollControlRecordInfo.Reason is not equal to  Null
  Then
    Raise Message  " PayrollDataChangeWarning " with Warning severity
                   WARNING: {reason}
                     Warning:  var_payrollControlRecordInfo.Reason
```



5. This is an example where it is possible to revert to the previous behavior by submitting a query to the payroll control record status.

```
Variables
var_get_Payroll_Area_control_Record = Get Payroll Area Control Record()
                    Pay Group:  Payment Information .Worker.Employment Details.Compensation Information ≡.Pay Group.Pay Group ID
                                The rule selects one entry from the collection "Compensation Information" where...
                      User ID:  Payment Information .Worker.Job Information.User ID
                   Start Date:  Payment Information .Effective start date

If
  var_get_Payroll_Area_control_Record.Payroll Area Status is equal to  PAYROLL_CHECK_PAYROLL_RESULTS
  Then
    Raise Message  " Payroll Data Change Error Without reason " with Error severity
                   Write your own error message
```

16.112 Get Pensionable Salary

This function calculates the pensionable salary on a certain date (effective date). It is intended to be calculated when the compensation information is saved. The result of this function is then written into the compensation information field *Pensionable Salary*.

Limitations

Use only with Employee Central.

Input Parameters

When using this function, you have to enter the following parameters:

For this parameter...	Make this entry:
User	Select *Context* > *Current User* to get the user for whom the pensionable salary should be calculated. This is the actual user whose compensation information is accessed.
Pay Component Group	Enter the external code of the pay component group that contains the pay components subject to the pensionable salary calculation.
Effective Date	Select *Context* > *Effective Date* to get the date for when the pensionable salary should be calculated. This is usually the effective date on which compensation changes are made.
Day of Compare Date	The compare date marks the beginning of a period (typically the fiscal year) to which the salary cap refers. For example, if the fiscal year begins on April 1, the cap refers to the salary valid on that exact date. Enter an allowed value between 1 and 28, 29, 30, 31, depending on the number of days allowed for the month entered as compare date.
Month of Compare Date	Enter an allowed value between 1 and 12 (January – December).
Cap Percentage	This is the percentage value used to restrict the increase in pensionable salary. For example, if the salary (represented by the *Pay Component Group*) increases from 50,000 USD on compare date to 52,500 USD on the effective date (= 5 percent more), and the cap percentage is 1.0 percent, the increase of the pensionable salary will be restricted to 500 USD.
Pay Component Group Sum	Select the field that corresponds to the name of the pay component group that you have selected in the *Pay Component Group* field.

Please note that the rule can fail for the following reasons:

- At least one of the function parameters is not filled
- Compare day/month is not in expected value range
- Pay component group not found
- Job information or compensation information not found on the effective date or compare date

- Pensionable salary not available on compare date

If the rule fails, the *Pensionable Salary* is set to null.

Example

The user changes the compensation information for user cgrant on 08/20/2013. The external code of the relevant pay component group is *AnnualizedSalary*. The fiscal year ends on March 31 every year. You want to restrict the increase of the pensionable salary to, at most, 1 percent. Consequently, the rule function to calculate the pensionable salary is called with the following parameters:

- *User*: cgrant
- *Pay Component Group*: AnnualizedSalary
- *Effective Date*: 08/20/2013
- *Day of Compare Date*: 31
- *Month of Compare Date*: 3
- *Cap Percentage*: 1.0 percent
- *Pay Component Group Sum*: AnnualizedSalary

Starting from the effective date (08/20/2013), the last fiscal year end is calculated as 03/31/2013 (= compare date). For both dates, the salary is determined for the pay component group AnnualizedSalary, for example:

- Pensionable salary on compare date: 15,000 USD
- Salary on compare date: 50,000 USD
- Salary on effective date: 52,500 USD
- Salary change: +2,500 USD (= +5 percent)

The pensionable salary is increased by the same percentage as the salary (in the above example, 5 percent of 15,000 USD = 450 USD). However, as a cap percentage of 1 percent has been defined, the pensionable salary is increased by only 1 percent = 150 USD. So in this example, as the pensionable salary on compare date was 15,000 USD, the function returns a new pensionable salary of 15,150 USD on the effective date.

Please note that the employee's FTE is considered in the calculation. Let's look at this example:

- Pensionable salary on compare date: 15,000 USD
- Salary on compare date: 50,000 USD
- FTE on compare date: 1.0
- FTE on effective date: 0.5
- Salary on effective date: 25,250 USD
- Salary change: +250 USD (= +1 percent compared to the salary on compare date when this is calculated with FTE=0.5)

In this example, the salary and pensionable salary on compare date are calculated with FTE=0.5, resulting in a pensionable salary of 7,500 USD and a salary on compare date of 25,000 USD. The salary is changed by 250 USD (=+1 percent), so the pensionable salary is increased by 1 percent=75 USD. As a result, the new pensionable salary on the effective date is 7,575 USD.

Limitations

The customer has to enter a start value for the pensionable salary on the employee's hire date. So, if the rule is called with effective date = hire date, no new pensionable salary is calculated, but just the existing value is returned. This ensures that a manually entered start value is not overwritten during save. If the employee was not yet hired on the compare date, the hire date is used as the compare date for the calculation.

16.113 Get Pensionable Salary with Global Assignment

This function calculates the pensionable salary on a certain date (effective date). It is intended to be calculated when the Compensation Information is saved. The result of this function is then written into the Compensation Information field *Pensionable Salary*. This function considers an alternative baseline to calculate the pensionable salary considering global assignments.

Limitations

Use only with Employee Central and Global Assignments enabled.

Input Parameters

When using this function, you have to enter the following parameters:

For this parameter...	Make this entry:
User	Select *Context* > *Current User* to get the user for whom the pensionable salary should be calculated. This is the actual user whose compensation information is accessed.
Pay Component Group	Enter the external code of the pay component group that contains the pay components subject to the pensionable salary calculation.
Effective Date	Select *Context* > *Effective Date* to get the date for when the pensionable salary should be calculated. This is usually the effective date on which compensation changes are made.
Day of Compare Date	The compare date marks the beginning of a period (typically the fiscal year) to which the salary cap refers. For example, if the fiscal year begins on April 1, the cap refers to the salary valid on that exact date. Enter an allowed value between 1 and 28, 29, 30, 31, depending on the number of days allowed for the month entered as compare date.
Month of Compare Date	Enter an allowed value between 1 and 12 (January – December).

For this parameter...	Make this entry:
Cap Percentage	This is the percentage value used to restrict the increase in pensionable salary. For example, if the salary (represented by the *Pay Component Group*) increases from 50,000 USD on compare date to 52,500 USD on the effective date (= 5 percent more), and the cap percentage is 1.0 percent, the increase of the pensionable salary will be restricted to 500 USD.
Pay Component Group Sum	Select the field that corresponds to the name of the pay component group that you have selected in the *Pay Component Group* field.
Current FTE	Select the *FTE* field of the job information. **Note** If the FTE has not been added, *1* is used as default.
Alternative Baseline Event	You have two options: • If you want to use an alternative baseline for calculating the pensionable salary, select *Value* and select the corresponding event reason from the dropdown list. When the rule is triggered, the system checks if the compensation record has been changed between today and the compare date; and if yes, uses the alternative baseline to calculate the pensionable salary. • If you do not want to use an alternative baseline event, select *Null*.

Please note that the rule can fail for the following reasons:

- At least one of the function parameters is not filled
- Compare day/month is not in expected value range
- Pay component group not found
- Job Information or Compensation Information not found on the effective date or compare date
- Pensionable salary not available on compare date

If the rule fails, the *Pensionable Salary* is set to null.

Example

Scenario – an employee is placed on attachment with an increase to Basic Pay or Scenario – an employee with reduced pay

Limitations

The customer has to enter a start value for the pensionable salary on the employee's hire date. So, if the rule is called with effective date = hire date, no new pensionable salary is calculated, but just the existing value is returned. This ensures that a manually entered start value is not overwritten during save. If the employee was not yet hired on the compare date, the hire date is used as the compare date for the calculation.

16.114 Get Picklist Value Label

This function gets the label of a picklist value in the logged-on user's locale.

If the label isn't added in the logged-on user's locale, the system follows the standard behavior for languages and locales in SAP SuccessFactors systems.

Please be aware that in case of mass transactions (for example, mass import) this function can slow down performance.

Input Parameters

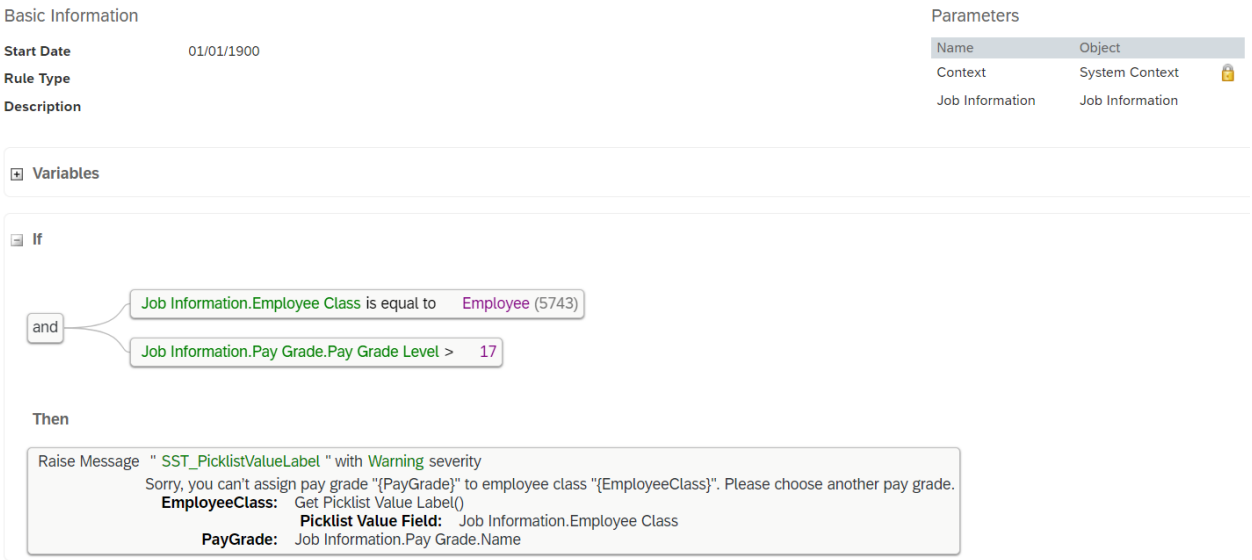
Input Parameter	Type	Required	User Entry
Picklist Value Field	Picklist	Yes	Select the field that contains the picklist value.

Behavior in Case of Errors

If the picklist value field is "null", the system returns "null".

Use Case

You want to raise a warning message when employees are assigned to a pay grade that is above their employee class. You use the rule function Get Picklist Value Label to make sure that the message displays the employee class affected in the user's locale.



Related Information

SAP SuccessFactors Languages and Locales

16.115 Get Previous Working Day()

This rule function determines the previous working day for a user and a specific date.

Overview

The calculation is based either on work schedule only or on work schedule and holiday calendar. The function tries to find the last working day within the last 365 days starting from the input date.In the case of a cross-midnight working day, the day on which the shift starts is defined as a working day. If there is no previous working day, the function returns “null”.

The following constellations lead to “null” as the return value:

- There is no job information in the past.
- There is job information in the past, but no work schedule is assigned.
- There is a job information in the past with a work schedule, but the work schedule has no working days.

Input Parameters

For this parameter	Make this entry
User ID	Select the *User* field. This is the user for which the previous working day will be determined.
Date	Select a date, which is used as a basis to determine the previous working day.
Consider Holidays	Choose whether holidays should be considered or not. If holidays are **not** taken into account, the previous working date is determined based on the work schedule. If holidays **are** taken into account, the previous working day is determined based on the work schedule and holiday calendar. If the day is a working day in the work schedule and there is no holiday, a none holiday, or a half-day holiday, the day is a working day. If the day is not a working day in the work schedule or there is a full holiday on that day, it is a non-working day.

Example 1: Non-Cross Midnight Work Schedule and Holidays Not Considered

Let's look at an example.

A.N.Other was hired on January 1, 2015 with a duration-based work schedule based on 8 hours working time Monday till Friday and Saturday and Sunday as days off. There is a public full day holiday on February 27, 2015.

Here's the data used to determine A.N. Other's previous working day:

- User ID: A.N.Other
- Date: March 1, 2015
- Consider Holidays: **NO**

Result 1: March 1, 2015 is a Sunday. The employee is not working on the previous day – February 28, 2015 – which is a Saturday. Therefore the function returns the previous Friday as previous working day: February 27, 2015, because holidays are not considered.

Example 2: Non-Cross Midnight Work Schedule and Holidays Considered

Here's another example.

A.N.Other was hired on January 1, 2015 with a duration-based work schedule based on 8 hours working time Monday till Friday and Saturday and Sunday as days off. There is a public full day holiday on February 27, 2015.

Here's the data used to determine A.N. Other's previous working day:

- User ID: A.N.Other
- Date: March 1, 2015
- Consider Holidays: **YES**

Result 2: March 1, 2015 is a Sunday. The employee is not working on the previous day – February 28, 2015 – which is a Saturday. On the next previous day there is a public holiday, so the function returns Thursday as previous working day: February 26, 2015.

Example 3: Cross-Midnight Work Schedule and Holidays Not Considered

Let's look at an example.

A.N.Other was hired on January 1, 2015 with a cross-midnight work schedule with working time Monday - Friday 22:00 - 06:00 (next day). Saturday and Sunday are days off. There is a public full day holiday on May 4, 2015.

Here's the data used to determine A.N. Other's previous working day:

- User ID: A.N.Other
- Date: March 1, 2015
- Consider Holidays: **NO**

Result 3: March 1, 2015 is a Sunday. The employee is not working on the previous day – February 28, 2015 – which is a Saturday. Although the employee is physically working on Saturday from 00:00 to 06:00, the working day is

specified as the day on which the shift starts, which, in this case, is Friday. So the function returns Friday as the previous working day: February 27, 2015, because holidays are not considered.

Example 4: Cross-Midnight Work Schedule and Holidays Considered

Here's another example.

A.N.Other was hired on January 1, 2015 with a cross-midnight work schedule with working time Monday - Friday 22:00 - 06:00 (next day). Saturday and Sunday are days off. There is a public full day holiday on February 27, 2015.

Here's the data used to determine A.N. Other's previous working day:

- User ID: A.N.Other
- Date: March 1, 2015
- Consider Holidays: **YES**

Result 4: March 1, 2015 is a Sunday. The employee is not working on the previous day – February 28, 2015 – which is a Saturday. Although the employee is physically working on Saturday from 00:00 to 06:00, the working day is specified as the day on which the shift starts, which, in this case, is Friday. So the function returns Thursday as the previous working day: February 26, 2015, because holidays are considered.

16.116 Get Purchased Leave Quantity or Equivalent Quantity (in Weeks) for Period

This rule function is used for purchased leave to get the sum of the time unit, either hours or days, or the equivalent quantity in weeks in a time period. It can be used to restrict the amount of leave purchased.

Use this rule function to prevent your users from purchasing leave of more than 12 weeks in a rolling period of 12 months.

This rule can also be used to validate that not more than 5 days of leave can be bought in a calendar year.

Input Parameters

Input Parameter	Type	Required	Make this entry
User ID	User	Yes	Enter the User ID of the user who has purchased leave.
Start Date	Date	Yes	Enter the start date of period.
End Date	Date	Yes	Enter the end date of period.
Time Account Type	Value	Yes	Enter the corresponding time account type that is being used for purchase leave.

Input Parameter	Type	Required	Make this entry
Time Account Purchase	Value	No	Enter `Time Account Purchase` here. **Note** You only need to use this parameter if you need the rule function to take account of a time account purchase that is not yet stored in the system.
Unit	Value	Yes	Select either: *Time Account Type Unit* unit, for the total purchased leave in units of days or hours. *Equivalent Units*, for the total purchased leave in units of weeks.

Behavior in Case of Errors

If the *Start Date* is greater than the *End Date*, the rule execution is stopped and an error message is shown.

Example Rule Limiting Leave Purchase Amount

Here are some examples of the If... Then statements that limit leave purchase to not more than 12-weeks during a 12-month period.

> **Note**
> You also create an error message for use in the Then statement of the rule.

Create the rule by going to the Admin Center and choosing *Configure Business Rules* > *Time Management (Rules scenario)* > *Time Account Purchase Leave Validation*. Create your rule name, using the information in the table below as an example.

If for *Get Purchased Leave Quantity or Equivalent Quantity (in Weeks) for Period()*:

Select Left Expression	Entry
User	Time Account Purchase User

Select Left Expression	Entry
Start Date	Date Plus()
Base Date	Time Account Purchase Requested Date
Number of Months	-12
Number of Days	0
End Date	Time Account Purchase Requested Date
Time Account Type	Time Account Purchase Time Account Type
Unit	Equivalent Units

Then

Raise Message **"<error message previously created, for example, EQWEEKS12>"** with *Error* severity.

You cannot request more than 12 weeks in a rolling period of 12 months.

Example Scenarios

The following table provides the information used for the three scenarios:

User	Time Account Type	Requested Date	Quantity	Equivalent Quantity	Approval Status
cgrant	purchaseleave	May 15, 2020	10 days	2	Approved
cgrant	purchaseleave	June 15, 2020	15 days	3	Pending
cgrant	purchaseleave	July 15, 2020	20 days	4	Approved
cgrant	purchaseleave	September 18, 2020	5 days	1	Decline
cgrant	purchaseleave	October 2, 2020	5 days	1	Cancelled
cgrant	purchaseleave	October 3, 2020	5 days	1	Approved
cgrant	purchaseleave	October 2, 2021	5 days	1	Approved
cgrant	purchaseleave	Current (today's) date	10 days	2	

Scenario 1: Get the sum of purchases from June 15, 2020 to June 14, 2021 in equivalent units.

User ID: cgrant

Time Account Type: Purchase leave

Unit: Equivalent Units

Start Date: June 15, 2020

End Date: June 14, 2021

Expected result: 3+4+1 = 8 Equivalent Quantity units

Scenario 2 : Get the sum of purchases from period June 15, 2020 to June 14, 2021 in time account type unit (Days/Hours).

User ID: cgrant

Time Account Type: Purchase leave

Unit: Time Account unit

Start Date: June 15, 2020

End Date: June 14, 2021

Expected result: Quantity 15+20+5 = 40 days

Scenario 3: Get the sum of purchased leave during the period January 1, 2020 to December 31, 2020.

User ID: cgrant

Time Account Type: Purchase leave

Unit: Equivalent Units

Start Date: January 1, 2020

End Date: December 31, 2020

Expected result: 2+3+4+1 = 10 Equivalent Quantity units

Scenario 4: Get the sum of purchase leave during the period from January 1, 2020, the current date to December 31, 2020. You are creating a new request for 10 days. The *Time Account Purchase* parameter is found in the system.

User ID: cgrant

Time Account Type: Purchase leave

Unit: Equivalent Units

Start Date: January 1, 2020

End Date: December 31, 2020

Expected result: 2+3+4+1+10 = 20 Equivalent Quantity units

16.117 Get Retention Period Start Date for User

This function returns the start date of the retention period for the specified user. For purging data, any data after the retention period start date is not purged, while data before the start date is deleted.

Configuration Requirements

You've enabled Time Off.

