

# Oracle Fusion Cloud Human Resources

---

## **Administering Fast Formulas**



# Contents

<b>Get Help</b>	i
<b>1 Overview</b>	1
Overview of Using Formulas	1
<b>2 Writing Formulas</b>	3
Overview of Writing Formulas	3
Improve Formula Performance	4
Formula Compilation Errors	7
Formula Execution Errors	8
Example of Writing a Fast Formula Using Expression Editor	10
Example of Writing a Fast Formula Using Formula Text	11
Formula Compilation	12
FAQs for Writing Formulas	14
<b>3 Fast Formula Components</b>	15
Overview of Using Fast Formula Components	15
Types of Variables	18
Formula Statements	18
Naming Schemes for Variables	22
Database Items	24
Overview of Generating Flexfield Database Items	25
Example of Generating Flexfield Database Items	26
Formula Operators	28
Literals	29
How Formulas Determine Variable Data Types	31
Array Variables	32
Formula Contexts	34
Overview of Working Storage Area	35
Overview of Calling a Formula from a Formula	36
Examples of Calling a Formula from a Formula	38

<b>4</b>	<b>Fast Formula Functions</b>	<b>41</b>
	Overview of Fast Formula Functions	41
	Text Formula Functions	41
	Numeric Formula Functions	45
	Date Formula Functions	46
	Data Conversion Formula Functions	49
	Miscellaneous Formula Functions	49
<b>5</b>	<b>Formulas for Payroll</b>	<b>53</b>
	Overview of Payroll Formulas	53
	Create Conversion Formulas for Proration	54
	Configure Periodicity Conversion Rules	55
	Element Input Validation Formula Type	56
	User Table Validation Formula Type	59
	Payroll User Interface Configuration Formula Type	60
	Payroll Access to HR Formula Type	65
	Payroll Relationship Group Formula Type	68
	Flow Schedule Formula Type	69
	Balance Exception Formula Type	71
	Element Skip Formula Type	72
	Get the Rate Amount for an Assignment Base Rate	73
<b>6</b>	<b>Formulas for Absence Management</b>	<b>75</b>
	Options to Define Qualification Plans	75
	Options to Define Accrual Plans	76
	Formulas for Accrual Plans	78
	Global Absence Accrual Matrix Formula	80
	Global Absences Annual Accrual Limit	84
	Global Absences Annual Accrual Limit Proration	87
	Global Absence Carryover	89
	Global Absence Carryover Proration	92
	Global Absence Ceiling	94
	Global Absence Ceiling Proration	97
	Global Absence Discretionary Disbursement Rule	99
	Global Absence Elective Disbursement Evaluation Formula	102
	Global Absence Discretionary Donation Rule	105
	Global Absence Partial Period Accrual Rate	108

---

Global Absence Plan Duration	111
Global Absence Plan Enrollment End	114
Global Absence Plan Enrollment Start	116
Global Absence Plan Period Anniversary Event Date	118
Global Absence Plan Use Rate	120
Global Absence Proration	124
Global Absence Rollover	127
Global Absence Rollover Proration	129
Global Absence Target Plan	132
Global Absence Transfer	134
Global Absence Transfer Proration	137
Global Absence Vesting Period	139
Formulas for Qualification Plans	142
Global Absence Plan Duration	142
Global Absence Band Entitlement	146
Global Absence Plan Enrollment Start Date	151
Global Absence Plan Entitlement	155
Global Absence Plan Roll Backward End	163
Global Absence Plan Roll Forward Start	168
Global Absence Plan Use Rate	173
Formulas for Absence Type	178
Global Absence Entry Validation	178
Global Absence Linkage Exclusion	184
Global Absence Type Duration	190
Frequently Used Formula Functions	193
Database Items	200

---

<b>7 Formulas for Compensation Plans</b>	<b>201</b>
Overview of Formulas for Compensation Plans	201
Formula Type Is Compensation Currency Selection	201
Formula Type Is Compensation Default and Override	203
Formula Type Is Compensation Hierarchy Determination	206
Formula Type Is Compensation Person Selection	208
Formula Type for Total Compensation Item	210
Using Profiles DBI in Fast Formula	214

## 8 Formulas for Benefits 217

---

Overview of Benefits Formulas	217
Person Change Causes Life Event	217
Participation and Rate Eligibility	222
Rounding	225
Age Calculation	226
Person Selection	228
Benefits Extract Custom Data Rule	230
Life Event Reason Timeliness	231
Coverage Amount Calculation	233
Rate Value Calculation	235
Dependent Eligibility	238
Age Determination Date	240
Enrollment Opportunity	242
Beneficiary Certification Required	246
Compensation Calculation	248
Post Election Edit	250
Enrollment Period Start Date	256
Default Enrollment	257
Evaluate Life Event	262
Waiting Period Value and UOM	264
Coverage Amount Limit	266
Rate Periodization	267
Dependent Certification Required	271
Length of Service Date to Use	273
Enrollment Certification Required	275
Coverage Upper Limit	278
Rate Start Date	279
Enrollment Coverage Start Date	281
Default to Assign Pending Action	282
Extra Input	289
Common Mistakes in Formulas	291
Formula Debugging and Troubleshooting	296
Formula – FAQ	297
Appendix 1: Contexts	300
Appendix 2: Input Values	321
Appendix 3: Formula Function	341

Appendix 4: Formula Errors: Compile time	350
Appendix 5: Formula Errors: Run time	351
Appendix 6: Database Items	352

## **9 Formulas for Performance Management** **369**

---

How You Use Fast Formulas in Performance Documents	369
--	-----

## **10 Formulas for Time and Labor and Workforce Management** **371**

---

Where to find DBI and Fast Formula Info for Time and Labor and Workforce Management?	371
--	-----



# Get Help

There are a number of ways to learn more about your product and interact with Oracle and other users.