Input Parameters

Input Parameter	Type	Required	User Entry
User	User	Yes	Enter the name of the user for which you want to retrieve the start date of the retention period.

Behavior in Case of Errors

If no retention period start date has been defined, the rule function returns "null".

Use Case

No data is purged after the retention period start date. You can use this rule function to determine the retention period start date of a user.

16.118 Get Start Date of Next Pay Period

This function returns the start date of the next pay period in the pay calendar based on user ID and base date.

Input Parameters

Input Parameter	Type	Required	User Entry
Base Date	Date	Yes	Enter the date based on which the next pay period is determined and the start date is returned.
User ID	Text	Yes	Enter the user ID of the user for whom the start date of the next pay period is fetched.

Behavior in Case of Errors

The rule function returns "null" in the following scenarios:

- *Compensation Information* is not maintained for the entered user ID.
- *Pay Group* is not maintained in *Compensation Information*.
- *Pay Periods* are not maintained in *Pay Calendar*.
- The pay period next to the pay period of the base date is not maintained.

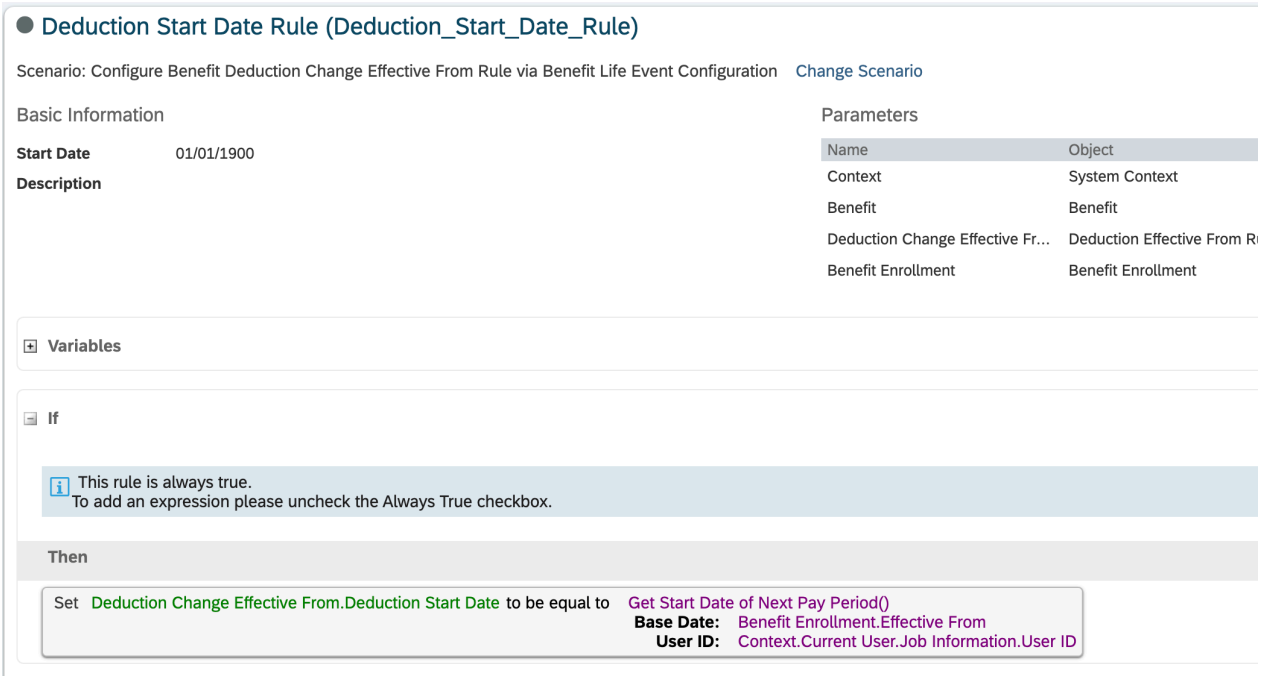
Example

User ID	Base Date	Result	Details
JOHN	March 23, 2020	April 1, 2020	The pay period start date returned by the rule function depends on the pay period configuration in the pay calendar of the user.
CARLA	October 19, 2020	November 1, 2020	The pay period start date returned by the rule function depends on the pay period configuration in the pay calendar of the user.

Use Case

An administrator wants to configure a benefit for which the deduction start date is always the start date of the next pay period irrespective of the pay check issue date. This scenario can be achieved by using the *Get Start Date of Next Pay Period* rule function as this rule function returns the start date of the next pay period for the provided

User ID and date. The following screenshot shows a rule created using the *Get Start Date of Next Pay Period* rule function:



16.119 Get Start Date of Next Pay Period Based on Pay Check Issue Date

This function returns the start date of the next pay period in the pay calendar based on the pay check issue date. If the enrollment effective date is on or after the pay check issue date, the function returns the start date of the next pay period. If the enrollment effective date is before the pay check issue date, the function returns the enrollment effective date.

Input Parameters

Input Parameter	Type	Required	User Entry
Base Date	Date	Yes	Enter the date based on which the next pay period is determined and the start date is returned.
User ID	Text	Yes	Enter the user ID of the user for whom the start date of the next pay period is fetched.

Behavior in Case of Errors

The rule function returns "null" in the following scenarios:

- *Compensation Information* is not found for the entered user ID.
- *Pay Group* is not found in *Compensation Information*.
- *Pay Periods* are not found in *Pay Calendar*.
- The pay period next to the pay period of the base date is not found.

Example

An employee wants to enroll into a plan. The pay periods are configured monthly as shown below:

Pay Period Configuration

Pay Period Start Date	Pay Period End Date	Pay Check Issue Date
Jan 1st, 2021	Jan 31st, 2021	Jan 25th, 2021
December 1st, 2020	December 31st, 2020	December 25th, 2020
November 1st, 2020	November 30th, 2020	November 25th, 2020
October 1st, 2020	October 31st, 2020	October 25th, 2020
September 1st, 2020	September 30th, 2020	September 25th, 2020

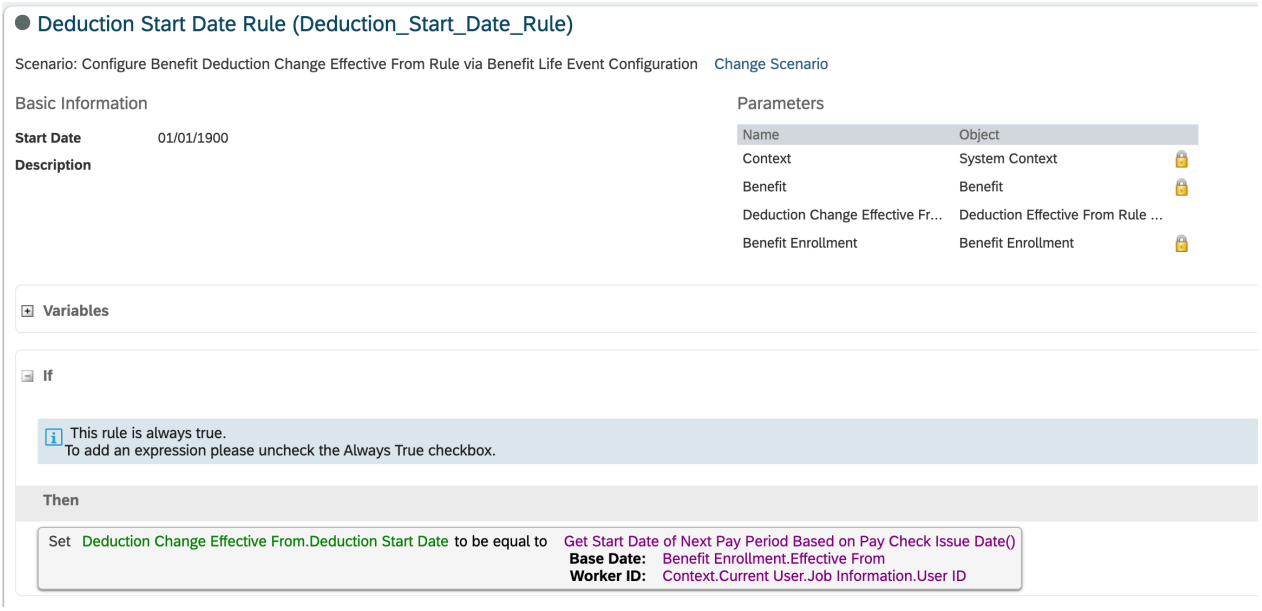
Use Case 1: If the employee enrolls into the plan on September 20th, 2020, the deduction start date will be September 20th, 2020 as the enrollment effective date is before the pay check issue date September 25th, 2020.

Use Case 2: If the employee enrolls into the plan on September 26th, 2020, the deduction start date will be October 1st, 2020 as the enrollment effective date is after the pay check issue date September 25th, 2020.

Use Case

An administrator wants to configure a benefit for which the deduction start date is based on the the pay check issue date. This scenario can be achieved by using the *Get Start Date of Next Pay Period Based on Pay Check Issue*

Date rule function as this rule function returns a date based on the pay check issue date. The following screenshot shows a rule created using the *Get Start Date of Next Pay Period Based on Pay Check Issue Date* rule function:



16.120 Get Start Time Of Working Day ()

This function calculates the start time of the working day on a given date.

It reads the work schedule associated with the user to calculate the start time of the date provided to the rule function. The rule function returns a value in 24 hour clock format.

If the user applied for time off using clock times, you can use this rule function for specific validation against the beginning of the day. This is typically done in a take rule (absence validation rule) that you can set to be triggered when absence is created or modified.

The rule function returns null if:

- A duration-based work schedule is assigned to the user as of the requested date.
- The requested date is a non-working day.

Input Parameters

For this parameter.	Make this entry
User	Select the *User* field. This is the user for which the start time of working day should be calculated.Prerequisites: User A.N.Other should have Time Recording Variant as Clock Time and Work Schedule should be configured for working days.
Date	Date for which the start time is required

Example 1: Non-Cross Midnight Work Schedule

Here's the data for A.N.Other's start time calculation:

User ID: A.N. Other

Date = July 1, 2020 (Wednesday)

Prerequisites: A.N.Other should have the Clock Times time recording variant. The work schedule is configured with Monday to Friday as working days from 08:00 to 17:00. Saturday and Sunday are non-working days.

Result 1: The rule function returns a start time of 08:00:00

Example 2: Cross-Midnight Work Schedule

This time, A.N Other's information looks like this:

- User: A.N.Other
- Date: Monday, July 6, 2020

Prerequisites: A.N.Other should have the Clock Times time recording variant. The work schedule is configured with Monday to Friday as working days from 22:00 to 06:00 (next day). Saturday and Sunday are non-working days.

Result 2: The rule function returns a start time of 22:00:00.

16.121 Get Singapore Citizenship Acquisition Date For Dependent()

This rule function is Singapore-specific. It gets the date a dependent acquired Singapore citizenship for a given list of relationship types and ranks on a given date.

The rule function is required for use cases where certain benefits are given only to citizens of Singapore. It also considers dependents where no date of birth is found in the system. Dependents are sorted based on date of birth. In the case of twins or triplets, or if no date of birth is found in the system, sorting is done by last name followed by first name. Dependents where no date of birth is not found in the system are added to the end of sorted list.

Input Parameters

For this parameter	Make this entry
Employee Person ID	Person ID external, accessed from the biographical information on the rule UI.

For this parameter	Make this entry
User ID	User ID for which youngest dependent is calculated.
Date	The date on which youngest dependent is fetched.
Relationship Types	List of relationship types for which dependents will be fetched.
Rank	Rank of dependent whose age is to be calculated.
Citizenship Type 1	Citizenship type to check whether dependent is Singapore citizen.
Citizenship Type 2 (optional)	Citizenship type to check whether dependent is Singapore citizen.
Citizenship Type 3 (optional)	Citizenship type to check whether dependent is Singapore citizen.
Citizenship Type 4 (optional)	Citizenship type to check whether dependent is Singapore citizen.
Citizenship Type 5 (optional)	Citizenship type to check whether dependent is Singapore citizen.

Examples

Let's look at an example.

Employee is hired on January 1, 2010, so the dependent card has the following data for the employee's dependents:

Relationship Type	Date of Birth	Last Name	First Name		Date Citizenship Acquired
Child	September 1, 2009	Klarc	Adam	By Birth	September 5, 2009
Child	September 1, 2009	Calley	Bob	By Birth	September 5, 2009
Mother		Eethen	Alen	Not a Singapore citizen	
Child		Eethen	Gable	By Descent	January 1, 2015
Spouse	January 27, 2016	Alaric	Anna	By Descent	

Sorting results in the following ranking:

Sorted Dependents

Rank	Dependent
1	Klarc, Adam
2	Calley, Bob
3	Alaric, Anna
4	Eethen, Alen
5	Eethen, Gable

To get the date of acquisition of citizenship type of dependent for the rank 5 for the relationship type Child when this rule function is called, date of acquisition of citizenship type of dependent returned is January 1, 2015.

16.122 Get Sum Of Accruals Since Last Transfer()

This function is required to retrieve the sum of accruals of a given time account with entitlement method “Entitled as Transferred” or time type since the last transfer date (ad-hoc transfer or regular transfer, whichever is later).

> ⚠ Restriction
>
> Do not use this rule function for regular accounts (“Entitled as Accrued”) - use the "Calculate Balance" rule function instead.

This rule function only considers accruals since the last entitlement transfer. It can be used, for example, to cap the accruals within an entitlement period to a maximum value. If the time variant “clock time” is used, this rule function can be used for a specific validation against the beginning of the day. This is typically done in the absence validation rule (Take Rule) which can be set to be triggered when an absence is created or modified.

> ⓘ Note
>
> If both the time type and the time account are passed, precedence is given to the time account
>
> The simulation should be used with great care. It will not work if the rule function is used inside the accrual rule of a time account that is also used as a parameter in the rule function.

Input Parameters

For This Field	Make This Entry
User ID	Select the User field. This is the user for which the balance is calculated.
Date	Date for which the start time is required.
Time Type	Select the time type for which the balance is to be calculated.
Time Account	Select the time account for which the balance is to be calculated.
Simulate Accruals and Entitlements	Boolean indicator for Simulation

Examples

Let's look at some examples.

Example 1

The employee is hired on January 1, 2015, the accrual frequency is monthly and he accrues 3 days every month.

His Next Transfer Date is on January 1, 2016, and he has accrual balance of 09 days as of March 1, 2015

When we call this rule function to retrieve the accrual balance as of Nov 1, 2015, the rule function will return 9 days.

Example 2

The employee is hired on January 1, 2015, the accrual frequency is monthly and she accrues 3 days every month.

The last entitlement transfer happened on June 30, 2015. When we call this rule function to retrieve the accrual balance as of August 1, 2015, the rule function will return 6 days (3 days for July + 3 days for August). The accruals before the last transfer date are not considered for this rule function.

16.123 Get Sum Of Advances()

This rule function is required to retrieve the advance balance from a given time account on a given date. The rule function only makes sense for Time Accounts with entitlement method “Entitled as Transferred” because for other accounts there are no advances.

> ### Note
>
> Using this rule function only makes sense for time accounts with entitlement method “Entitled as Transferred” because for other accounts there are no advances.

The advance balance is the total of all employee times that are marked as advance and have a booking date before or equal to the given date.

Example

Here's an example.

An employee has an entitlement balance of 2 days and he has already accrued 3 days since the last transfer. The next transfer will be on January 1, 2016.

Now the employee wants to create an absence of 5 days. Since the entitlement balance is 2, the remaining 3 days have to be flagged as advances.

Now let us assume we execute the rule function on December 24, 2015. This means the transfer has not yet been executed. If we executed the rule function for the date parameter January 1, 2016 the rule function would return 3.

If we executed the rule function again with the very same parameters after January 1, 2016 (after the transfer) the return value would be 0, because when the transfer is happening the advances are immediately removed if the newly created entitlements allow it.

Input Parameters

For This Field	Make This Entry
User ID	Select the User field. This is the user for which the balance is calculated.
Date	Date for which the start time is required.
Time Type	Select the time type for which the balance is to be calculated.
Time Account	Select the time account for which the balance is to be calculated.

16.124 Get Time Management Earliest Recalculation Date ()

This rule function returns the earliest date only if recalculation is active.

Otherwise, it returns January 1, 1900, as the date. This means that recalculation is always active.

However, if you need to restrict the earliest date from which recalculation should be done, you can use this configuration. Time Management Configuration is checked to verify whether recalculation is active and determine the earliest date.

The primary use case for this rule function is in absence validation rules (take rules) to control changes in Employee Time.

16.125 Get Time Sheet Approval Workflow Configuration

When you submit a time sheet, the system checks if a workflow has been configured. This rule function gets the workflow configuration which is stored as an internal ID in the time recording profile of the person for whom the time sheet has been created.

Limitations

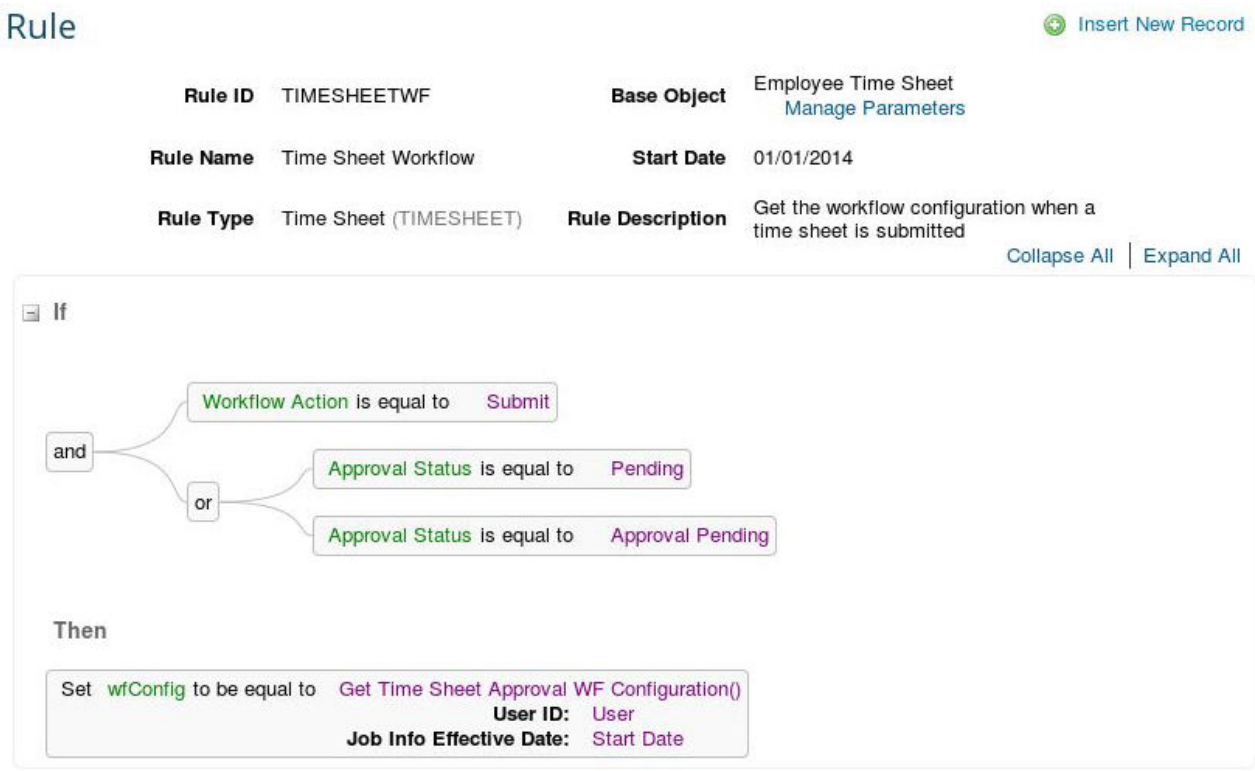
Use only with Time Sheet.