## Get Help in the Applications

Some application pages have help icons  to give you access to contextual help. If you don't see any help icons on your page, click your user image or name in the global header and select Show Help Icons. If the page has contextual help, help icons will appear.

## Get Training

Increase your knowledge of Oracle Cloud by taking courses at [Oracle University](#).

## Join Our Community

Use [Cloud Customer Connect](#) to get information from industry experts at Oracle and in the partner community. You can join forums to connect with other customers, post questions, suggest [ideas](#) for product enhancements, and watch events.

## Share Your Feedback

We welcome your feedback about Oracle Applications user assistance. If you need clarification, find an error, or just want to tell us what you found helpful, we'd like to hear from you.

You can email your feedback to [oracle\\_fusion\\_applications\\_help\\_ww\\_grp@oracle.com](mailto:oracle_fusion_applications_help_ww_grp@oracle.com).

Thanks for helping us improve our user assistance!



# 1 Overview

## Overview of Using Formulas

Oracle Fast Formula is a simple way to write formulas using English words and basic mathematical functions. You can use information from your database in formulas without learning the database structure or a programming language.

Each formula usage summarized in this topic corresponds to one or more formula types, requiring specific formula inputs and outputs.

The formulas delivered in Oracle HCM Cloud contain English language names, but you can write formulas in other languages. You can provide formula names and descriptions in any language. Formula text isn't translated, but can handle non-English user-defined elements, input values, or balances. For example, if you define an element name in Chinese, the base element name is stored in Chinese. If you create a formula, it can have variable names or string values in other languages and character sets like Chinese.

### Calculate Payrolls

You can write payroll calculations and skip rules for elements to represent earnings and deductions.

You can do these tasks using fast formulas:

- Control the repetition of a payroll flow.
- Control when elements are processed in a payroll run.
- Define a set of people for payroll processing, data entry, or reporting.
- Validate one or more element entry values. Provide a default value for an element entry value, or calculate entry values based on the user's entries in other entry values.
- Validate entries in user-defined tables.
- Proration formulas control how payroll runs prorate element entries when they encounter an event such as a mid-period change in an element entry value.
- Associate more than one payroll formula with each element to perform different processing for employee assignments with different statuses.
- Transformation formulas convert inbound data into a format that HCM Data Loader understands.
- Configure payment method preferences. For example, you can limit the number and types of payment methods.

### Benefits Administration

You can use formulas to structure your benefit plans. Formulas provide a flexible alternative to the delivered business rules. Use formulas to configure:

- Date calculations, such as enrollment start and end dates, rate or coverage start and end dates, waiting periods and enrollment periods, or action item due dates
- Calculations of rate and coverage amount, minimum and maximum, or upper and lower limits
- Certification requirements

- Partial month and proration calculations
- Eligibility and participation evaluation

For example, you can write a formula to calculate benefits eligibility for those cases where the provided eligibility criterion doesn't accommodate your particular requirements.

**Note:** For more information, see Benefits Fast Formula Reference Guide (1456985.1) on My Oracle Support at <https://support.oracle.com>.

## Define Configuration for Compensation

Here's what you can modify using formulas and add flexibility to existing compensation plan configuration:

- Start and end dates for compensation allocations under individual compensation plans
- Person selection, hierarchy determination, column default values, and currency selection for workforce compensation plans
- The source of items displayed in total compensation statements

## Define Formulas to Create Rule Templates for Time and Labor

Use formulas with time repository rule templates to create rules. The formulas contain delivered combinations of rule parameters and output results. You can use one formula with multiple rule templates by varying the template configuration.

When creating a rule template, you select a formula name, and then configure the parameter type and display name of the parameters and variables. You don't have to redo the entire formula statement to determine what details to change to achieve a particular outcome.

Here's what you can use formulas to apply in Time and Labor:

- Logic for processing or calculating time
- Parameters that enable rules to pass values to the formula for use in calculations
- Output variables that the formula uses to return calculation results to the rules

For example, the Period Maximum Hours Template uses the WFM\_PERIOD\_MAXIMUM\_TIME\_ENTRY\_RULE formula to compare reported time category hours to defined maximum hours.

# 2 Writing Formulas

## Overview of Writing Formulas

Writing fast formulas enables you to perform a wide variety of tasks using English words and basic mathematical functions. There are several techniques that you can use to ensure your formulas are easy to read, use, and understand.

You write fast formulas to validate or calculate data for payroll and other Oracle Fusion applications.

This table lists the key aspects of writing fast formulas.

Key Aspects	Description
Using the expression editor	<p>Use the expression editor with these formula types:</p> <ul style="list-style-type: none"><li>• Deduction component groups</li><li>• Deduction component detail groups</li><li>• Payroll relationship groups</li><li>• Work relationship groups</li></ul> <p>Use these formula types when creating dynamic object groups. Object groups define subsets of deduction components or people for processing or reporting.</p>
Using formula text	<p>Most formula types use the text editor to define formulas. The text editor is more flexible than the expression editor. Here's what you can do with the text editor:</p> <ul style="list-style-type: none"><li>• Add database items applicable to the formula type</li><li>• Use functions to manipulate data</li><li>• Use user-defined global values</li></ul> <p>Database items are read-only variables that you can use to get application data without having knowledge of the underlying data model, and writing queries to get the data.</p> <p>The input values that you have passed and the output values returned depend upon the context in which you're using the formula. The names and types of these values are defined in many scenarios, so refer the appropriate product documentation.</p>
Improving performance	<p>Here's what you can do to improve performance:</p> <ul style="list-style-type: none"><li>• Shorten your formulas.</li><li>• Don't refer to database items until needed.</li><li>• Use an <b>ALIAS</b> instead of assigning a database item to a local variable.</li><li>• Use input statements instead of database items.</li><li>• Leave a WHILE loop as early as possible.</li><li>• Limit use of CHANGE_CONTEXT blocks.</li><li>• Change as many contexts as possible in one CHANGE_CONTEXTS block instead of having CHANGE_CONTEXTS blocks inside other CHANGE_CONTEXTS blocks.</li></ul>

Key Aspects	Description
	<ul style="list-style-type: none"> <li>Don't use <code>CHANGE_CONTEXTS</code> to set contexts that you would reasonably expect to be set already, and aren't going to change.</li> <li>Use the database item that would retrieve the required data. Some database items are used in reporting, and are unsuitable for use in formulas because they return a lot of inefficient data. Avoid these database items in formulas because they require more complex formulas to get the required data, and they make performance worse.</li> </ul>
Avoiding compilation errors	<p>Compile a modified formula before using it. Here's what you can do to avoid common compilation errors:</p> <ul style="list-style-type: none"> <li>Look for syntax errors, such as using <code>IF1</code> instead of <code>IF</code>.</li> <li>Ensure that the statement order is correct. For example <code>ALIAS</code>, <code>DEFAULT</code>, or <code>INPUT</code> statements come after other statements.</li> <li>Don't misuse <code>assignment</code> statements by assigning a value to a database item.</li> <li>Don't misuse the <code>ALIAS</code> statement. It can only be used for database items.</li> <li>Don't misuse <code>DEFAULT</code> statements. You must specify a <code>DEFAULT</code> statement for a variable other than an input or database item.</li> <li>Look out for missing <code>DEFAULT</code> statements. Database items that have a default value specified must contain a <code>DEFAULT</code> statement.</li> </ul>
Avoiding execution errors	<p>Here's what you can do to avoid common execution errors:</p> <ul style="list-style-type: none"> <li>Ensure that variables are initialized. For example, a variable initialized inside an <code>IF</code> block might be initialized when it's used.</li> <li>Verify that your formula calculations won't end up dividing a number by zero.</li> <li>Ensure that the formula compiler can find the data.</li> <li>Ensure that the value that your formula generates doesn't exceed the permissible range.</li> </ul>

## Improve Formula Performance

Use various approaches to ensure that your formulas are easy to read, use, understand, and processed efficiently.

### Variable Names and Aliases

Use concise and meaningful variable names. If the names of database items are long, you can use aliases. The length of database item name doesn't impact the performance or memory usage of a formula.

### Inputs Statements

**Tip:** The best approach is to use `INPUTS` statements rather than database items whenever possible. You don't have to access the database for input variables, which in turn speeds up your payroll run.

Here's an example of a formula without the `INPUTS` statement:

```
SALARY = SALARY_ANNUAL_SALARY / 12
```

```
RETURN SALARY
```

Here's how you can use the `INPUTS` statement efficiently:

```
INPUTS ARE ANNUAL_SALARY
SALARY = ANNUAL_SALARY / 12
RETURN SALARY
```

## CHANGE\_CONTEXTS Statement

**Tip:** The best approach is to use `CHANGE_CONTEXTS` statement only when required, because `CHANGE_CONTEXTS` can cause database item values to be fetched again from the database.

You can perform multiple context changes using a single `CHANGE_CONTEXTS` statement, instead of calling `CHANGE_CONTEXTS` from other `CHANGE_CONTEXTS` blocks.

For example, use a single `CHANGE_CONTEXTS` statement in the first formula, which is efficient than the second formula:

```
CHANGE_CONTEXTS(EFFECTIVE_DATE = l_eff_date, AREA1= 'California')
(
/*Do something here*/
)

CHANGE_CONTEXTS(EFFECTIVE_DATE = l_eff_date)
(
CHANGE_CONTEXTS(AREA1 = 'California')
(
/*Do something here*/
)
)
```

Don't use the `CHANGE_CONTEXTS` statement to set contexts that you would reasonably expect to be already set.

For example, the `PERSON_ID` context is generally already set in a formula that processes person information. If this context isn't set, then `GET_CONTEXT` doesn't fetch any value. `GET_CONTEXT` only returns the default value of -1. This code doesn't achieve anything:

```
l_person_id = GET_CONTEXT(PERSON_ID, -1)
l_effective_date = GET_CONTEXT(EFFECTIVE_DATE, '0001-01-01 00:00:00) (DATE))

CHANGE_CONTEXTS(PERSON_ID = l_person_id, EFFECTIVE_DATE = l_effective_date)
(
/*Do something here*/
)
```

## Database Items

**Tip:** The best approach is to retain only the required database items in a formula. Don't refer to database items in a formula unless it's absolutely required. Sometimes a formula might contain databases that it doesn't need, which causes the formula to make unnecessary database calls. As a result, the efficiency of the formula is impacted.

In this example, the use of database items always causes a database fetch for `AGE`.

```
S = SALARY
A = AGE
IF S < 20000 THEN
IF A < 20 THEN
```

```
TRAINING_ALLOWANCE = 30
ELSE
TRAINING_ALLOWANCE = 0
```

In this example, the use of database items fetches `AGE` only if salary is less than 20000:

```
IF SALARY < 20000 THEN
IF AGE < 20 THEN
TRAINING_ALLOWANCE = 30
ELSE
TRAINING_ALLOWANCE = 0
```

## Database Items for HCM Extracts

Some database items are intended only for HCM Extracts reporting. These database items return a high amount of data and retrieve it inefficiently when used in a formula.