Input Parameters

For this parameter...	Make this entry:
User ID	This is the user for whom the time sheet is created. Select *User* from the dropdown list. This option appears when you have selected *Employee Time Sheet* as base object.
Job Info Effective Date	This is the relevant date for the job info of the user. The system reads the time recording profile from this job info record is read and uses the corresponding workflow that has been configured for it. For example: • The workflow configuration valid at the time sheet start date is relevant: Select the field *Start Date* from the dropdown list. • The workflow configuration valid at the current date is relevant: Select the function *Today()* from the dropdown list.

Use Case

Here's an example for a complete rule:



16.126 Get Weekdays

This function returns a list of weekdays.

Configuration Requirements

You've enabled Time Off.

Input Parameters

Input Parameter	Type	Required	User Entry
Weekday	Value	Yes	Enter the days that are weekdays. Enter: • Any day of the week except Saturday and Sunday, or in some regions/countries, except only Sunday.

Behavior in Case of Errors

If *Weekday* is "null", the result is "null".

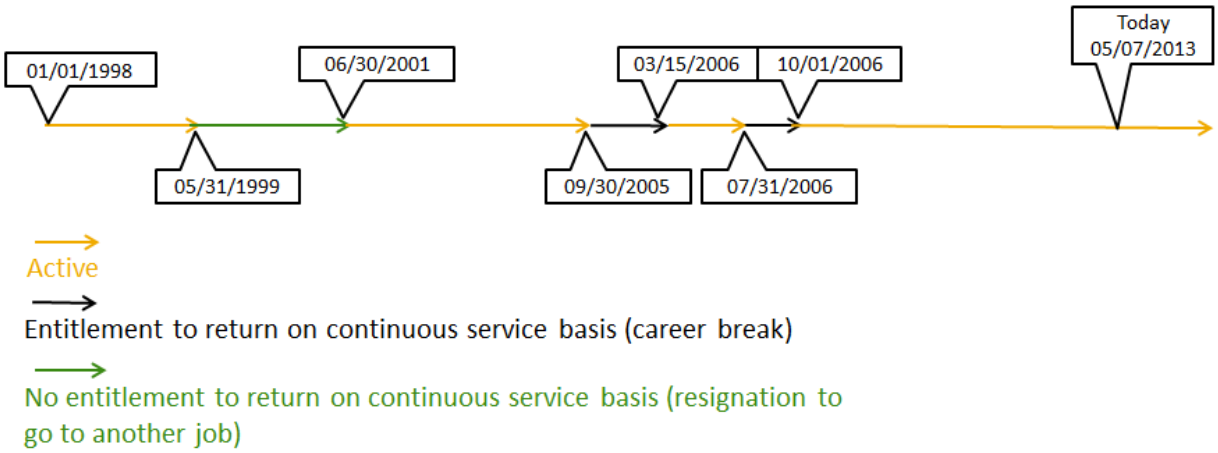
Use Case

The rule function returns the list of weekdays based on the weekdays specified in the input. It can be used when you need the list of weekdays in your business rule.

16.127 Get Work History Days ADD ALL

If an employee has repeatedly been employed at one company, you can sum up the days of all employments using this function.

Here's an example:



Calculation:

All orange arrows = 516 + 1554 + 139 + 2410 = 4619 days

This function sums up the valid days of all Job Information entries where:

- Employee is active and employee class is not excluded
- Employee is not active and event is not Termination

The final end date for the sum calculation is the day when the function is executed.

Note that this function is used to cover country/region-specific and industry-specific requirements for the Public Sector in the United Kingdom.

Limitations

- Use only with Employee Central
- Use only with Job Information as the base object
- Use an *onView* event as the work history in days has the character of transient data and not of a persistent value

Input Parameters

For this parameter...	Make this entry:
User	Select *Context* > *Current User* from the dropdown list. This gets the user ID of the user for whom the days should be counted. For example, if the admin looks at the Job Information for Carla Grant, the validity in days for Carla Grant is calculated and displayed.
Employee Class	If you want to exclude Job Information entries with one of these employee classes from the sum, although the employee is active, then enter the corresponding employee class here. For example, you want to exclude periods during which the employee worked as contractor. Then you have to enter the external code for contractors of the employee-class picklist. If you leave this field empty, **all** employee classes are included in the calculation.

Examples

Example 1

User: 1

Employee Class: Contractor

Job Info entry	Employee is:	Event	Employee class	Effective start date
No. 1 (User 1)	Active	Hire	Employee	01/01/2013
No. 2 (User 1)	Inactive	Unpaid Leave	Employee	06/01/2013

This happens during calculation:

No. 1 counts for sum, because employee is active and employee class is not Contractor.

No. 2 counts for sum, because employee is inactive, but event is not Termination.

Example 2

User: 1

Employee Class: Contractor

Job Info entry	Employee is:	Event	Employee class	Effective start date
No. 1 (User 1)	Active	Hire	Contractor	01/01/2013
No. 2 (User 1)	Inactive	Termination	Contractor	06/01/2013
No. 3 (User 1)	Active	Rehire	Employee	11/01/2013

This happens during calculation:

No. 1 doesn't count for sum calculation because the employee is active but the employee class is Contractor.

No. 2 doesn't count for sum calculation because the employee is inactive and the event is Termination.

No. 3 counts for sum calculation because the employee is active and the employee class is not Contractor.

Use Case

A company wants to foster loyalty of its employees by providing awards for long service and provides "Long Service Awards". To enable all managers as well as employees to monitor their service length, the function can be used to show the aggregated service length in Employment Details. Using the function to fill a customer field on loading the page provides the user always with the actual count of their service length.

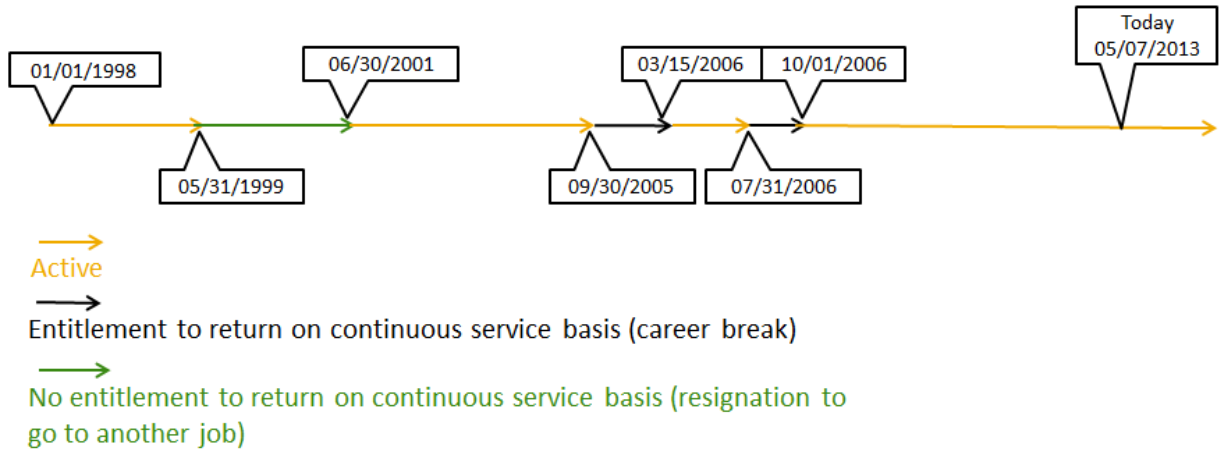
See also an example rule under Get Work History Days PREVIOUS [page 321].

16.128 Get Work History Days CONTINUOUS

If an employee has repeatedly been employed at one company, you can sum up the days of all consecutive employments that were ended with a specific event reason (for example, Entitlement to Return). The function

sums up all consecutive employments from today until a previous employment that was ended with a different event reason.

Here's an example:



Calculation:

Orange arrows (from 30/06/2001 onward) = 1554 + 139 + 2410 = 4103 days

The function sums up the valid days of all Job Information entries after the latest terminated entry without the specified event reason(s) where:

- Employee is active and employee class is not excluded
- Employee is not active and event is not termination

The final end date for the sum calculation is the day when the function is executed.

If the user status is currently terminated or retired, or the employee class is excluded, this function returns 0.

Note that this function is used to cover country/region-specific and industry-specific requirements for the Public Sector in the United Kingdom.

Limitations

- Use only with Employee Central.
- Use an *onView* event as the work history in days has the character of transient data and not of a persistent value.

Input Parameters

For this parameter...	Make this entry:
User	Select *Context* > *Current User* from the dropdown list. This gets the user ID of the user whose days should be counted. For example, if the admin looks at the Job Information for Carla Grant, the validity in days for Carla Grant is calculated and displayed.
Employee Class	If you want to exclude Job Information entries with one of these employee classes from the sum, although the employee is active, then enter the corresponding employee class here. For example, you want to exclude periods during which the employee worked as a contractor. Then you have to enter the external code for contractors of the *employee-class* picklist. If you leave this field empty, **all** employee classes are included in the calculation.
Event Reason 1-5	Select the exceptional event reason that does not end the count of consecutive periods of employment included in the calculation done by this function. You can select up to 5 event reasons.

Examples

Example 1

User: 1

Employee Class: Contractor

Event Reason: Entitlement to return

Job Info entry	Employee is	Event	Event Reason	Employee class
No. 1 (User 1)	Active	Hire	New Hire	Employee
No. 2 (User 1)	Inactive	Unpaid Leave	Maternity	Employee

This happens during calculation:

There is no termination without this event reason. Both entries count for sum calculation.

Example 2

User: 1

Employee Class: Contractor

Event Reason: Entitlement to return

Job Info entry	Employee is	Event	Event Reason	Employee class
No. 1 (User 1)	Active	Hire	New Hire	Employee

Job Info entry	Employee is	Event	Event Reason	Employee class
No. 2 (User 1)	Inactive	Termination	Entitlement to return	Employee
No. 3 (User 1)	Active	Rehire	New Rehire	Employee

This happens during calculation:

No. 3 counts for sum calculation because the employee is active and the employee class is not Contractor.

No. 2 is a Termination event, but with the special event reason. Therefore, the count is not stopped.

No. 1 counts for sum calculation because the employee is active and the employee class is not Contractor.

Example 3

User: 1

Employee Class: Contractor

Event Reason: Entitlement to return

Job Info entry	Employee is	Event	Event Reason	Employee class
No. 1 (User 1)	Active	Hire	New Hire	Employee
No. 2 (User 1)	Inactive	Termination	Quit	Employee
No. 3 (User 1)	Active	Rehire	New Rehire	Employee

This happens during calculation:

No. 3 counts for sum calculation because the employee is active and the employee class is not Contractor.

No. 2 is a Termination event and does not have the special event reason. Therefore, the count will be stopped and No. 1 does **not** count in sum.

Example 4

User: 1

Employee Class: Contractor

Event Reason: Entitlement to return

Job Info entry	Employee is	Event	Event Reason	Employee class
No. 1 (User 1)	Active	Hire	New Hire	Employee
No. 2 (User 1)	Inactive	Termination	quit	Employee
No. 3 (User 1)	Active	Rehire	New Rehire	Employee
No. 4 (User 1)	Inactive	Termination	Entitlement to return	Employee
No. 5 (User 1)	Active	Rehire	New Rehire	Employee
No. 6 (User 1)	Active	Data Change	Manager change	Employee

This happens during calculation:

No. 6 counts for sum calculation because the employee is active and the employee class is not Contractor.

No. 5 counts for sum calculation because the employee is active and the employee class is not Contractor.

No. 4 doesn't count and doesn't stop the count because it is a Termination event with the special event reason.

No. 3 counts for sum calculation because the employee is active and the employee class is not Contractor.

No. 2 doesn't count for sum and **stops** the count because the event is Termination without the special event reason.

No. 1 does not count because the count has been stopped at No. 2.

Use Case

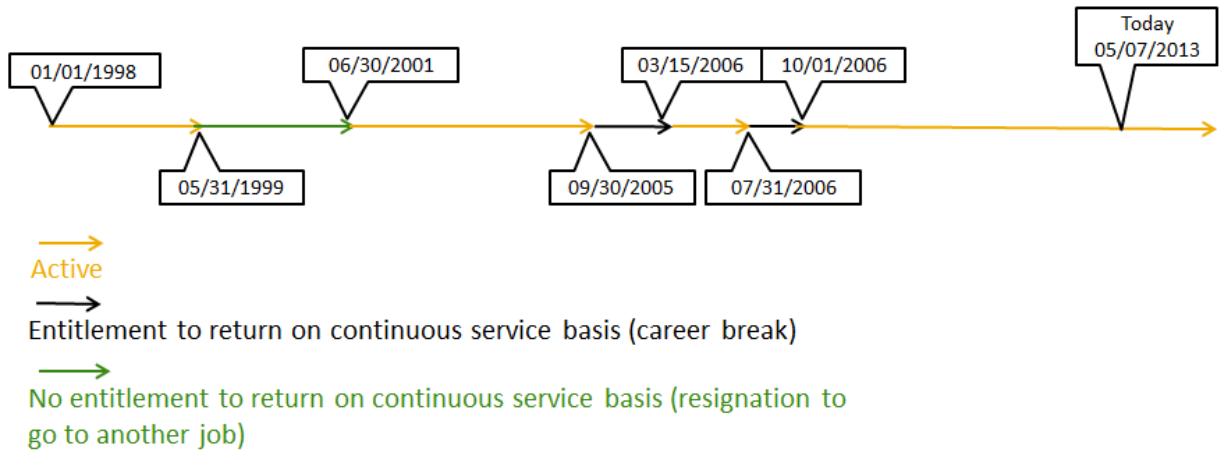
In countries/regions such as the United Kingdom (UK), redundancy payments are dependent on benchmarks such as continuous service. The employers and employees alike can use such figures to calculate redundancy payments using redundancy calculators to assess the actual redundancy payments employees can expect in case of unemployment. Using this function to fill a customer field on loading the page provides users always with the actual figure of their continuous service length.

See also an example rule under Get Work History Days PREVIOUS [page 321].

16.129 Get Work History Days CURRENT

You can sum up the days of only the current employment even if the employee has been employed at one company several times.

Here's an example:



Calculation:

Right orange arrow only = 05/07/2013 – 10/01/2006 = 2410 days

This function sums up the valid days of all Job Information entries after the latest terminated entry where:

- Employee is active and employee class is not excluded
- Employee is not active and event is not Termination

The final end date for the sum calculation is the day when the function is executed.

If the user status is currently terminated or retired, or the employee class is excluded, this function returns 0.

Note that this function is used to cover country/region-specific and industry-specific requirements for the Public Sector in the United Kingdom.

Limitations

- Use only with Employee Central.
- Use an *onView* event as the work history in days has the character of transient data and not of a persistent value.

Input Parameters

For this parameter...	Make this entry:
User	Select *Context* > *Current User* from the dropdown list. This gets the user ID of the user for whom the days should be counted. For example, if the admin looks at the Job Information for Carla Grant, the validity in days for Carla Grant is calculated and displayed.
Employee Class	If you want to exclude Job Information entries with one of these employee classes from the sum, although the employee is active, then enter the corresponding employee class here. For example, you want to exclude periods during which the employee worked as a contractor. Then you have to enter the external code for contractors of the *employee-class* picklist. If you leave this field empty, **all** employee classes are included in the calculation.

Examples

Example 1

User: 1

Employee Class: Contractor

Job Info entry	Employee is	Event	Employee class	Effective start date
No. 1 (User 1)	Active	Hire	Employee	01/01/2013

Job Info entry	Employee is	Event	Employee class	Effective start date
No. 2 (User 1)	Inactive	Unpaid Leave	Employee	06/01/2013

This happens during calculation:

There is no termination event used, so the result is the same as for Get Work History Days ADD ALL.

Example 2

User: 1

Employee Class: Contractor

Job Info entry	Employee is	Event	Employee class	Effective start date
No. 1 (User 1)	Active	Hire	Contractor	01/01/2013
No. 2 (User 1)	Inactive	Termination	Contractor	06/01/2013
No. 3 (User 1)	Active	Rehire	Employee	11/01/2013

This happens during calculation:

No. 2 is the key entry with the event Termination.

Only No. 3 counts for sum calculation because the employee is active and the employee class is not Contractor.

Example 3

User: 1

Employee Class: Contractor

Job Info entry	Employee is	Event	Employee class	Effective start date
No. 1 (User 1)	Active	Hire	Contractor	01/01/2013
No. 2 (User 1)	Inactive	Termination	Contractor	02/01/2013
No. 3 (User 1)	Active	Rehire	Employee	04/01/2013
No. 4 (User 1)	Inactive	Termination	Employee	06/01/2013
No. 5 (User 1)	Active	Rehire	Employee	08/01/2013
No. 6 (User 1)	Active	Data Change	Employee	10/01/2013

This happens during calculation:

No. 4 is the key entry with termination because it is the latest one.

No. 5 and No. 6 both count for sum calculation because the employee is active and the employee class is not Contractor.

Use Case

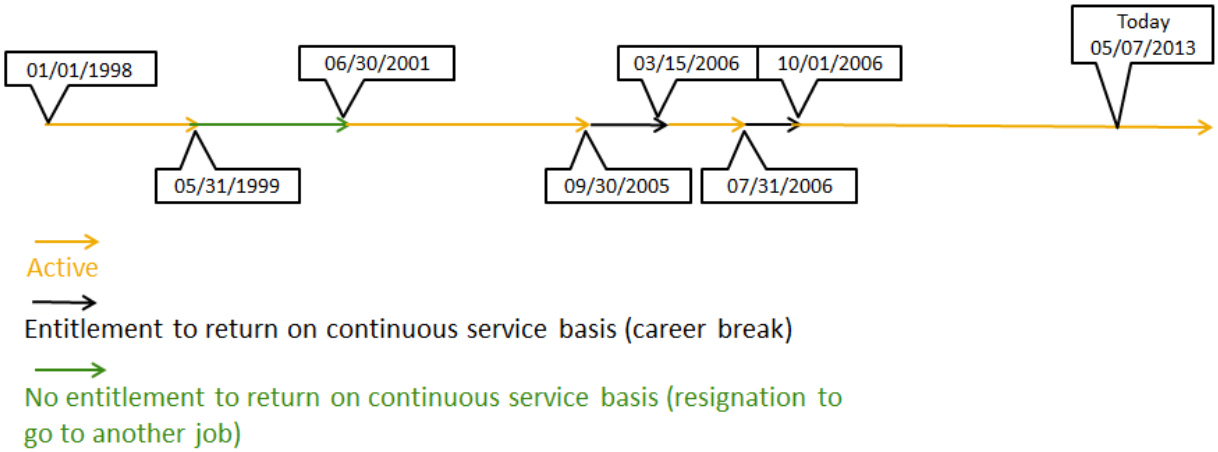
This function might be used to show how long an employee has served in the current employment. Such a figure might be interesting context information for the employee's line manager or HR Admin.

See also an example rule under Get Work History Days PREVIOUS [page 321].

16.130 Get Work History Days PREVIOUS

If an employee has repeatedly been employed at one company, you can sum up all previous employments excluding the current employment.

Here's an example:



Calculation:

All active employments – Current employment (4619 – 2410) = 2209 days

The function sums up the valid days of all Job Info entries before the latest terminated entry where:

- Employee is active and employee class is not excluded
- Employee is not active and event is not termination

Limitations

- Use only with Employee Central.
- Use an *onView* event as the work history in days has the character of transient data and not of a persistent value.

Input Parameters

For this parameter...	Make this entry:
User	Select *Context* > *Current User* from the dropdown list. This gets the user ID of the user whose days should be counted. For example, if the Admin looks at the Job Information for Carla Grant, the validity in days for Carla Grant is calculated and displayed.
Employee Class	If you want to exclude job info entries with one of these employee classes from the sum, although the employee is active, then enter the corresponding employee class here. For example, you want to exclude periods during which the employee worked as a contractor. Then you have to enter the external code for contractors of the *employee-class* picklist. If you leave this field empty, **all** employee classes are included in the calculation.

Examples

Example 1

User: 1

Employee Class: Contractor

Job Info entry	Employee is	Event	Employee class	Effective start date
No. 1 (User 1)	Active	Hire	Employee	01/01/2013
No. 2 (User 1)	Inactive	Unpaid Leave	Employee	06/01/2013

This happens during calculation:

No Job Info entry counts because there is no termination event used.

Example 2

User: 1

Employee Class: Contractor

Job Info entry	Employee is	Event	Employee class	Effective start date
No. 1 (User 1)	Active	Hire	Contractor	01/01/2013
No. 2 (User 1)	Inactive	Termination	Contractor	06/01/2013
No. 3 (User 1)	Active	Rehire	Employee	11/01/2013

This happens during calculation:

No. 2 is the key entry with the Termination event.

No. 1 does **not** count for the sum calculation because the employee is active and the employee class is Contractor.

Example 3

User: 1

Employee Class: Contractor

Job Info entry	Employee is	Event	Employee class	Effective start date
No. 1 (User 1)	Active	Hire	Contractor	01/01/2013
No. 2 (User 1)	Inactive	Termination	Contractor	02/01/2013
No. 3 (User 1)	Active	Rehire	Employee	04/01/2013
No. 4 (User 1)	Inactive	Termination	Employee	06/01/2013
No. 5 (User 1)	Active	Rehire	Employee	08/01/2013
No. 6 (User 1)	Active	Data Change	Employee	10/01/2013

This happens during calculation:

No. 4 is the key entry with the Termination because it is the latest one.

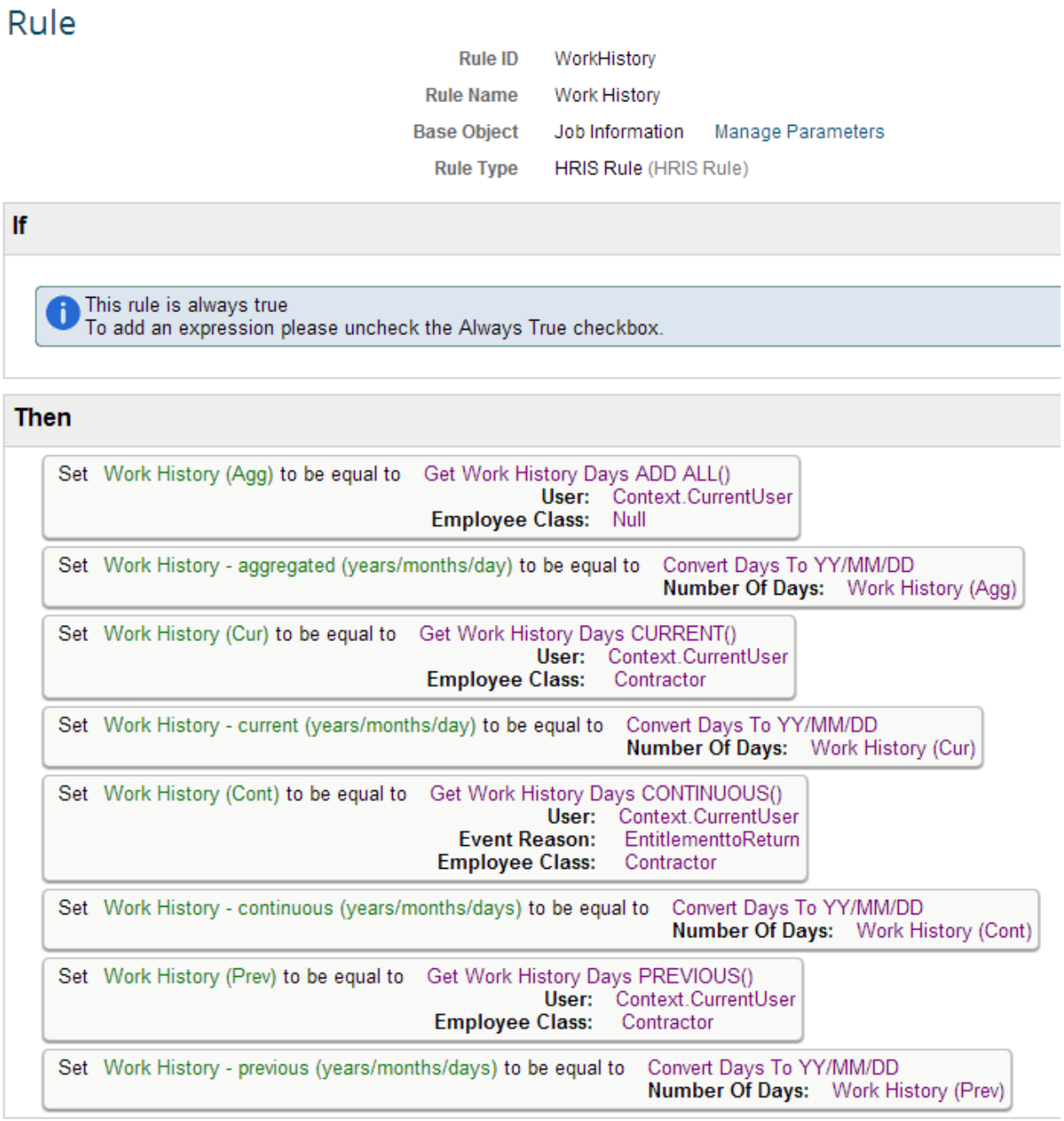
No. 1 does **not** count for sum calculation because the employee is active, but the employee class is Contractor.

No. 2 does **not** count for sum calculation because the employee is inactive and the event is Termination.

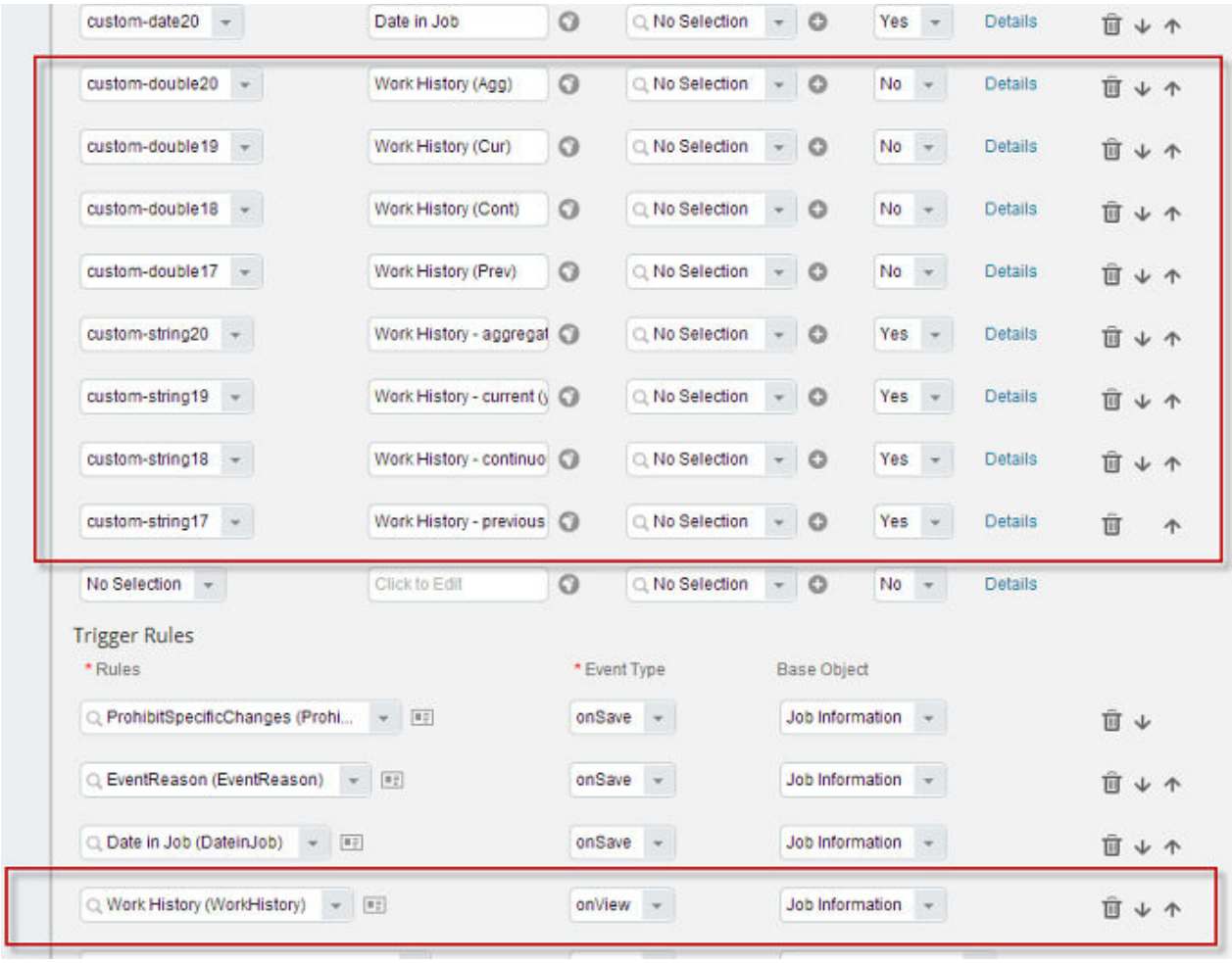
No. 3 counts for sum calculation because the employee is active and the employee class is not Contractor.

Use Case

The Admin wants to calculate the work history in days using all *Get Work History* functions; the result is displayed in the readable format of years/months/days in custom fields. This is the rule the Admin has created:



The result of the calculation is displayed on these custom fields, which the customer created using the Business Configuration UI:



Note: You can also use the data model XML files to create these custom fields and assign the rule to a field.

These fields are part of Job Information (*jobInfo* HRIS element). At the top, you can see the custom fields; at the bottom you can see the *onView* event that triggers the rule. For each work history calculation result, one custom field (of type "double") is filled with the result in days of the corresponding function. As this result is not readable when it is a high number (for example, 1245 days), the Admin has created one custom field (of type "string"), which is filled with the days reformatted into years/months/days using the *Convert Days Into YY/MM/DD* function. Those fields are set to "visible", whereas the double fields are kept invisible.

16.131 Get Working Time Account on Key Date()

This rule function gets the working time account type from the time type profile assigned to the user's job info profile on the key date.

Use this rule function in conjunction with the Time Off rule function *Get Balance For Types()* and the Time Sheet rule function *Get Number of Valuated Hours for Time Sheet()*.

Input Parameters

Here's a table showing the entries you need to make in the input parameters:

For this parameter:	Make this entry
User ID	The user for whom you need this information.
Date	The key date as of which you need this information.

16.132 Get Working Time In Hours ()

This function calculates the working time in hours for the given duration and given time, based on the time recording variant and work schedule configured for the relevant user.

If the time recording variant is **`Duration`**, the start time and end time will not be considered for the calculation. If the time recording variant is **`Clock Times`** then the start time and end time **are** considered for the calculation.

> ⓘ Note
>
> This function does not include the holiday information. This function provides you with the planned working time based on the work schedule.

If 00:00:00 is entered as the start or end time, it is defined as the start of the corresponding date.

Input Parameters

For this parameter...	Make this entry:
User	Select the *User* field. This is the user for which the start time of working day should be calculated.
Start Date	The start date of the duration.
End Date	The end date of the duration.
Start Time	The start time of the duration.
End Time	The end time of the duration.

Example 1: Clock-Time Based Employee with Non-Cross Midnight Work Schedule

Let's look at an example.

Here's the data for A.N.Other's working time calculation:

User ID = A.N. Other

Start Date = January 4, 2016

End Date = January 6, 2016

Start Time = 13:00:00

End Time = 14:00:00

The work schedule is configured like this:

- Day 1: Working time starting at 08:00:00 and ending at 17:00:00, with an unpaid break between 12:00:00 and 13:00:00 (4 hours)
- Day 2: Working time starting at 09:00:00 and ending at 18:00:00, with an unpaid break between 13:00:00 and 14:00:00 (8 hours)
- Day 3: Working time starting at 10:00:00 and ending at 18:00:00, with an unpaid break between 11:00:00 and 12:00:00 (3 hours)

Result: The rule function returns a working time of 15 hours. That is comprised of 4 hours on day 1, 8 hours on day 2, and 3 hours on day 3.

Example 2: Clock-Time Based Employee with Cross-Midnight Work Schedule

Let's look at an example.

Here's the data for A.N.Other's working time calculation:

User ID = A.N. Other

Start Date = January 3, 2016

End Date = January 6, 2016

Start Time = 00:00:00 (midnight)

End Time = 14:00:00 (midnight)

The work schedule is configured like this:

- Day 1 (January 3, 2016) is a non-working day.
- Day 2: Working time starting at 22:00:00 and ending at 06:00:00 on day 3.
- Day 3: Working time starting at 22:00:00 and ending at 06:00:00 on day 4.
- Day 4 (January 6, 2016): Working time starting at 22:00:00 and ending at 06:00:00 on day 5.

Result: The rule function returns a working time of 10 hours. That is comprised of start time 22:00 on January 4, 2016 to end time 06:00 on January 5, 2016, and start time 22:00 on January 5, 2016, to end time 00:00 on January 6, 2016.

Example 3: Duration-Based Employee

Now let's look at another example where the time recording variant is set to **`Duration`**.

Here's the data for A.N.Other's working time calculation:

User ID = A.N. Other

Start Date = January 6, 2016

End Date = January 8, 2016

Start Time = 13:00:00

End Time = 14:00:00

The work schedule is configured like this:

- Day 1: Working time duration is 5 hours
- Day 2: Working time duration is 8 hours
- Day 3: Working time duration is 6 hours

Result: The rule function returns a working time of 19 hours. That is comprised of 5 hours on day 1, 8 hours on day 2, and 6 hours on day 3. The Start Time and End Time parameters in the rule function are not considered.

16.133 Has Absences In Period()

This rule function returns whether any active absences exist in a given period or after a given start date. For cross-midnight absences, the absence is considered based on the working day to which the absence belongs.

Input Parameters

For this parameter...	Make this entry:
User ID	User ID for which active absences have to be checked.
Start Date	Enter a start date. Example: Termination date.
End Date (optional)	Enter an end date if you would like to.
Time Types (optional)	Enter one or more time types for which active absences have to be checked. Active absences are absences with status *Pending*, *Pending Cancellation*, and *Approved*. You can only enter time types classified as Absence time types. If you do not enter any time types, all absences will be checked.

Examples

Let's look at some examples.

Example 1: Overlapping Absence

A.N. Other's data looks like this:

- User ID: A.N.Other
- Start Date: November 5, 2014
- End Date: Empty
- Time Type External Code: Empty

Prerequisites: A.N.Other has a work schedule. He works Monday to Friday from 8am to 5pm. Saturday and Sunday are non-working days.

Existing absences for this user:

- October 1, 2014 - October 4, 2014: Time Type - Vacation
- October 31, 2014 - November 5, 2014: Time Type - Vacation

Result: The result is Yes, because the second Employee Time overlaps with the rule function period.

Example 2: No Absence

A.N. Other's data looks like this:

- User ID: A.N.Other
- Start Date: October 5, 2014
- End Date: October 10, 2014
- Time Type External Code: Empty

Prerequisites: A.N.Other has a work schedule. He works Monday to Friday from 8am to 5pm. Saturday and Sunday are non-working days.

Existing absences for this user:

- October 1, 2014 - October 4, 2014: Time Type - Vacation
- October 31, 2014 - November 5, 2014: Time Type - Vacation

Result: The result is No, because no Employee Time overlaps with the rule function period.

Example 3: Cross Midnight with Overlapping Absence

A.N. Other's data looks like this:

- User ID: A.N.Other
- Start Date: July 1, 2020
- End Date: July 31. 2020
- Time Types: Vacation, Sickness

Prerequisites: A.N.Other has a cross-midnight work schedule. He works Monday to Friday from 10pm to 6am. Saturday and Sunday are non-working days.

Existing absences for this user:

- June 15, 2020 - June 30, 2020 (in detail June 15, 2020, 10pm, to July 1, 2020, 6am): Time Type = Sickness
- July 3, 2020 - July 3, 2020 (in detail July 3, 2020, 10pm, to July 4, 2020 6am): Time Type = Special Leave
- July 30, 2020 - August 30, 2020 (in detail July 30, 2020, 10pm to August 31, 2020, 6 am): Time Type = Vacation

Result

- The number of absences is 1.
- The Sickness absence is **not** counted because it does not overlap with the period. Although the absence ends on July 1, 2020 6am, this belongs to the shift starting on June 30, 2020 and is not counted.
- The Vacation absence **is** counted because it overlaps with the period and the time type is part of the time types list.
- The Special Leave is **not** counted because it is not part of the time types list.

16.134 Has Allowances In Period()

This rules function enables you to check whether a user has any allowances within a given period of time.

As such, you need to enter both the user and period of time. If you want, you can search for specific allowance types. The rule function will then check all active time sheets for allowances, including time sheets that have not been approved yet.

> ⓘ Note
>
> The rule function returns a value either confirming or denying the existence of allowances - it doesn't specify the **number** of allowances within the time period. The Get Number of Allowances in Period() [page 261] rule function does that.

Input Parameters

For this parameter	Make this entry
User ID (Required)	The user whose allowances should be checked.
Start Date (Required)	The start of the time period for which you want to check allowances.
End Date (Required)	The end of the time period for which you want to check allowances.

For this parameter	Make this entry
Amendment Scenario (Optional)	The allowances that you want to consider in case of an amendment scenario. If you don't enter a parameter value, all allowances are considered, except for those from inactive, rejected or cancelled time sheets. • *Only consider the amended time sheet.*: This value considers allowances from time sheets with the statuses *Pending* or *Pending Approval* if there is an amendment. This value also considers approvals if there is no amendment. • *Only consider the approved time sheet.*: This value considers only allowances from approved time sheets. • *Consider only allowances that have changed.*: This value lets you find amendments for monthly time sheets that are stored in the *Allowance Recording* object.
Allowance Type (Optional)	The type(s) of allowances you want to check for. If you leave this blank, the rule function will automatically search for all allowance types.

16.135 Has Payouts in Period()

This rule function returns whether any active payouts exist in a given period or after a given start date.

Input Parameters

For This Field	Make This Entry
User ID	Select the *User* field. This is the user for whom payouts have been made in the specified period.
Start Date	Enter a start date.