For example, an HCM Extracts database item retrieves information for all employees on the application, or complete information of a particular employee since the start of employment. This data retrieval is fine for reporting as reports process lots of data. Formulas usually process small amount of data limited to specific cases. HCM Extracts allow extra conditions to be added to limit the data returned, but formulas don't.

With these database items, more complex formulas are required to get the right data and the performance of the formulas is extremely poor because of the large data volumes. For example, the array database item `PER_EXT_ALL_PHONE_AREA_CODE` returns all phone area codes as of an effective date. Together with `PER_EXT_ALL_PHONE_PERSON_ID`, and `PER_EXT_ALL_PHONE_TYPE`, it's possible to get a person's home phone's area code.

But for a simpler formula and better performance, you can use the `PER_PER_HOME_PHONE_AREA_CODE` database item to retrieve the home phone's area code. `PER_PER_HOME_PHONE_AREA_CODE` uses `PERSON_ID` and `EFFECTIVE_DATE` contexts to get its data. `PER_EXT_ALL_PHONE_AREA_CODE` only uses the `EFFECTIVE_DATE` context.

## While Loop

Use an `EXIT` statement or a change to the loop condition to leave a `WHILE` loop immediately once you complete your task. For example, don't continue when you have already found a single item that you were looking for.

## Formula Logging

Formula logging can help during implementation when developing formulas. It can also help with problem diagnosis. When investigating problems logged through a support service request, Oracle may ask for logging information.

Although logging can be very useful, the logging process may impact performance and slow down processing in general. Other processes waiting on resources utilized by the logging process, may take longer to complete or be delayed in starting.

It is advisable to remove unnecessary logging statements once the purpose is served. If there is a need to retain logging statements for diagnostic purposes then it is suggested that they are executed conditionally, for example:

```
debug = 0
if (debug = 1) then
(
/* Log statements here. */
)
```

When logging is required, set `debug = 1` and recompile the formula.

**Note:** Don't forget to disable logging when finished.

# Formula Compilation Errors

Compilation errors are displayed in the Fast Formulas page after you compile the formula. The compiler stops the compilation process when it encounters an error. Error messages display the line number and the type of error.

## Common Compilation Errors

Here's some of the common formula compilation errors.

Formula Compilation Error	Description
Syntax Error	The formula text violates the grammatical rules for the formula language. For example, if you use <b>IF1</b> instead of <b>IF</b> in an <b>IF</b> statement.
Incorrect Statement Order	<b>ALIAS</b> , <b>DEFAULT</b> , or <b>INPUT</b> statements come after other statements.
Misuse of <b>ASSIGNMENT</b> Statement	Occurs when any of these conditions exist: <ul style="list-style-type: none"><li>An <b>ASSIGNMENT</b> assigns a value to a database item.</li><li>A context is assigned a value externally to a <b>CHANGE_CONTEXTS</b> statement.</li><li>The formula assigns a value to a non-context variable within a <b>CHANGE_CONTEXTS</b> statement.</li></ul> You can use <b>CHANGE_CONTEXTS</b> statements in a formula.
Misuse of <b>ALIAS</b> Statement	You can use an <b>ALIAS</b> statement only for a database item.
Missing <b>DEFAULT</b> Statement	A database item that specifies a default value must have a <b>DEFAULT</b> statement.
Misuse of <b>DEFAULT</b> Statement	You specify a <b>DEFAULT</b> statement for a variable other than as an input or a database item.
Uninitialized Variable	The compiler detects that a variable is uninitialized when used. The compiler can't do this in all cases. This error often occurs when the formula includes a database item that requires contexts that the formula type doesn't support. The formula treats the database item as a local variable. For example, balance database items require the <b>PAYROLL_REL_ACTION_ID</b> , <b>PAYROLL_ASSIGNMENT_ID</b> and <b>CALC_BREAKDOWN_ID</b> contexts. Typically, you use these statements in formulas of type Oracle Payroll.
Missing Function Call	The compiler doesn't recognize a function call. The combination of return type, function name, and parameter types doesn't match any available function.
Incorrect Operator Usage	An instance of a formula operator use doesn't match the permitted uses of that operator.  For example, the <b>+</b> operator has two permitted uses. The operands are both of data type <b>NUMBER</b> , or both of data type <b>TEXT</b> .
Inconsistent Data Type Usage	The formula uses a formula variable of more than one data type. Or the formula uses a database item or context with the wrong data type.

Formula Compilation Error	Description
	For example, Variable A is assigned a <b>NUMBER</b> value at the start of the formula, but is assigned a <b>TEXT</b> value later in the formula.
<b>EXIT</b> Statement Not Within <b>WHILE</b> Loop	A condition that eventually becomes false or an <b>EXIT</b> call for exiting the loop doesn't exist.
Misuse of Context	The formula uses a variable as a context, or a context as a variable.  For example, a formula assigns a value to <b>AREA1</b> as an ordinary variable, but later uses <b>AREA1</b> as a context in a <b>GET_CONTEXT</b> call.

## Formula Execution Errors

Fast formula execution errors occur when a problem arises while a formula is running. Typically, data-related issues cause these errors either in the formula or in the application database.

### Formula Execution Errors

Here's some of the formula execution errors.