End Date (optional)	Enter an end date if you would like to.
Time Account Types (optional)	Enter one or more time account types you want to check for payouts made. If you do not enter anything here, all time account types are checked..

Use Case

Termination of an employee should not be allowed if any payouts exist after the termination date - for example, because an administrator has already created them for a future date before the termination action was taken. You can use this rule function for a check that will be assigned as an onSave rule to the Job Information object.

The reference date is the TimeAccountPayout Posting Date.

16.136 Has Permission to Make Retroactive Data Changes

This function checks whether the logged-on user has the permission to make retroactive data changes.

Configuration Requirements

You have the *Admin Permissions* > *Manage User* > *Allow Retroactive Employee Data Changes* permission.

If the permission setting is missing, then the rule function returns the value **No**.

Input Parameters

There are no input parameters.

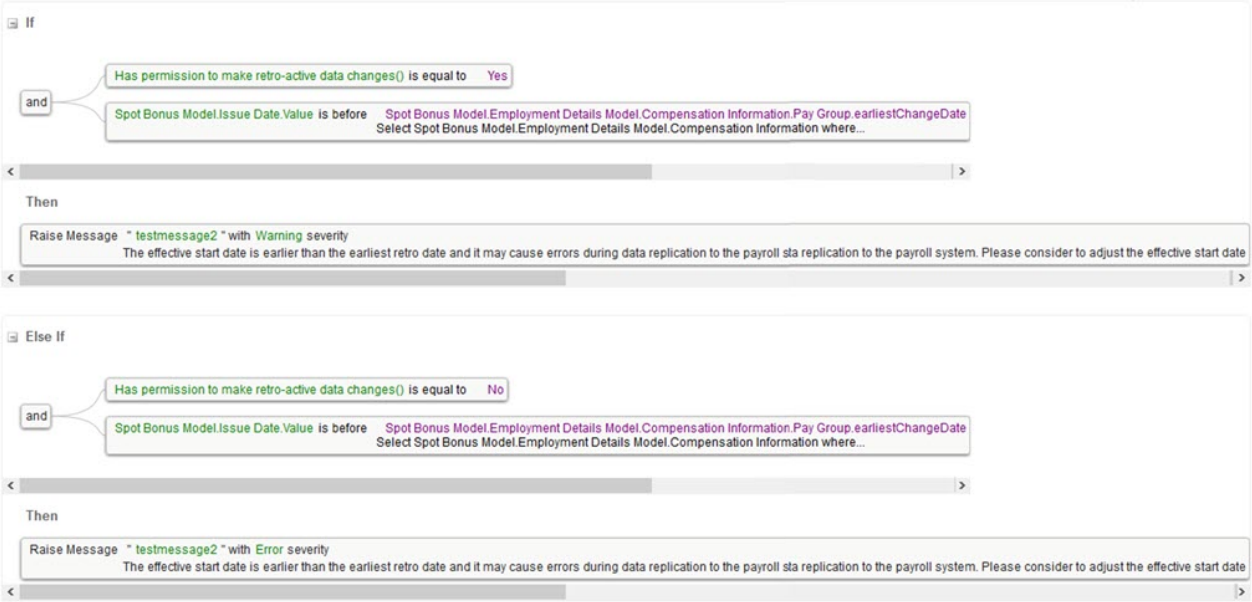
Behavior in Case of Errors

Problem	Possible Cause	What to Do
The function always returns the same values after I have changed the permission configurations.	The permission buffer is not refreshed properly.	Log out of all sessions and log on again. You may need to have the RBP settings refreshed in Provisioning. → Remember As a customer, you don't have access to Provisioning. To complete tasks in Provisioning, contact your implementation partner or Account Executive. For any non-implementation tasks, contact Technical Support.

Problem	Possible Cause	What to Do
The permission setting *Allow Retroactive Employee Data Changes* cannot be found.	The Payroll Integration feature is not activated.	The Payroll Integration feature must be activated in Provisioning. → Remember As a customer, you don't have access to Provisioning. To complete tasks in Provisioning, contact your implementation partner or Account Executive. For any non-implementation tasks, contact Technical Support.
The rule is not executed.	The base object of the rule is not consistent with the Succession Data Model element.	Make sure that the rule is an onSave rule and is assigned to the proper data model element.
	The earliest retro date is not added.	Go to *Manage Data* and review the pay group assigned to the targeted employee in his/her compensation information; then make sure that the field earliest retro changes is updated.
	The earliest retro date is NOT earlier than the date used for the comparison.	Make sure that the date field of the employee data is before the earliest retro date specified.
	The pay group is not included in the employee's Compensation Information.	Assign the employee a pay group found in the Compensation Information and make sure that the effective date is within the effective period of the information to be changed. For example, if the spot bonus issue date is 12/31/2016, the effective start date of the Compensation Information in which the pay group is included must be on or earlier than 12/31/2016.

Use Case

You want to prevent a user from making employee data changes where the effective date is earlier than the earliest retro-active date defined in the employee's pay group assigned in the employee's Compensation Information. If the user does not have permission, the user will get an error message that prevents them from changing their data. And if the user does have permission, then they will still receive an warning message that they are making retroactive changes.



This rule could be set, for example, on the Pay Component Non-Recurring element as an OnSave rule.

16.137 Has Temporary Work Schedule()

This function determines whether an employee has been assigned a temporary work schedule as of a specified date. The answer returned is *Yes* or *No*.

Input Parameters

Here's a table showing the entries you need to make in the input parameters.

For this parameter	Make this entry
User ID	Enter the user for whom you want to make the check.
Date	Enter the date as of which you want to make the check.

16.138 Has Working Time In Interval ()

This function validates whether the employee has any working time interval for the given duration and given time based on work schedule configured for that user.

If the time recording variant is **`Duration`** for any of the dates in question, the function returns an error message, and if the time recording variant is **`Clock Time`**, it will validate whether the user has any working time interval in the time that has passed.

If 00:00:00 is entered as the start or end time, it is defined as the start of the corresponding date.

Input Parameters

For this parameter...	Make this entry:
User	Select the *User* field. This is the user for which you want to calculate the working time in hours.
Start Date	The start date of the duration.
End Date	The end date of the duration.
Start Time	The start time of the duration. **Note** If you want to calculate for full days when using clock times as the time recording variant, you always have to enter a specific time here (for example 00:00:00) and not the **`Employee Time.Start Time`** value. Otherwise, you'll get an error message.
End Time	The end time of the duration. **Note** If you want to calculate for full days when using clock times as the time recording variant, you always have to enter a specific time here (for example 23:59:00) and not the **`Employee Time.End Time`** value. Otherwise, you'll get an error message.

Example 1: Non-Cross-Midnight Work Schedule

Let's look at an example with clock times as the time recording variant and a check over multiple days.

Here's the data for calculating A.N.Other's working time in hours:

User ID = A.N. Other

Start Date = January 4, 2016

End Date = January 6, 2016

Start Time = 13:00:00

End Time = 14:00:00

The work schedule is configured like this:

- Day 1: Working time starting at 08:00:00 and ending at 17:00:00, with an unpaid break between 12:00:00 and 13:00:00
- Day 2: Working time starting at 09:00:00 and ending at 18:00:00, with an unpaid break between 13:00:00 and 14:00:00
- Day 3: Working time starting at 10:00:00 and ending at 18:00:00, with an unpaid break between 11:00:00 and 12:00:00

Result 1: The employee has a working time of 17 hours. That is comprised of 4 hours on day 1, 9 hours on day 2, and 4 hours on day 3. So the result is True.

Example 2: Clock-Time Based Employee With Cross-Midnight Work Schedule

Let's look at an example with clock times as the recording variant and the cross-midnight feature activated.

Here's the data for calculating A.N.Other's working time:

User ID = A.N. Other

Start Date = January 3, 2016

End Date = January 6, 2016

Start Time = 00:00:00 (midnight)

End Time = 00:00:00 (midnight)

The work schedule is configured like this:

- Day 1 (January 3, 2016) is a non-working day.
- Day 2: Working time starting at 22:00:00 and ending at 06:00:00 on day 3.
- Day 3: Working time starting at 22:00:00 and ending at 06:00:00 on day 4.
- Day 4: Working time starting at 22:00:00 and ending at 06:00:00 on day 5.

Result 2: The employee has a working time of 10 hours. That is comprised of start time 22:00 on January 4, 2016, to end time 06:00 on January 5, 2016, and start time 22:00 on January 5, 2016, to end time 00:00 on January 6, 2016.

Example 3: Clock-Time Based Employee With Cross-Midnight Work Schedule

Here's another example.

Here's the data for calculating A.N.Other's working time:

User ID = A.N. Other

Start Date = January 3, 2016

End Date = January 4, 2016

Start Time = 00:00:00 (midnight)

End Time = 00:00:00 (midnight)

The work schedule is configured like this:

- January 2 is a working day with start time 22:00:00 and end time at 06:00:00 next day (January 3).
- January 3 is a non-working day.

Result 3: The employee has a working time of 6 hours. That is comprised of start time 00:00 on January 3, 2016, to end time 06:00 on January 3, 2016.

16.139 Index Of

This function returns the index within this string of the first occurrence of the specified substring, starting at the specified index. If no such substring exists, then null is returned.

Syntax

The syntax looks like this:

indexOf(String text, String searchText [, Integer startIndex])

Input Parameters

This parameter	With this type	And is required?	Means this
text	String	Yes	This is the original text from which the index is obtained.
searchText	String	Yes	Any string.
startIndex	Integer	No	This is the starting index from which to look up the index..

Here are some things you need to know:

- The first index starts from 1, not 0 (zero).
- The start position is not required. If no start position is specified, then the whole text is checked (from start position 1).
- If the start position is negative or zero, take it as the first position.
- If the start position is greater than the whole string size, null is returned.
- If the text is NULL, or the search text is NULL, or the search text is empty, then NULL is returned.

And here are some examples:

This index...	Produces this result
index("123456789", "1", null	1
index("123456789", "1", 2)	Null
index("123456789", "345", 1)	3
index("123456789", "13", 1)	Null
index("123456789", "1", 99)	Null
index("123456789", "\u0034", null)	4

16.140 Initiate PM Form On Job Change Event

Launch a Performance Management form after Job Information changes in Employee Central for Hire, Rehire and, Job Change scenarios.

Configuration Requirements

- The Performance Management module is implemented in the system, and your intended Performance Management form templates are all correctly configured, tested thoroughly, and ready to be launched to end users.
- Intelligent Services is enabled in your system.
- The `pm-form` field is enabled in the *Employee Central* > *Job Information* data model with visibility=VIEW and mandatory=No.
- For the parameter, the name of the pm-form is given in plain text.
- Grant the permission for the `pm-form` field.
- You have used the Trigger onPostSave Events for the Job Information rule scenario and used Job Information Model as the base object, where the If condition is based on values in the Job Information model and the Then condition is Execute Initiate PM Form on Job Change Event. No other Then condition is allowed.

Behavior in Case of Errors

- For future hires, there is a restriction in Performance Management, which means that forms cannot be generated if the Hire date is later than today. To overcome this limitation, an off-cycle batch job needs to be setup and run daily in the system. For more information, refer to the Creating an Off Cycle Event Batch Object topic.
- Business rules cannot be configured to launch forms for terminated employees.
- Only logged-in users who are part of an employee's Form Route Map can see a blue (clickable) link in Job Information. For example, if the roles defined in the steps of the Form Route Map are Employee, Manager, and

HR Manager, only those users will see the blue hyperlink in Job Information. All other users of the system (including system admins) only see the Form Template Label as a string and not a blue hyperlink.

- ⓘ Note

  Currently, this function does not work for systems with Talent Intelligence Hub.

16.141 Is Consecutive Limit for Time Type Reached()

This function returns whether the limit of consecutive days or hours has already been reached considering the current (created, edited) employee time.

Here's a sample use case: the employee should not be allowed to take more than 3 consecutive days of sick leave. The function is able to identify consecutive days also if there are multiple employee times created - for example, one for each day.

This function works for both days and hours. You can simply enter the quantity in the unit that is relevant for your time type.

If you choose “Work Schedule” as a basis, the rule function checks the next and previous working days, based on the work schedule and considering holidays.

⚠ Restriction

The function does not work for flexible requesting.

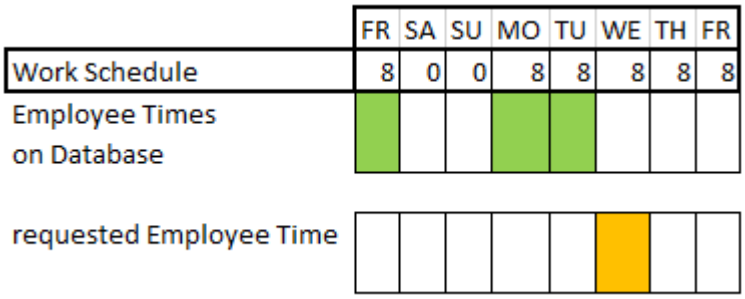
Input Parameters

For this parameter	Make this entry
Employee Time	Enter the employee time created or changed.
Based On	Decide how the consecutive days should be identified. If you choose calendar day, non-working days are considered as well and no holidays are considered. If you choose work schedule, only working days based on work schedule are considered, as well as holidays.
Limit	Enter a limit quantity here in the same unit as the corresponding time type in the employee time. Examples: 3 (days) or 24 (hours).

Details And Examples

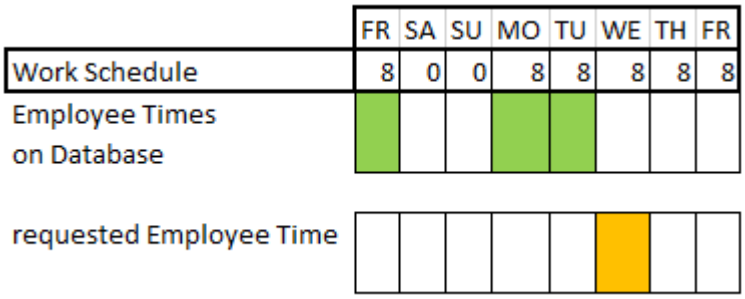
Defining Behavior for Duration-Based Employee Times

- **Example 1A**
  If you request a full-day employee time, the rule function checks all previous working days and sums it up until there is some open booking quantity on that specific day. In addition,, it sums up every subsequent working day until there is some open booking quantity.



  Result> Employee Time creation is not allowed, because 3 days are already booked.

- **Example 1B**
  Maximum 3 consecutive days (based on calendar days). Employee Times on the database are employee times of the same time type as the requested one.



  Employee Time creation is allowed because only 2 days are booked.

16.142 Is Employee Active()

This rule function returns a Boolean value, depending on whether the employee is active or inactive, for the User Id as a mandatory input parameter and the date as an optional input parameter.

By default, the current date is considered. The return value is true for **Active** and false for **Inactive** employee status. The rule function can be consumed in a Take Rule.

It is required for the use case where the employee is inactive and should not be allowed to raise further employee time.

Input Parameters

Here's a table showing the entries you need to make in the input parameters.

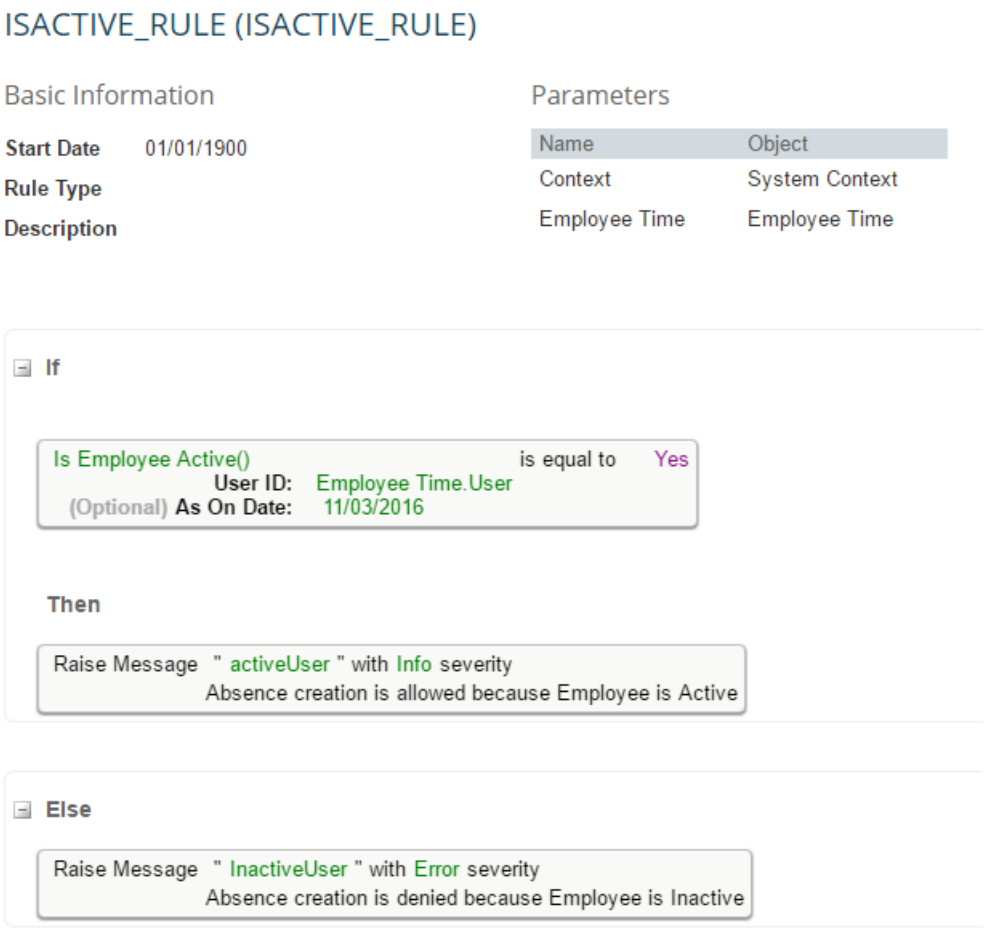
For this parameter:	Make this entry
User Id	User Id for employee whose status is to be fetched.
Date (optional)	Date at which employee status is to be fetched.

Validation of Input Parameters

- The User Id is a required parameter. If it is not provided, an exception is raised and an appropriate error is logged.
- The Date is an optional parameter. If it is not passed, the current date is used to fetch the employee status.

Example

Let's look at an example.



Employee is suspended on September 20, 2016, and the employee will be Active.

The rule function as shown in the screenshot (configured as a take rule) will be executed when the user attempts to create an employee time for this employee. The IF condition in the above rule function will be true and the employee time creation will be allowed.

> **Note**
>
> During new hire and termination if the rule function is called in Job Info save to fetch the status of an employee, it will not return to the right status because the status hasn't been saved yet.

16.143 Is Employee Full Time Worker

This function determines whether the employee is a full-time worker on a specific date. The employee is full-time if FTE in job information has been defined as 1.

Limitation

Use only with Employee Central.

Input Parameters

For this parameter...	Make this entry:
UserId	Select or enter the employee for whom you want to determine full-time employment.
Valid On Date	Select the date that is used for determining if the employee is full-time.

16.144 Is Empty

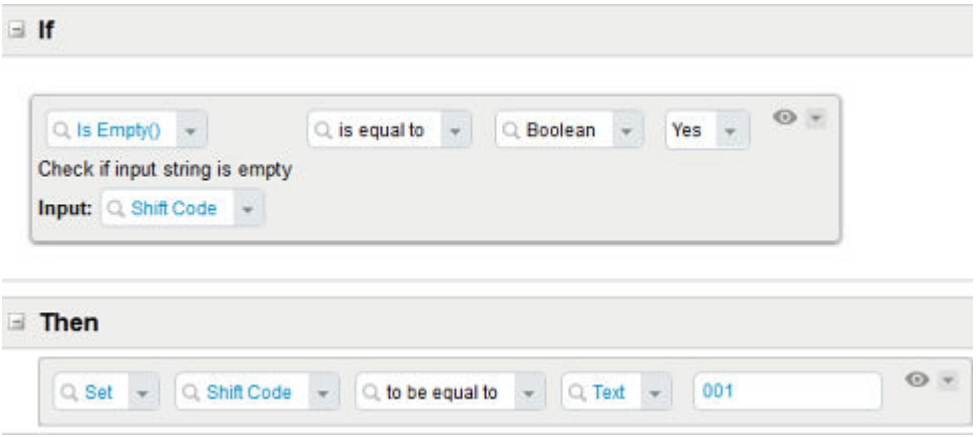
You use this function to check if an input string is empty; for example, when the user has not made an entry in a specific field.

Input Parameters

For this parameter...	Make this entry:
Input	Select a field of field type *Text*. The function returns true if either the parameter Input is "null" or if it contains only spaces.

Use Case

You want to define that if the field *Shift Code* is empty, a default value '001' is assigned to this field. This is what this rule looks like:



16.145 Is Position Below User's Position In Hierarchy

Indicates whether the position is in user's position hierarchy and below user's position.

Limitations

This function is only available if Position Management is activated.

Use Case

You can find a rule example in the Position Management Guide, under *Rule Functions in Position Management*.

16.146 Is Time Management Recalculation Active ()

This rule function is used to validate whether time management recalculation is active or not.

Time Management Configuration is applicable system-wide. This means the recalculation activation applies across all employees and all time types in the system.

The primary use case for this rule function is in absence validation rules (take rules) to control changes in Employee Time if recalculation is not allowed or not applicable for certain dates.

16.147 Is User in a Group

This function indicates whether the user belongs to at least one specified group.

Input Parameters

For this parameter...	Make this entry:

User ID	Enter or select the user ID, for example:
	• Select the *Text* field type, and enter the user ID in the text field.
	• Select a rule that returns the user ID, for example:
	  • *Login User*
	  • *Get Next Available Manager By Position*
	• Select a field that contains the user ID, for example, the *User ID* field from the Job Information.

For this parameter...	Make this entry:
Group Name	Enter or select the name of the corresponding group, for example: • Select the *Text* field type, and enter the corresponding group name in the text field. • Select an attribute that determines to which group the user belongs, for example, *jobTitle*. • The group name you enter in the text field must exactly match the actual group name. Make sure there are no extra spaces at the end and the spelling is the same. → Tip The rule functions check all groups, including permission and workflow groups. If you intend to use this function more than once, we recommend that you enter or select multiple group names the first time using it. Don't run it repeatedly with a different group name each time. The system evaluates successive conditions for Group Name with logical operator **OR**, not AND.

16.148 Is User in Permission Group

This function indicates whether the user belongs to a specified permission group.

Input Parameters

For this parameter...	Make this entry:

User ID	Enter or select the user ID, for example:
	• Select the *Text* field type, and enter the user ID in the text field.
	• Select a rule that returns the user ID, for example:
	  • *Login User*
	  • *Get Next Available Manager By Position*
	• Select a field that contains the user ID, for example, the *User ID* field from the Job Information.

For this parameter...	Make this entry:
Permission Group Name	Enter or select the name of the corresponding group, for example: • Select the *Text* field type, and enter the corresponding group name in the text field. • Select an attribute that determines to which group the user belongs, for example, *jobTitle*. • The permission group name (text field) needs to be **exactly the same** as the name of the group. Make sure there are no extra spaces at the end and the spelling is the same. → Tip The rule functions only checks permission groups. If you intend to use this function more than once, we recommend that you enter or select multiple group names the first time using it. Don't run it repeatedly with a different group name each time. The system evaluates successive conditions for Permission Group Name with logical operator **OR**, not AND.

16.149 Latest Date

This function gets the latest date of two dates. For example, if you have a date in January and one in November 2013, the result returns the November date.

Input Parameters

For this parameter...	Make this entry:
Date	You can: • Select *Date* and enter a date. • Select a date field, for example: *Event Date*.
Date	You can: • Select *Date* and enter a date. • Select a date field, for example: *Event Date*.

The minimum and maximum number of dates for this function is 2.

ⓘ Note

If one of the dates are passed a "Null", then null is returned.

Use Case

You can find a rule example in the Time Off Guide, under *Period-End Processing*.

16.150 Length

This function returns the length of the specified string.

Syntax

The syntax looks like this:

length(String text)

Input Parameters

This parameter	With this type	And is required?	Means this
text	String	Yes	This is the original text from which the index is obtained.

Here are some examples:

This length...	Produces this result
length(null)	0
length(“”)	0
length(“123456789”)	9
length(“\u0031\u0032\u0033\u0034”)	4
length(“123\u00345”)	5

16.151 Login User

This function identifies the logged-in user and returns the user ID.

16.152 Lookup

This function is used to look up values from a Metadata Framework object that serves as a lookup table.

Configuration Requirements

Before you can reference a lookup table in a business rule, you have to create it as a Metadata Framework object in which you define the combination of values.

The status of the Metadata Framework object has to be active; inactive objects are ignored.

> ⓘ Note
>
> In the lookup function, you can only select objects with one-to-one associations. If you need to reference an object with one-to-many associations, create a variable first, and then reference that variable in the lookup function.

Input Parameters

Input Parameter	Type	Required	User Entry
Base object	na	Yes	Select the Metadata Framework object from which you want to gather the data and that serves as your lookup table.

Input Parameter	Type	Required	User Entry
Output field	na	Yes	Select the corresponding field of the Metadata Framework object. After selecting the field, you need to define a collection filter where you define which data records you're searching for. **Note** If the Metadata Framework object is effective-dated, you see a field called *Effective Date*. With this field, you specify which data records should be considered by the lookup function. Here, you can select one of the following options: • *System Context.Effective Date* (Recommended) The effective date of the business process that triggered the rule. For example, if the rule is triggered for an organizational reassignment, the start date of the organizational reassignment. • *Today()* The rule takes the current date. That is, the date of the day on which the rule is run. • *As Specified in Expressions Below* You specify the effective date or period in the expressions

Input Parameter	Type	Required	User Entry
			of the Lookup collection filter. With this option, you can consider data records whose effective date deviates from the effective date of the business process, or from the day on which the rule is run. If you don't explicitly specify a date in the expressions, the system considers data records that exist throughout the whole validity period of the object.

> **Note**
>
> Please note the following when using the Lookup function in variables:
>
> Navigation from the variable to another object mostly doesn't consider the validity of the object returned by the lookup. Instead, it considers only the effective date of the business process that triggered the rule (*System Context.Effective Date*).
>
> Example: You create a variable to get future-dated data records for incumbents. When using this variable in the If or Then sections and navigating to another object (for example, *var_Incumbent.Job Information.Cost Center*), the system considers only the data records that are valid on the effective date of the business process that triggered the rule. Incumbents that are going to be hired in the future are not considered.
>
> This doesn`t apply when you navigate directly to a field of the lookup object (for example, *var_CostCenter.Code*).

Behavior in Case of Errors

If no data record is found, the function returns "null".

For example, if you selected *Today* as effective date for an effective-dated MDF object, but the object is valid only in the future, the function returns "null".

When to Use Lookup Tables

You use lookup tables to define specific combinations of values. Let's have a look at this value combination example in plain language:

If *Seniority From* is 0, and *Seniority To* is 2, then the valid accrual amount is 1.8.

But If *Seniority From* is 3, and *Seniority To* is 10, then the accrual amount is 1.9.

But If ...

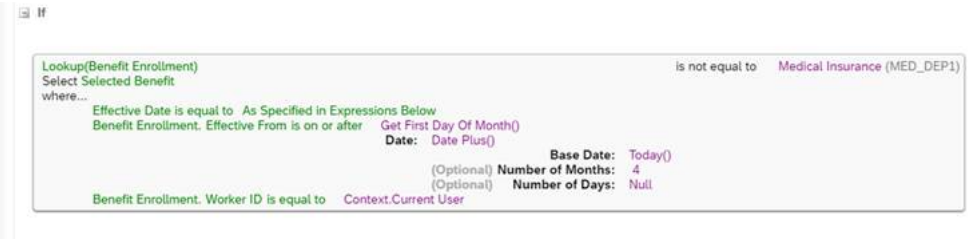
And so on.

In such a case, you can define lookup tables that define the value combinations. Lookup tables can be compared to decision tables as they help the system to understand which value combinations are valid.

> **Note**
>
> You could also define such rules by listing all valid combinations in the rule itself. However, using a lookup table helps you reduce the size and complexity of the rule. In addition, you can reuse the lookup table in several rules.

Use Case

Your employees are eligible to enroll in only one of the following benefits: medical insurance or life insurance. The enrollments are effective in 4 months from today. You create an eligibility rule to check whether employees are already enrolled. The option *As Specified in Expressions Below* allows you to search for future-dated records. Here's an example of what the rule could look like:



Restrictions

- In the lookup function, you can't define a restriction for the same field with the same operator in the Where condition.
  For example:
  - Username is not equal to “Name1”
  - Username is not equal to “Name2”

  In such cases, only the first restriction is taken into account, the subsequent restrictions are ignored.
- It is possible to use the Lookup function during the hire process. However, the system can't access any pending data, such as pending draft data or pending approvals.

Related Information

Creation of a Rule with Lookup Function [page 44]
Accrual Based on Seniority

16.153 Matches

This function checks if a string matches a Java regular expression. For example, you can check if a field contains numbers only, or if it also contains an e-mail address. It returns "true" if the regular expression is matched, and "false" otherwise.

> → Recommendation
>
> As there's no syntax check of the regular expression in the system, we recommend that you test your Java regular expression in an online testing tool first.

Input Parameters

Input Parameter	Type	Required	User Entry
String to be checked	Text	Yes	Select a field containing a text.
Regular expression	Text	Yes	Enter your regular expression.

Behavior in Case of Errors

If one of the input parameters is "null", the function returns "false".

Example

To check if a string only contains numbers, use the regular expression **\d+**.

Related Information

Creating a Rule That Validates the Format of User Entries [page 50]

Examples of Java Regular Expressions [page 52]

16.154 Math Expression

With this function, you can define a custom calculation formula using typical mathematical expressions.

The operators can include the following:

+ - * / ()

...for adding, subtracting, multiplying, dividing, or using brackets in the formula. The operands can be either static number or decimals, or number/decimal fields.

Limitations

- Collection filters cannot be used in *Math Expression*.
- Other functions (like *Round*, *Minimum*, and so on) cannot be used in *Math Expression*.
- Object references that are 1-to-many relationships are not supported.
- When creating the rule, you have to make an entry in the field for defining a *Math Expression* function to be able to see the fields of the business object. The system uses type ahead to display the fields.

If you need to create a rule that is not restricted by these limitations, use the basic calculation functions Add, Minus, Multiply or Divide instead.

Use Case

The system can automatically adjust the value of dependent fields based on new values for key fields. This can be used for:

- Salary calculation based on FTE
  For example, the salary of an employee can be automatically adjusted when the FTE of the employee is changed.
- Vacation days based on service years
- Final rating based on two positions

16.155 Maximum

This function finds the largest value in the list of values you specify. You have to enter at least 2 components.

Input Parameters

Input Parameter	User Entry
Component	You can: • Enter a decimal value. • Select a field, for example: *FTE*.
Component	You can: • Enter a decimal value. • Select a field, for example: *FTE*.

Behavior in Case of Errors

- If all components have the value "null", then "null" is returned.
- If only some of the components have the value "null", then these components are ignored.

16.156 Minimum

This function finds the smallest value in the list of values you specify. You have to enter at least 2 components.

Input Parameters

Input Parameter	User Entry
Component	You can: • Enter a decimal value. • Select a field, for example: *FTE*.

Input Parameter	User Entry
Component	You can: • Enter a decimal value. • Select a field, for example: *FTE*.

Behavior in Case of Errors

- If all components have the value "null", then "null" is returned.
- If only some of the components have the value "null", then these components are ignored.

Related Information

Period-End Processing: General

16.157 Minus

This function subtracts one value from another.

Limitations

You can only use the US format for decimals (using dots, not commas).

Input Parameters

Input Parameter	Type	Required	User Entry
First Value	Decimal	Yes	Enter or select the number from which you want to subtract the second value.
Second Value	Decimal	Yes	Enter or select the number to be subtracted.

Behavior in Case of Errors

If the first or second value is "null", the function returns "null".

> **Note**
>
> Please note that "null" does not mean the number zero in the context of business rules. Instead, it represents a missing value. A field with a null value is a field with no value.

16.158 Modulo

Returns the remainder of a number divided by a divisor.

The syntax looks like this:

modulo(long number, long divisor)

Limitations

- Only non-negative numbers are supported.
- The divisor cannot be zero.

Input Parameters

Name	Type	Mandatory	Description
Number	Decimal	Yes	The number to be divided.
Divisor	Decimal	Yes	The divisor used to divide the number.

Behavior in Case of Errors

- If the number or the divisor is "null", the function returns "null".
- If the divisor is Zero, the rule cannot run and stops with an error.

Example

Modulo	Result
modulo(1, 2)	1
modulo(2, 2)	0
modulo(9, 10)	9
modulo(100, 9)	1
modulo(5918760350, 97)	91

16.159 Month of Year

This function determines the month of the year for a given date.

Input Parameters

For this parameter	Make this entry
Date	The date, such as the start date of an accrual period, whose month you want the system to determine. The default is for the system to use the current date if NULL is passed.

Example

You want to calculate an accrual for December 2023, which means the accrual period runs from December 1, 2023, to December 31, 2023. You can then use this rule function to enable the system to recognize that the month included in the start date for the accrual is December.

To do this, you would make the following entries:

- *Date*: Choose Accrual Start Date from the dropdown.

Result: 12

16.160 Multiply

This function multiplies values.

Limitations

You can only use the US format for decimals (using dots, not commas).

Input Parameters

Input Parameter	Type	Required	User Entry
Value	Decimal	Yes	Enter or select the number you want to be multiplied.
Factor	Decimal	Yes	Enter or select the multiplication factor.

Behavior in Case of Errors

If the value or factor is "null", the function returns "null".

> ⓘ Note
>
> Please note that "null" does not mean the number zero in the context of business rules. Instead, it represents a missing value. A field with a null value is a field with no value.

16.161 Next Pay Level from Pay Scale Structure

Gets the **next** *Pay Scale Level* from the pay scale structure based on the pay scale level and effective date. The pay scale structure is configured in the MDF object *Pay Scale Level*.

Limitations

Use only if the fields Pay Scale Group and Pay Scale Level are set up in Job Information

Parameters

For this parameter...	Of type...	Which is...	Make this entry...
Pay Scale Level	Text	Required	Select the pay scale level you want to use. This is usually the pay scale level value that is entered in the job information.
Effective Date	Date	Required	Enter or select the effective date you want to use to read the data from the pay scale structure. Usually this is the event date of the job information.

16.162 Number/Percentage/Rate/Unit from Pay Scale Structure

Gets the *Number/Percentage/Rate/Unit* from the pay scale structure based on the pay scale level, pay component, and effective date. The pay scale structure is configured in the MDF object *Pay Scale Level*.

Limitations

Use only if the fields *Pay Scale Group* and *Pay Scale Level* are set up in Job Information.

Parameters

For this parameter...	Of type...	Which is...	Make this entry...
Pay Scale Level	Text	Required	Select the pay scale level you want to use. This is usually the pay scale level value that is entered in the job information.

For this parameter...	Of type...	Which is...	Make this entry...
Pay Component	Value	Required	Select field type **Value**, and select the pay component you want to search for in the pay scale structure. Usually this is the pay component you want to create in the compensation model.
Effective Date	Date	Required	Enter or select the effective date you want to use to read the data from the pay scale structure. Usually this is the event date of the job information.

16.163 Opposite Sign

This function gets a value with the opposite sign. For example, if the value is 5, the result of this function is -5.

Input Parameters

For this parameter...	Make this entry:
Value	You can: • Select *Decimal* and enter a value. • Select a field, for example: *FTE*.

Use Case

You can find a rule example in the Time Off Guide, under *Period-End Processing*.

16.164 Pay Range from Job Information

The pay range associated to a Job Information record is determined on-the-fly when calculating the compa-ratio and range penetration in Compensation Information and is usually not visible to the user. This rule function can be used to make this information visible. If the rule function uses Job Information as the base object, then the effective start date is taken from that Job Information record. If the rule function uses Compensation Information as the

base object, then the effective start date of the relevant Job Information record (the one that is valid/effective when compared to the start date of the Compensation Information record being created) is taken.

Limitations

Use only if pay ranges are set up.

Input Parameters

For this parameter...	Of type...	Which is...	Make this entry:
Job Information	Job Information	required	a Job Information record

If no associated pay range is found or if errors occur during the determination of the pay range, the function returns nothing.

Use Case

You can create a rule that displays the associated pay range when the relevant fields of the Job Information are changed.

Related Information

Showing the Pay Range in Job Information

16.165 Random

Returns a random number between zero (0) and 1.

Limitations

- The number can be zero, but not 1.
- The number is accurate to 20 decimal places.

16.166 Replace

This function searches in the original text for all occurrences of a specified pattern, replaces them with the specified replacement, and returns a result with the replacement made. The specified pattern can either be simple text or a Java regular expression.

> → Recommendation
>
> As there's no syntax check of the regular expression in the system, we recommend that you test your Java regular expression in an online testing tool first.

Input Parameters

Input Parameter	Type	Required?	User Entry
Text	String	Yes	Enter the original text where search and replace takes place. You can either: • Enter the text. • Enter a field that contains the text.
Pattern	String	Yes	Enter the pattern you want to search for. You can either: • Enter a text. • Enter a Java regular expression. • Select a field that contains the text or a Java regular expression

Input Parameter	Type	Required?	User Entry
Replacement	String	No	Enter the replacement text. This text replaces all sub-strings found in the original text that match the pattern. You can either: • Enter the replacement text. • Select a field that contains the replacement text. • Leave the field empty to remove all found sub-strings without any replacement. You achieve the same behavior by selecting *Null* in this field.

Behavior in Case of Errors

- If the (original) text is "null" or the pattern is "null", the function returns "null".
- If the pattern is an empty string, the original text is returned unchanged.
- If the pattern is an invalid Java regular expression, the rule stops with status `FAIL`.

> **Note**
>
> Please note that "null" does not mean the number zero in the context of business rules. Instead, it represents a missing value. A field with a null value is a field with no value.

Specific behavior:

- If two overlapping sub-strings match the pattern, only the first occurrence is replaced.
- If the pattern is not found, the original text is returned unchanged.

Examples

When you select this...	You get this result	Comments
Replace Text: "ababa" Pattern: "bab" Replacement: "c"	"aca"	The function searches for the text "bab" and replaces it with "c"

When you select this...	You get this result	Comments
Replace Text: “ababa” Pattern: “aba” Replacement: “c"	"cba"	The pattern "aba" is found twice in the original text "ababa". However, as these found places are overlapping, only the first occurrence is replaced.