Formula Execution Error	Description
Uninitialized Variable	When the formula compiler can't fully determine if a variable or context is initialized, it generates a code to test if the variable is initialized.  When the formula runs, the code displays an error if the variable or context isn't initialized.
Divide by Zero	Raised when a numeric value is divided by zero.
No Data Found	Raised when a non-array type database item unexpectedly fails to return any data. If the database item can't return data, then it should provide a default value.  You can specify a default value using a <b>DEFAULT</b> statement. An error in formula function code can also cause this error message.
Too Many Rows	Raised when a non-array type database item unexpectedly returns more than a single row of data. The cause is an incorrect assumption made about how the data is being accessed.  An error in the formula function code can also cause this error message.
<b>NULL</b> Data Found	Raised when a database item unexpectedly returns a <b>NULL</b> data value. If the database item can return a <b>NULL</b> value, then it provides a default value.

Formula Execution Error	Description
	<p><b>Note:</b> Some database items can't return a <b>NULL</b> value. If the database items can return a <b>NULL</b> value, then you can provide a default value for that database item.</p>
Value Exceeded Allowable Range	Raised for a number of reasons, such as exceeding the maximum allowable length of a string.
Invalid Number	Raised when a formula attempts to convert a nonnumeric string to a number.
User Defined Function Error	Raised from within a formula function. The error message text is provided as part of the formula error message.
External Function Call Error	A formula function returns an error, but doesn't provide any additional information to the formula code. The function may have sent error information to the logging destination for the executing code.
Function Returned <b>NULL</b> Value	A formula function returns a <b>NULL</b> value.
Too Many Iterations	A single <b>WHILE</b> loop, or a combination of <b>WHILE</b> loops has exceeded the maximum number of permitted iterations. This error is raised to terminate loops that can never end, which indicates a programming error within the formula.
Array Data Value Not Set	The formula tries to access an array index that has no data value. This error occurs in the formula code.
Invalid Type Parameter for <b>WSA_EXISTS</b>	You specify an invalid data type in the <b>WSA_EXISTS</b> call.
Incorrect Data Type For Stored Item	When retrieving an item using <b>WSA_GET</b> , the actual data type doesn't match that of the stored item. This error occurs within the calling formula.
Called Formula Not Found	The called formula isn't found when attempting to call a formula from a formula. This error may occur due to an issue in the calling formula, or because of installation issues.
Recursive Formula Call	An attempt was made to call a formula from itself. The call could be made directly or indirectly from another called formula. Calling a formula in a recursive manner isn't permitted.
Input Data Has Different Types in Called and Calling Formulas	When calling a formula from a formula, the input data type within the called formula doesn't match the data type specified in the calling formula.
Output Has Different Types In Called and Calling Formulas	When calling a formula from a formula, the output data type within the called formula doesn't match the data type specified in the calling formula.
Too Many Formula Calls	When a formula calls another formula in its text, resulting in a hierarchy. The maximum depth of the hierarchy is 10.

# Example of Writing a Fast Formula Using Expression Editor

In this example, you create a fast formula that groups executive workers for reporting and processing. All executive workers are in department EXECT\_10000.

After you create the formula, you need to add it to the object group parameters, so that only the workers that belong to the EXECT\_10000 department are used in processing.

Here are the key decisions when deciding on which formula to create.

Key Decisions	In This Example
Is the formula for a specific legislative data group?	Yes, InVision
What's the formula type for this formula?	Payroll Relationship Group

## Creating a Fast Formula Using the Expression Editor

1. On the Home page, click the **Fast Formulas** quick action under the **My Client Groups** tab.
2. Click **Create**.
3. Complete these fields:

Fields for Fast Formula	Values for Fast Formula
Formula Name	Executive Payroll Relationship Group
Type	Payroll Relationship Group
Description	Executive Workers
Legislative Data Group	Vision LDG
Effective As-of Date	1-Jan-2010
Type of Editor	Expression Builder

Fields for Fast Formula	Values for Fast Formula
	<p><b>Note:</b> For more complex conditions to create a group, you can select Text. However, after you save the formula, you can't change the type of editor.</p>

- Click **Continue**.
- In the Formula Details section, click **Add After** to add a row and complete these fields:

Conjunction	Database Item Name	Data Type	Operand	Literal Value
None applicable	DEPARTMENT	Character	=	'EXECT_10000'
And	SELECT_EMP	Character	=	'YES'

- Click **Compile**.
- Click **Save**.

## Example of Writing a Fast Formula Using Formula Text

In this example, you use the text editor to create a fast formula.

Here are the key decisions when deciding on which formula to create:

Key Decisions	In This Example
Is the formula for a specific legislative data group?	No, this is a global formula that can be used by any legislative data group.
Are there any contexts used in this formula?	No
Are there any database item defaults?	Yes, PER_ASG_JOB_NAME
Are there any input value defaults?	No
What are the return values?	MIN_HOURS, MAX_HOURS, FREQUENCY

## Creating a Fast Formula Using the Text Editor to Determine a Manager's Scheduled Hours

1. Search for and select **Fast Formulas** in **My Client Groups**.
2. Click **Create**.
3. Complete these fields:

Fields for the Fast Formula	Values for the Fast Formula
Formula Name	Manager Range of Scheduled Hours
Formula Type	Range of Scheduled Hours
Description	Manager's Range of Hours
Effective Start Date	1-Jan-2010

4. Click **Continue**.
5. Enter these formula details in the Formula Text section:

```
/* DATABASE ITEM DEFAULTS BEGIN */
DEFAULT FOR per_asg_job_name IS ''
/* DATABASE ITEM DEFAULTS END */
JOB_1 = PER_ASG_JOB_NAME
IF JOB_1 = 'Manager' then
  (MIN_HOURS = 25
  MAX_HOURS = 40
  FREQUENCY = 'H')
else
  (MIN_HOURS = 20
  MAX_HOURS = 35
  FREQUENCY = 'H')
return MIN_HOURS, MAX_HOURS, FREQUENCY
```

6. Click **Compile**.
7. Click **Save**.

## Formula Compilation

All fast formulas must be compiled before they can be utilized or executed within the application.

Use the **Submit a Flow** task to submit the Compile Formula process.

1. Navigate to My Client Groups, Payroll, Submit a Flow.
2. Select the Legislative Data Group.
3. Search for the **Compile Formula** flow.
4. Enter/Select values as required:

Name	Values	Description
Payroll Flow	Enter any meaningful name that can be used to monitor the progress of the flow.	NA
Formula (Optional)	Enter the formula name.	Compile a single formula based on the formula name.
Formula Type (Optional)	For example, Oracle Payroll, Rate conversion, User Table Validation etc.	You can compile only those formula defined with a particular Formula Type.
Status	<ul style="list-style-type: none"> <li><input type="radio"/> Compiled And Not Compiled</li> <li><input type="radio"/> Not Compiled</li> </ul>	<p>You can compile only</p> <ul style="list-style-type: none"> <li><input type="radio"/> not compiled formula</li> </ul> <p>or compile all <b>not compiled and compiled</b> formula.</p>
Scope	<ul style="list-style-type: none"> <li><input type="radio"/> Predefined and User Defined Formula</li> <li><input type="radio"/> Predefined Formula</li> <li><input type="radio"/> User Defined Formula by Enterprise and LDG</li> <li><input type="radio"/> User Defined Formula by LDG</li> <li><input type="radio"/> User Defined Formula by Legislation</li> </ul>	You can determine which formula to compile. This parameter restricts the set of formulas to those selected for compile.
Process Multithreaded	<ul style="list-style-type: none"> <li><input type="radio"/> No</li> <li><input type="radio"/> Yes</li> </ul>	<p>You can select whether to run multiple threads.</p> <p>Compile Formula process uses its own logic to determine the number of threads when running multi threaded. It does not use the Threads parameter within the Process Configuration Group.</p>

5. Click Submit.
6. Monitor the status of the process.

## FAQs for Writing Formulas

### When do I run the Compile Formula process?

When you create or update multiple fast formulas simultaneously, run the Compile Formula process on the Submit a Process or Report page.

### What's the difference between a formula compilation error and an execution error?

Compilation errors occur on the Fast Formulas page when you compile the formula. An error message explains the nature of the error.

Common compilation errors are syntax errors resulting from typing mistakes. You can view error messages on the dashboard or go to the messages tab directly after the process is run.

Execution errors occur when a problem arises while a formula is running. Typically, data-related issues either in the formula or in the application database cause these errors.

# 3 Fast Formula Components

## Overview of Using Fast Formula Components

When you're developing a fast formula, you must understand the formula language, the rules that the application imposes on the fast formula, and the calculation requirements.

Create fast formulas using these components:

- Assignment statements
- Return statements
- Variables
- Input statements
- Expressions
- Conditions
- Comments

**Note:** Other topics explain additional components that you can use in fast formulas. These include literals, database items, working storage area, calls to other formulas, functions, and operators.

Let's look at an example to understand how each component is used in a fast formula. Suppose you want to calculate the pay value for the `WAGE` element by multiplying the number of hours an employee works each week by the hourly rate. Here's how you can write the formula in this example:

```
WAGE = HOURS_WORKED * HOURLY_RATE
RETURN WAGE
```

### Assignment Statements

An assignment statement assigns a value to the `WAGE` element.

### Return Statements

A return statement passes the `WAGE` value back to the payroll run. You can use a return statement to stop the formula execution without passing any values.

### Variables

Variables are of these classes:

- Input variables appear in INPUTS statements and bring values into a fast formula.
- Output variables appear in RETURN statements and return values from a fast formula. A variable can be both an input and output.
- Local variables are only used within one formula.

You can change a local variable within the formula by assigning a value to it using an assignment statement. To calculate the `WAGE` value, the fast formula needs to get the value for the `HOURS_WORKED` variable.

You can use local variables to store data in a fast formula. You might want to hold data temporarily while you perform some other calculations, or pass data back to the application. Here's an example of the `ANNUAL_LEAVE` variable.

```
/* Formula: Annual Leave Formula */
IF YEARS_SERVICE >= 10
THEN
ANNUAL_LEAVE = 25
ELSE
ANNUAL_LEAVE = 20 + FLOOR (YEARS_SERVICE/2)
RETURN ANNUAL_LEAVE
```

## Input Statements

You can use `HOURS_WORKED` as an input value of the `WAGE` element. To pass the element input values to the fast formula during processing, define an input statement like this:

```
INPUTS ARE HOURS_WORKED
WAGE = HOURS_WORKED * HOURLY_RATE
RETURN WAGE
```

**Note:** This is a payroll application example. The name used in the input statement must be the same as the name of the element input value. Multiple words must be joined by underscores. Other input statements that have nothing to do with elements would have their own rules for formula input variables. In this example, the `HOURS_WORKED` input variable is numeric. If the input variable isn't numeric, you must specify the type. For example,

```
INPUTS ARE START_DATE (DATE)
```

## Expressions

Each function or calculation is one expression. You can nest expressions to create more complex calculations. You can use brackets to control the order in which calculations are done.