Replace Text “ababa” Pattern: “ab” Replacement: “c”	"cca"	The pattern "ab" is found twice in the original text "ababa". Both occurrences are replaced.
Replace Text: “ababa” Pattern: “bab” Replacement: null	"aa"	Specifying `null` as replacement results in removing the pattern found.
Replace Text “ab1” Pattern: “\u0031” Replacement: “\u0063”	"abc"	In this example, you use unicode characters: • u0031 is unicode for the number one • u0063 is unicode for latin small character "c" Please note that you have to put a backslash prefix before the unicode character.
Replace Text: “ab\\c” Pattern: “\\\\” Replacement: null	"abc"	In Java, you have to escape metacharacters like a backslash by a preceding backslash. That is why you end up with 4 backslashes for the pattern to replace the 2 backslashes in the original text.
Replace Text: “abc1234d” Pattern: “\\d” Replacement: null	"abcd"	In this example, you use `\\d` as a Java regular expression to match digits (0 to 9).
Replace Text: “abc” Pattern: “d” Replacement: “e”	"abc"	The pattern "d" is not found. The original text is returned unchanged.

> **Note**
>
> You can find more information about unicode characters and Java regular expressions on the Internet.

Related Information

Examples of Java Regular Expressions [page 52]

16.167 Round

With this function, you can define how values are rounded by defining the degree of accuracy and as of which number a value is rounded.

Input Parameters

Input Parameter	Type	Required	User Entry
Value	Decimal	Yes	You can: • Select *Decimal* and enter a value. • Select a field, for example: *FTE*.
Precision	Number	No	Select *Number* and enter a number that defines as of which pre-decimal point position or decimal place the rounding is applied. This way you define the degree of accuracy for the rounding. For example, if the value is 123.456, the precision is 1 (=one decimal place), the result is 123.5 (with a threshold of 0.05). If the precision is 2 (=two decimal places), the result is 123.46. If the precision is -1 (=1 pre-decimal point position), the result is 120. If you leave this field empty or select *Null*, the function uses the number zero.

Input Parameter	Type	Required	User Entry
Threshold	Decimal	No	Select *Decimal* and enter a rounding threshold. The threshold defines as of which number the system rounds up. If you don't enter a value, the threshold is selected by default as 0.05, 0.5, 5, and so on (depending on the precision value). If you want to use other thresholds, enter the corresponding number. For example, if the value is 21.5, and you have selected a threshold of 0.6, the result is 21.

Behavior in Case of Errors

If *Value* is "null", the result of the function is "null".

> ⓘ Note
>
> Please note that "null" doesn't mean the number zero in the context of business rules. Instead, it represents a missing value. A field with a null value is a field with no value.

Example

Here's an example of how the rounding is applied:

Value	Precision	Threshold	Result
50,782.765	2	0.005	50,782.77
50,782.765	1	0.05	50,782.80
50,782.765	0	0.5	50,783
50,782.765	-1	5	50,780
50,782.765	-2	50	50,800
50,782.765	-3	500	51,000
50,782.765	-4	5000	50,000

Examples

Here's how you can round a value down to the nearest 0.5:

Sample Code

```
Divide()
Dividend: Round()
     Value: Multiply()
           Value: valueToBeRounded
           Factor: 2
     Precision: 0
     Threshold: 1
Divisor: 2
```

Note

To round up to the nearest 0.5, simply set the threshold to 0.

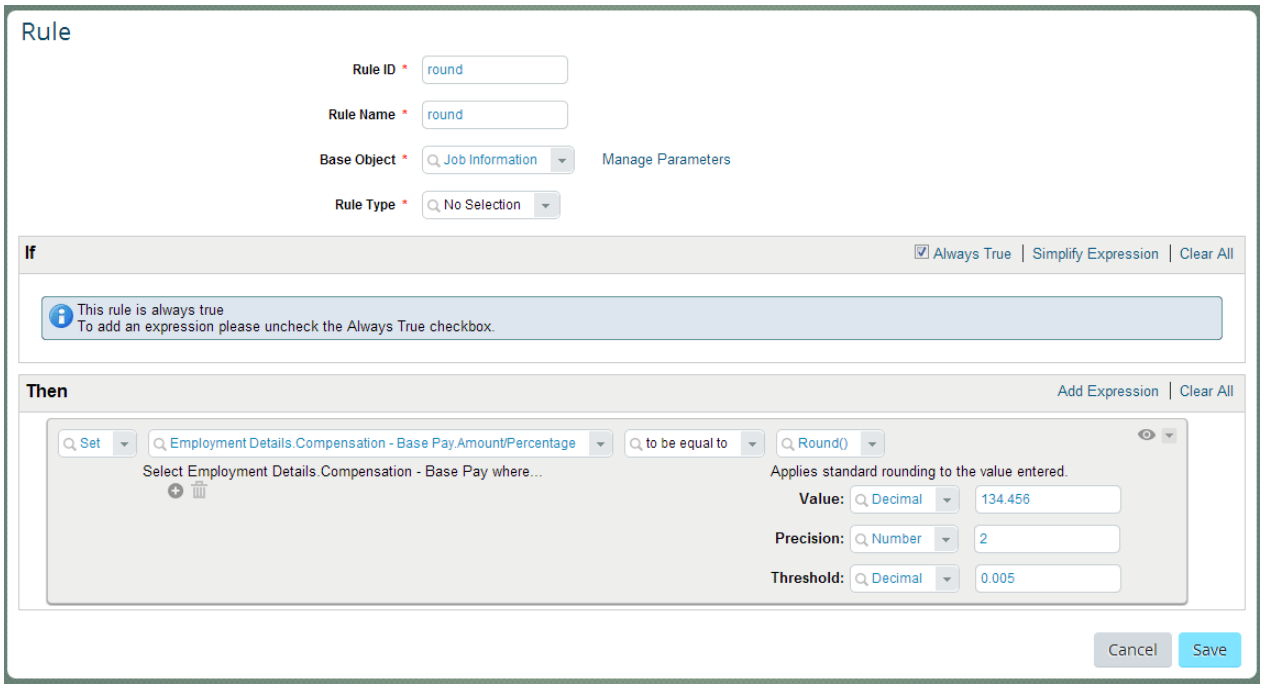
And here's how you can round a value up to the nearest 0.25:

Sample Code

```
Divide()
Dividend: Round()
     Value: Multiply()
           Value: valueToBeRounded
           Factor: 4
     Precision: 0
     Threshold: 0
Divisor: 4
```

Use Case

In this example, the base pay is rounded, returning a result of 134.46:



16.168 Substring

This function returns a new string that is a substring of the given string. The substring begins at the specified start index and extends to the character at index start+length-1.

Syntax

The syntax looks like this:

substring(String text, Integer start [, Integer length])

Input Parameters

This parameter	With this type	And is required?	Means this
text	String	Yes	This is the original text from which to get the substring. It can be any form of text.
Start	Integer	Yes	This is the beginning index. It is inclusive and starts from 1.
length	Integer	No	This is the length of the sub-string, from the starting index.

Here are some things you need to know:

- The first index starts from 1, not 0 (zero).
- If the start position is negative or zero, take it as the first position.
- If the start position is greater than the whole length, the function returns NULL.
- If the length is zero or negative, NULL is returned.
- If the text is NULL or the start position is NULL, then NULL is returned.
- If the length exceeds the length of the text, then the text up to the end is used.
- You can enter the length, but you don't have to. If you don't, it means extending to the last index.

And here are some examples:

This substring...	Produces this result
substring("123456789", 1, 4)	"1234"
substring("123456789", 2, 7)	"2345678"
substring("123456789", -1, 4)	"1234"
substring("123456789", 1, null)	"123456789"
substring("123456789", 1, 99)	"123456789"
substring("123456789", 2, 0)	Null
substring("123456789", 4, -5)	Null
substring("123456789", 100, 10)	Null
substring("123456789", 8, 5)	"89"
substring("\u0061b", 1, 10)	"ab"

16.169 Sum of Collection Field Values

This function sums up the values of a collection field that meet the conditions.

Input Parameters

Input Parameter	Type	Required	User Entry
Collection	Collection	Yes	Select the collection that contains the field whose values you want to sum up. • The collection needs to be a Metadata Framework object. • You can only select first-level collections. If you want to use collections that are below another collection, you need to use variables. • Optional: You can use a collection filter ("*select...where*").
Field in Collection	Decimal	Yes	Select the collection field relevant for calculation.

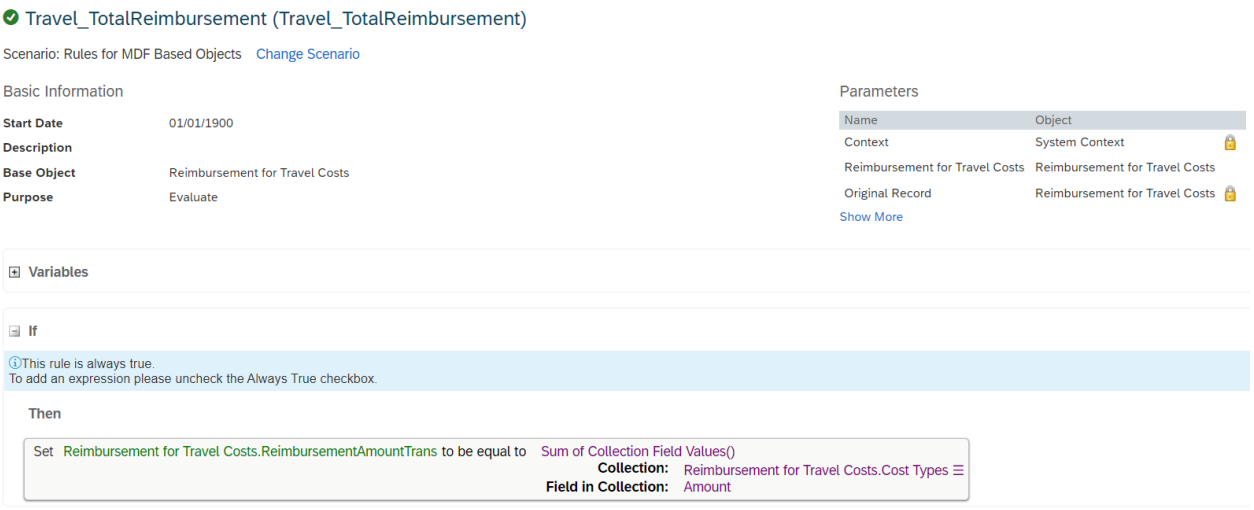
Behavior in Case of Errors

If *Collection* is "null" or empty, the system returns the number 0.

If the field value of a data record in the collection is "null", the system uses the number 0 for this data record when calculating the overall sum.

Use Case

You want to see the total amount of travel costs. You've created a custom parent object called "Reimbursement for Travel Costs" that has a child object called "Cost Types" that contains several child records, for example, for hotel and flight costs. You create a rule that uses the function Sum of Collection Field Values to get the overall amount of all travel costs on the parent object.



16.170 Timestamp Current Time UTC plus Offset Minutes

You use this function to create a timestamp of the current time in the time zone format UTC (Coordinated Universal Time). You can also shift the timestamp into the future, by defining how much time in minutes should be added to the current time. The result is a string that is organized according to ISO 8601, combining date and time in UTC. For example: 2014-02-19T15:42:00.000Z, where Z stands for time zone UTC.

Input Parameters

For this parameter...	Make this entry:
Minutes	Select the field type *Number* and enter the number of minutes that is added to the current time in UTC; for example: **`60`** . If you enter **`0`** , the timestamp is calculated as the current time in UTC.

Use Case

A customer that uses payroll integration with ERP can create rules to reschedule the replication of employee data according to the customer's specific needs. You can find a rule example in the SuccessFactors Employee Central Payroll Integration Guide, under *Rescheduling Employee Data Replication*.

16.171 Today

This function gets the current date in the tenant preferred time zone as defined in the *Admin Center* > *Platform Feature Settings*.

Refer to *Tenant Preferred Time Zone* given in the Related Information section.

> ⚠ Caution
>
> The rule function can produce unexpected results if the time zone the user is in differs from the tenant preferred time zone.

Related Information

Tenant Preferred Time Zone

16.172 To Lowercase

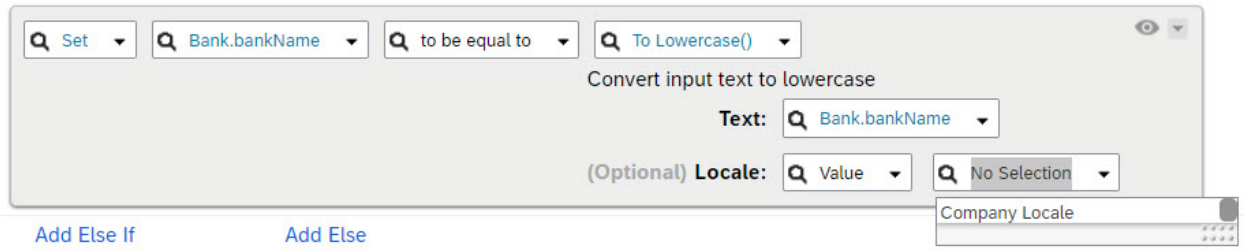
This function converts all of the characters in given text to lower case.

Input Parameters

For this parameter...	Make this entry:
text	Select a field of field type *Text*.
Locale	Select a field of field type *Value*. You can optionally select a different locale. The default is the system locale, but you can change this to the *Company* or *User* locale.

Use Case

You want to change the case of the field Bank Name to lowercase when you save your entries. This may be useful for IBAN or BIC/SWIFT entries.



16.173 To Number

This function converts a text into a number. This is useful when you want to compare numbers, and one of the numbers is stored in a text field.

Input Parameters

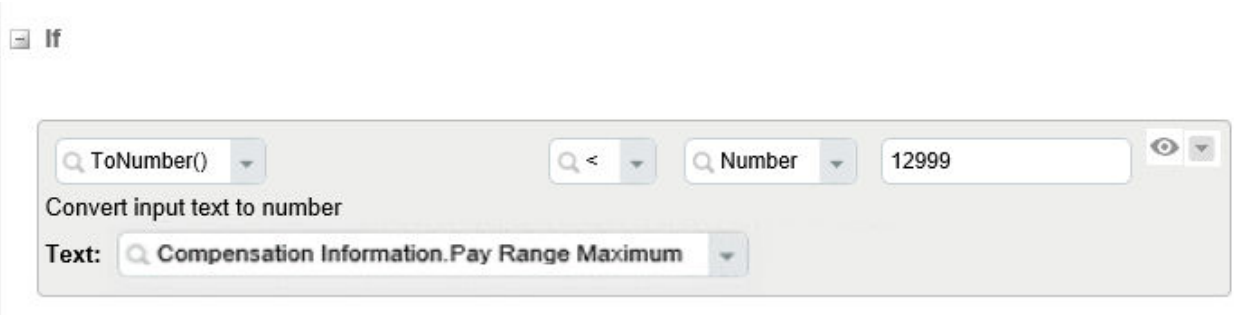
Input Parameter	Type	Required	User Entry
Text	Text	Yes	Select an object that contains numbers in text format. For example: Pay Range Maximum of Compensation Information. Both cardinal numbers and numbers with decimal points are supported; leading zeros are ignored. For example: • 12996 • 129.96 • 012996 (is interpreted by the function as 12996)

Behavior in Case of Errors

- If the object in the *Text* field returns an empty string or "null", the function returns "null".
- If the object in the *Text* field contains other characters than cardinal or decimal numbers (for example, letters, commas, hexadecimal numbers, or spaces), the rule stops being executed.

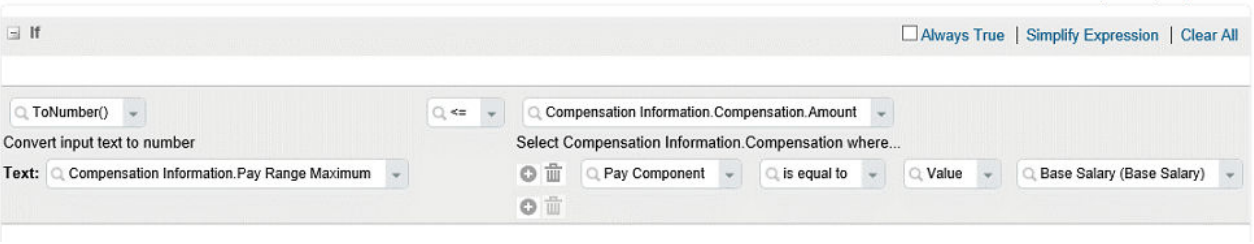
Example: Comparing with Numbers

You use the To Number function to convert the number contained in the *Pay Range Maximum* text field into a number, and then compare it with the number 12999. Here's an example of what that could look like:



Example: Comparing with Number Fields

You want to convert the number contained in the *Pay Range Maximum* text field into a number, and then compare it with the amount of a *Pay Component*. Here's an example of what that could look like:



16.174 To Uppercase

This function converts all of the characters in given text to upper case.

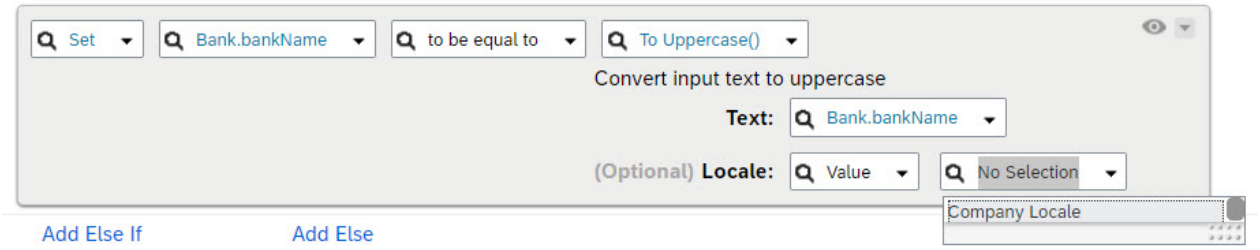
Input Parameters

For this parameter...	Make this entry:
Text	Select a field of field type *Text*.

For this parameter...	Make this entry:
Locale	Select a field of field type *Value*. You can optionally select a different locale. The default is the system locale, but you can change this to the *Company* or *User* locale.

Use Case

You want to change the case of the field Bank Name to lowercase when you save your entries. This may be useful for IBAN or BIC/SWIFT entries.



16.175 Treat Null As

You use this function to define that if a number field is "null", it should be treated as a specific value (usually the number zero). For example, when the user saves the job info without entering anything in the FTE (full-time equivalent) field, you can use this function to define that this empty value is considered by the system as if it has been filled by the user with the number zero.

Input Parameters

Input Parameter	Type	Required	User Entry
Number	Decimal	Yes	Enter or select a number of type *Decimal*.
Treated as Number	Decimal	No	Enter or select a number of type *Decimal*.

Behavior in Case of Errors

Although the parameter *Treated as Number* is not mandatory, you should always specify at least a value "<>" here. If you do not specify a value, or if the specified value is null, then the function will return null if the parameter number is null. In which case you could also omit the function entirely, as it does not have any effect.

Use Case

You want to define that if the FTE field is not filled, the system treats this field as if it contains the number zero. When FTE is zero, the vacation days are set to 12. Here's the rule:

If

Treat Null As(FTE, 0) == 0

Then

Set Vacation days = 12

16.176 Trigger Email Notification for Onboarding Processes

This function is used to trigger email notifications for Onboarding process steps that aren't supported by preconfigured email notifications. Examples of Onboarding processes for which you can use this rule function are new hire and rehire data review tasks, manage pending hires, personal data collection, and custom data collection.

Configuration Requirements

Onboarding is enabled.

Input Parameters

Input Parameter	Type	Required	User Entry
Email Trigger	Value	Yes	Select the appropriate email trigger.
Data Source	Value	Yes	Select the source from which the User IDs of recipients and the email content is retrieved before sending an email notification.

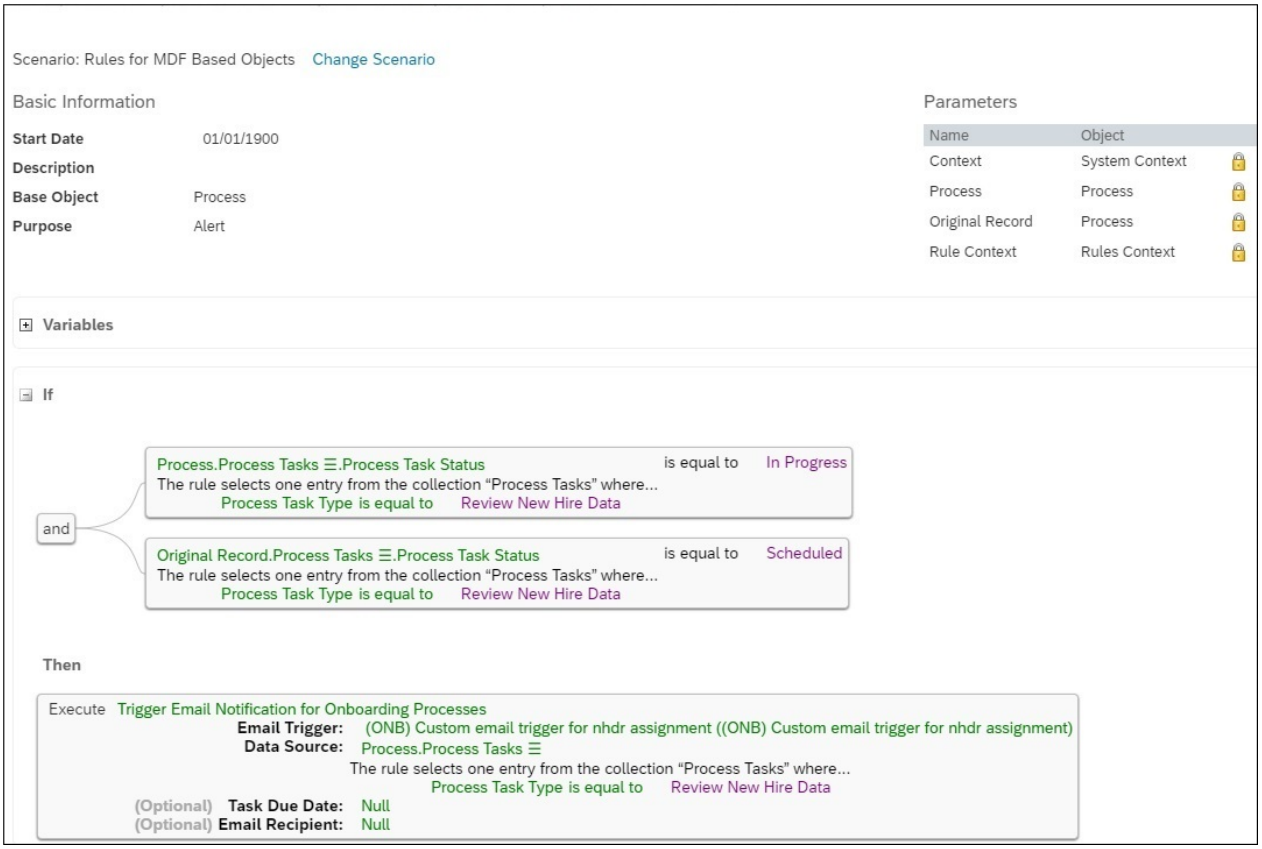
Input Parameter	Type	Required	User Entry
Task Due Date	Date	No	Set the due date of the task based on which the reminder email must be triggered.
Email Recipient	Value	No	Depending on the recipient builder used in the email template, specify the User IDs of the recipients of the email. • If *Onboarding User ID Recipient Builder* is used in the email template, specify the User IDs of the recipients. • For recipient builders other than the *Onboarding User ID Recipient Builder*, leave this field blank. Email notifications are sent only to the recipients listed in the email template.

Behavior in Case of Errors

In case of errors, email notifications fail to trigger. However, these failures don't impact the completion of the in-progress transaction in Onboarding.

Example

The following is an example of defining the If-Then conditions for the function to trigger an email notification when the New Hire Data Review step is in the *In Progress* status.



16.177 Trigger Event for Restarting Onboarding Process

This function restarts the onboarding process automatically as soon as the start date or the hiring manager of a new hire are changed. It then restarts the process by canceling the old process and creating a new one. Also, it sends a cancellation email to the hiring manager.

In detail, the function checks whether the fields for the start date or the hiring manager have been changed when the ONB2Process MDF object of the applicant is saved. The function parameters used are instances of the same object before and after the changes in the ONB2Process are saved.

Please note that you can define your customer-specific configuration in the IF condition of the rule. In the THEN part, you need to select the *Trigger Event for Restarting Onboarding 2.0 Process()* function from the list of functions, and enter the corresponding input parameters.

Limitations

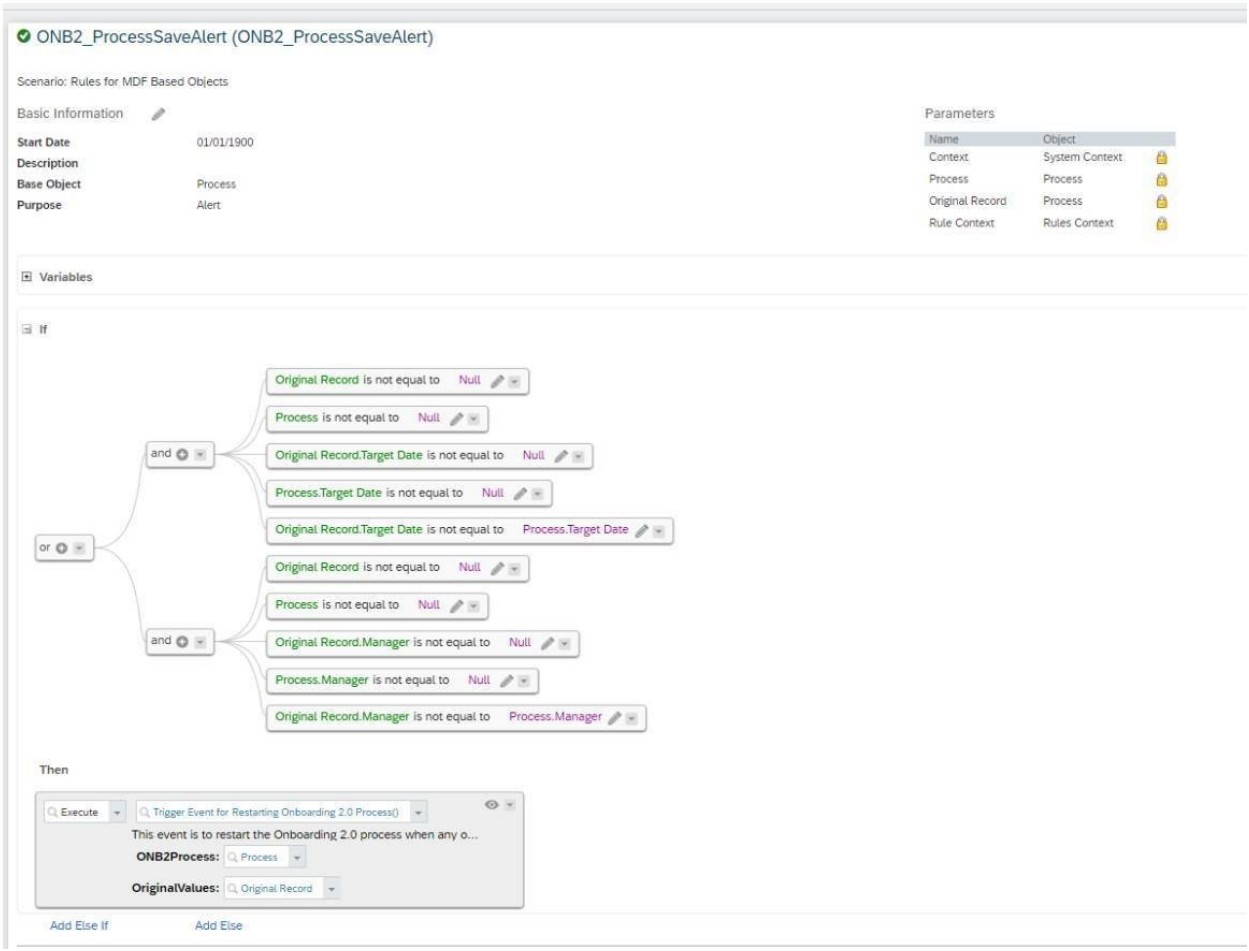
Use only with Onboarding enabled.

Input Parameters

For this parameter...	Of type...	Which is...	Make this entry:
ONB2Process	MDF object	Required	*Process*
Original Values	MDF object	Required	*Original Record*

Use Case

Here's an example showing how you can define the IF conditions so that the function triggers the desired behavior.



16.178 Trigger Worker Absence Event

This function is used to publish the Absence event.

Input Parameters

This parameter	With this type	And is required?	Means this
EmployeeTime	GO	Yes	This is the EmployeeTime record we will use to publish the event.

16.179 Trigger Worker Long Term Disability Event

This function is used to publish the longTermDisabilityEvent.

Input Parameters

This parameter	With this type	And is required?	Means this
EmployeeTime	GO	Yes	This is the EmployeeTime record we will use to publish the event.

16.180 Trigger Worker Short Term Disability Event

This function is used to publish the shortTermDisabilityEvent.

Input Parameters

This parameter	With this type	And is required?	Means this
EmployeeTime	GO	Yes	This is the EmployeeTime record we will use to publish the event.

16.181 Trim

With this function, you can remove whitespaces from the start or end of a text string.

Sometimes users might enter whitespaces (for example, blank spaces or tabs) unintentionally when creating a text string. This can cause problems when saving the input text. For example, you might end up with duplicate entries called “Role Name” and “Role Name <with unintentional whitespace at the end>”.

Input Parameters

Input Parameter	Type	Required	User Entry
Text	Text	Yes	The text string you want to trim.
Trim Mode	Value	No	One of the following options: • *Remove leading whitespaces* • *Remove trailing whitespaces* • *Remove leading and trailing whitespaces* If you leave this field empty or select *Null*, the function uses the last option as default (*Remove leading and trailing whitespaces*).

Behavior in Case of Errors

If *Text* is "null", the result of the function is "null".

> **Note**
>
> Please note that "null" doesn't mean the number zero in the context of business rules. Instead, it represents a missing value. A field with a null value is a field with no value.

Example 1:

Original text string: "<leading whitespace>Role Name"

Text string after running the Trim function: "Role Name"

Example 2:

Original text string: "Role Name<trailing whitespace>"

Text string after running the Trim function: "Role Name"

16.182 Validate Higher Duty/Temporary Assignment Compensation Information

This function validates the compensation information and ensures that any change in the pay group of the nominal assignment is also updated in the compensation information of the currently active and future-dated higher duty or temporary assignments. You can also ensure that only a pay-component recurring of type Estimated HD allowance is present in the compensation information.

Configuration Requirements

You need to have higher duty enabled.

Input Parameters

Input Parameters for Validate Higher Duty/Temporary Assignment Compensation Information

For this parameter...	Of this type...	Which is...	Make this entry
Compensation Information	Text	Required	The compensation details of the employee who also holds a higher duty or temporary assignment position.

For this parameter...	Of this type...	Which is...	Make this entry
Employment Information	Text	Required	The employment details of the employee who also holds a higher duty or temporary assignment position.
Higher Duty/Temporary Assignment	Text	Required	The details of the higher duty or temporary assignment position.

Behavior in Case of Errors

If the nominal employment has a higher duty assignment, then if any change in the pay group of the nominal assignment is not also updated in the compensation information of the currently active and future-dated higher duty or temporary assignments, a warning occurs. To prevent data inconsistencies, you must ensure that any change in the pay group of the nominal assignment is also updated in the compensation information of the currently active and future-dated higher duty or temporary assignments.

If you make any changes to the pay group information in the currently active and future-dated higher duty or temporary assignments, then an error occurs. You must first update the pay group information in the nominal assignment.

Use Case

When a higher duty or temporary assignment is created, it is associated closely with the corresponding nominal assignment of the same employee.

Using this rule function, you can ensure that any change in the pay group of the nominal assignment is also updated in the compensation information of the currently active and future-dated higher duty or temporary assignments. You can also ensure that only a pay-component recurring of type Estimated HD allowance can be present in the compensation information.

If this rule function is not configured, it can lead to data inconsistencies for the higher duty or temporary assignment.

Related Information

Define Rule to Validate Compensation Information

16.183 Validate Higher Duty/Temporary Assignment Employment Information

This function validates the employment information and ensures that the higher duty assignment duration is within the boundary of the nominal employment dates, that only one higher duty exists for a nominal employment, and that there are no overlapping higher duty assignments.

Configuration Requirements

You need to have higher duty enabled.

Input Parameters

Input Parameters for Validate Higher Duty/Temporary Assignment Employment Information

For this parameter...	Of this type...	Which is...	Make this entry
Employment Information	Text	Required	The employment details of the employee who also holds a higher duty or temporary assignment position.
Higher Duty/Temporary Assignment	Text	Required	The details of the higher duty or temporary assignment position.

Behavior in Case of Errors

If the nominal employment has a corresponding higher duty assignment, then if you update the employment information of the nominal assignment, you get an error if the higher duty assignment duration is not within the boundary of the nominal employment dates.

If you try to update the higher duty assignment duration to a duration that is outside the boundary of the corresponding nominal assignment, then an error occurs and the updated duration is not saved. You can modify the higher duty assignment duration to be within that of the corresponding nominal assignment. Alternatively, to make changes to both the nominal assignment and higher duty assignment durations, you can modify the nominal assignment duration first and then update the higher duty assignment duration to be within the nominal assignment duration.

Use Case

When a higher duty or temporary assignment is created, it is associated closely with the corresponding nominal employment assignment of the same employee.

The rule functions allow you to trigger the following validations:

- The higher duty or temporary assignment can be active only in the duration the nominal employment is active.
- If or when the nominal employment is terminated, the corresponding higher duty or temporary assignment must also be terminated.
- Only **one** higher duty or temporary assignment can be active for a nominal employment at any given point in time.
- Any changes to the employment information of a nominal employment or higher duty changes are validated against the start and end dates of the nominal employment assignments.

If this rule function is not configured, it can lead to data inconsistencies for the higher duty or temporary assignment.

Related Information

Define Rule to Validate Employment Information

16.184 Validate Higher Duty/Temporary Assignment Job Information

This function validates the job information and ensures that the company, country and employee class of the higher duty or temporary assignment is same as that of the corresponding nominal assignment.

Configuration Requirements

You need to have higher duty enabled.

Input Parameters

Input Parameters for Validate Higher Duty/Temporary Assignment Job Information

For this parameter...	Of this type...	Which is...	Make this entry
Job Information	Text	Required	The company, country, and employee class details of the employee who also holds a higher duty or temporary assignment position.
Employment Information	Text	Required	The employment details of the employee who also holds a higher duty or temporary assignment position.
Higher Duty/Temporary Assignment	Text	Required	The details of the higher duty or temporary assignment position.

Behavior in Case of Errors

If the nominal employment has a higher duty assignment, then if the company, country and employee class of the higher duty or temporary assignment are not the same as that of the corresponding nominal assignment, then a warning occurs when you try to update the nominal assignment. It is recommended that you update the job information in the higher duty or temporary assignment accordingly.

If you make changes to the company, country and employee class of the higher duty or temporary assignment, an error occurs if the modified information does not match the information in the nominal assignment, and the job

information is not updated. To ensure that the changes match the information in the nominal assignment, you must first update the job information in the nominal assignment.

Use Case

When a higher duty or temporary assignment is created, it is associated closely with the corresponding nominal employment assignment of the same employee.

This rule function validates that the company, country and employee class of the higher duty or temporary assignment is same as that of the corresponding nominal assignment.

Using this rule function, you can ensure that any change to the company or country is not allowed if an active or future-dated higher duty assignment exists for the employee.

If this rule function is not configured, it can lead to data inconsistencies for the higher duty or temporary assignment.

Related Information

Define Rule to Validate Job Information

16.185 Week Of Year ISO

This function determines the number of the week for a given date as defined by the ISO standard. The ISO standard defines week 1 as the week containing the first Thursday of January.

Example

Week of Year ISO: Jan 10, 2016

Result: 1 (= calendar week 1)

Related Information

Week of Year US [page 387]

16.186 Week of Year US

This function determines the number of the week for a given date as defined by the US calendar. The US calendar defines week 1 as the week containing January 01.

Example

Date: Jan 10, 2016

Result: 2 (= calendar week 2)

Related Information

Week Of Year ISO [page 386]

16.187 Year Of Date

This function determines the year of a given date.

The default is for the current date to be used if NULL is passed as a parameter.

Example

Date: June 10, 2021

Result: 2021

17 Change History

Learn about changes to the documentation for Implementing Business Rules in SAP SuccessFactors in recent releases.

1H 2025

Type of Change	Description	More Info
Changed	New Authorization Concept	Permissions for Business Rules [page 14]
New	You can use this function to set a value in the Job Location field on job requisitions, based on the Job Location generic object.	Get Job Location() [page 238]
Added	We added a note saying that "Initiate PM Form On Job Change Event" is currently not working for systems with Talent Intelligence Hub.	Initiate PM Form On Job Change Event [page 338]
New	We added information about the new **PermissionRolesHaveConsistentAuthorizationsForRules** check that you can use to validate if all permission roles have consistent authorizations for working with business rules.	Do All Permission Roles have Consistent Authorizations for Working with Business Rules? [page 88]

2H 2024

Type of Change	Description	More Info
Changed	The existing "Is User in Permission Group" rule function is now renamed "Is User in a Group". The parameter name for this rule function has also been changed to `Group name` since the search is not limited to permission groups only but also open to other groups, for example, workflow group or dynamic group.	Is User in a Group [page 344]

Type of Change	Description	More Info
New	This rule function provides the capability to check whether the given user is part of the given permission group, so that customers can restrict access based on this rule function. Use this new function to limit the search in the rule function to search only in permission groups.	Is User in Permission Group [page 345]
Changed	We added information about the Initiate PM Form On Job Change event.	Initiate PM Form On Job Change Event [page 338]

1H 2024

Type of Change	Description	More Info
Changed	We added information about the behavior of this rule for different status values of the Payroll Control Record.	Get Payroll Area Control Record [page 283]
Changed	We've added information about an additional amendment scenario in the rule functions *Get Number Of Allowances In Period()* and *Has Allowances In Period()*.	• Has Allowances In Period() [page 330] • Get Number of Allowances in Period() [page 261]

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

www.sap.com/contactsap

THE BEST RUN SAP