The formula evaluates expressions within the brackets first. Within nested brackets, evaluation proceeds from the least inclusive set to the most inclusive set. If you don't use brackets, the formula evaluates expression in this order:

- Multiplication, Division
- Addition, Subtraction

**Note:** For same priority, the formula evaluates the expression from left to right.

Expressions combine constants and variables with operators (+, -, \*, /), array methods, and functions to return a value of a certain data type. For example, the expression `(3 + 2)` returns a value of 5, and is a `NUMBER` data type. The format of an expression is:

```
SUBEXPRESSION [operator SUBEXPRESSION ...]
```

You can combine a number of sub-expressions into a single expression. For example, you can combine the sub-expressions `(3 + 2)` and `MONTHS_BETWEEN(start_date, end_date)` into a single expression as follows:

```
(3 + 2) + MONTHS_BETWEEN(start_date, end_date)
```

You can also use expressions inside functions, such as:

```
salary = GREATEST(minimum_wage, (hourly_rate * hours_worked))
```

Operands in an expression are usually of the same data type, which is the data type of the expression as a whole. Here's an example of an expression in which all the operands are numeric and the expression itself is numeric:

```
GREATEST(MINIMUM_WAGE, (HOURLY_RATE * HOURS_WORKED)) + BONUS
```

**BONUS** is the operand for the above expression. The return value is **GREATEST**. The arguments for **GREATEST** are separate expressions.

## Conditions

You can use conditions to process expressions based on whether a certain condition occurs. For example:

```
TRAINING_ALLOWANCE = 0
IF (AGE < 20) THEN
  TRAINING_ALLOWANCE = 30
```

This formula checks if the condition (AGE < 20) is true or false. If it's true, the formula processes the statement that follows the word **THEN**. If the condition is false, the formula ignores this statement.

## Comments

Use comments to explain all or part of a fast formula. Also, you can change some formula lines into comments until they're ready to be used. You can place comments anywhere within a formula. The beginning of a fast formula should contain these comments:

- The formula title and a short purpose statement.
- A description of the formula inputs.
- A list of variables and literals that may require updating.
- An explanation of the formula's calculation.
- The dates of any modifications, the name of the person modifying the formula, and the reason for the change.

## Multi-Line Comments

Multi-line comments are designated by the comment delimiters of /\* and \*/. Anything written inside these delimiters is a comment.

**Note:** Do not put a multi line comment within a multi line comment, because it causes a syntax error when you compile the formula.

## Single Line Comments

Fast formula also supports single line comments. The # character is used for the start of single line comments. The # character itself and any text after it to the end of the line are ignored.

Comments Example:

```
# This line is a single line comment and will be ignored.

/*
 * This is a multi-line comment.
 */
```

```
a = 1 # Ignore the # character and everything after it on the line.  
b = '####' # b is the string ####. The remainder of the line is ignored.  
return a, b
```

## Types of Variables

Formula variables can have values that change frequently. The variable's data type indicates the type of information the variable holds. The fast formula determines how you can use the variable.

For example, if you set a variable to 'J. Smith', the fast formula interprets it as a TEXT variable. The application displays a warning if you try to perform any inconsistent operations, such as trying to add a number to a text string.

Variables can be of the following types:

- *Input* variables appear in `INPUTS` statements and bring values into a fast formula.
- *Output* variables appear in `RETURN` statements and return values from a fast formula. A variable can be both an input and output.
- *Local* variables are only used within one fast formula.

You can change variable values by using an assignment statement and by referencing them within expressions. However, if you reference a variable to which you didn't assign a value, the fast formula returns an error.

### Related Topics

- [Overview of Using Fast Formula Components](#)
- [Array Variables](#)
- [Formula Contexts](#)

## Formula Statements

You use formula statements to provide instructions that you want your formula to carry out. When working with statements, it's important to have knowledge of the different statement types, the required order, and how to group statements.

### Statement Types

Here are the statements that you can use in your formulas to provide instructions.

Statement	Statement Form	Description and Example
<code>ALIAS</code>	<code>ALIAS name1 AS name2</code>	Provides a different name for a database item or global value. Sometimes the database item names that the application provides are too long to use in a formula.

Statement	Statement Form	Description and Example
		<p>Use the ALIAS statement to shorten the name of a database item. Once the ALIAS is created, use it instead of the database item name. Using an alias is more efficient than assigning the database item to a local variable with a short name.</p> <p><code>ALIAS OVERTIME_QUALIFYING_LENGTH_OF_SERVICE AS OT_QLS</code></p>
<b>ASSIGNMENT</b>	<code>variable = expression</code> <code>array[index] = expression</code>	<p>Assigns an expression value to a variable or an array variable at an index position. A formula evaluates the expression on the right-hand side of the statement. It places its result in the variable you name on the left-hand side. The left side of an assignment statement must always be a local variable because a formula can only change the value of local variables.</p> <p>Within a "CHANGE_CONTEXTS" statement, assign values only to contexts. Outside a "CHANGE_CONTEXTS" statement, assign values only to input, output, and local variables.</p> <p><code>RATE = HOURLY_RATE + 14</code>  <code>WAGE = HOURS_WORKED * RATE</code></p>
<b>CHANGE_CONTEXTS</b>	<code>(context1 = expression1 [,context2 = expression2 ])</code>	<p>Changes one or more contexts within a formula. Within the "CHANGE_CONTEXTS" statement, use <b>ASSIGNMENT</b> statements to assign the new values.</p> <pre>CHANGE_CONTEXTS (AREA1 = TAX_REPORTING_UNIT_INCOME_TAX_JURISDICTION_GEOGRAPHY_ID) (   CHANGE_CONTEXTS (DEDUCTION_TYPE =     'SBJ_TO_REGULAR_TAX')   (     L_TAXATION_METHOD = 'NONE'     EXECUTE ('TAXABILITY_RULE_EXISTS')     IF GET_OUTPUT('TR_EXISTS', 'N') =       'Y' THEN       L_TAXATION_METHOD = 'REGULAR_TAX'     ) /* DEDUCTION_TYPE context change     undone here. */   ) /* AREA1 context change undone   here. */</pre>
<b>DEFAULT</b>	<code>DEFAULT FOR variable IS literal</code> <code>DEFAULT_DATA_VALUE FOR variable IS literal</code>	<p>The DEFAULT FOR statement provides a value that the formula uses for a formula input or database item in these situations:</p> <ul style="list-style-type: none"> <li>• The input wasn't assigned a value.</li> <li>• The database item's value couldn't be retrieved from the database.</li> <li>• A non-array database item's value is NULL.</li> </ul>

Statement	Statement Form	Description and Example
		<p>The DEFAULT_DATA_VALUE FOR statement provides a value for an array database item where individual data values are NULL.</p> <p>Some database items are defined to require a default value because they could return no data or NULL values from the database.</p> <pre>DEFAULT FOR HOURLY_RATE IS 3.00 INPUTS ARE HOURLY_RATE X = HOURS_WORKED * HOURLY_RATE</pre>
EXIT	EXIT	<p>Immediately exits from the enclosing WHILE loop. You can't use the EXIT statement outside of a WHILE loop.</p> <pre>FOUND = -1 /* -1 is not a valid index for A. */ I = A.FIRST(-1) WHILE (A.EXISTS(I)) LOOP ( /* EXIT-clause for early exit. */ IF A[I] = KEY THEN ( FOUND = I /* Exit the loop. */ EXIT; ) I = A.NEXT(I,-1) )</pre>
FORMULA CALLING FORMULA	<pre>SET_INPUT(input [,value]): Set an input or context value in the called formula.</pre> <pre>EXECUTE(formula): Execute the called formula.</pre> <pre>GET_OUTPUT(output, default-value): Get a value returned from the called formula.</pre> <pre>IS_EXECUTABLE(formula): Test whether a formula is executable.</pre>	<p>Calls a formula from another formula. For example, formulas can call a small formula that performs a common calculation. You can use this approach to avoid writing long formulas.</p> <pre>SET_INPUT('UNIT','Hourly') EXECUTE('RATE_FORMULA') HOURLY_RATE = GET_OUTPUT('RATE',0.0) WAGE = HOURS_WORKED * HOURLY_RATE RETURN WAGE</pre>
IF	<pre>IF condition THEN statements</pre> <pre>IF condition THEN statements ELSE statements</pre>	<p>Executes one or more statements if a condition is true. Use the IF ELSE statement to specify a set of statements to execute if the condition is false.</p> <pre>IF (AGE &lt; 20) THEN TRAINING_ALLOWANCE = 30 ELSETRAINING_ALLOWANCE = 40</pre>
INPUT	INPUTS ARE input1 [,input2]	<p>Lists the input variables for the formula. There's only one INPUT statement in a formula.</p> <pre>INPUTS ARE HOURS_WORKED WAGE = HOURS_WORKED * HOURLY_RATE RETURN WAGE</pre>

Statement	Statement Form	Description and Example
<b>RETURN</b>	<b>RETURN</b> [ output1 ] [,output2]	<p>Causes a formula to stop executing immediately. For its value to be returned to the caller, you must enter a formula output variable in the <b>RETURN</b> statement that stopped the formula.</p> <p>You can enter multiple return statements in a formula.</p> <pre>INPUTS ARE HOURS_WORKED IF HOURS_WORKED &lt;= 10 THEN(   RETURN   /* This is ignored. */   BONUS = 10 ) /* This is executed if HOURS_WORKED &gt; 10. */ BONUS = 50 RETURN BONUS</pre>
<b>WHILE</b>	<b>WHILE</b> condition <b>LOOP</b> statements	<p>Executes a number of statements as long as one condition is true.</p> <p>If the <b>WHILE</b> statement loop performs too many iterations, an error occurs to prevent endless looping.</p> <pre>/* -1234 is not a valid index for A in this instance, so use as default. */ NI = A.FIRST(-1234) WHILE A.EXISTS(NI) LOOP   VA = A[NI] /* Do some processing with element at index NI. */   NI = A.NEXT(NI,-1234) /   Go to next index. )</pre>
<b>WORKING STORAGE</b>	<p><b>WSA_DELETE</b>([item]) - Deletes values from the storage area.</p> <p><b>WSA_EXISTS</b>(item[,type]) - Determine if an item exists .</p> <p><b>WSA_GET</b>(item, value) - Fetches values from the storage area.</p> <p><b>WSA_SET</b>(item, value) - Sets values from the storage area.</p>	<p>Stores reference data, which you can set, fetch, or delete.</p> <pre>/* Formula: RATE_SETTER */ WSA_SET('RATE:HOURLY1', 3.5) WSA_SET('RATE:HOURLY2', 4.0) WSA_SET('RATE:HOURLY3', 4.5) WSA_SET('RATE_FLAG', 'Y') /* Flag to say that the rates have been set. */</pre>

## Ordering Statements

Place the statements in this order in the formulas:

1. ALIAS statements, if any
2. DEFAULT statements, if any
3. INPUT statements, if any

#### 4. Other statements

## Grouping Statements

If you want to group more than one statement under IF/THEN statements, ELSE clauses, WHILE loops, or CHANGE\_CONTEXTS, enclose the group of statements within brackets. In the absence of brackets, the preceding statement applies only to the first statement.

Here's an example of how you can group statements:

```
I = A.FIRST
WHILE (A.EXISTS(I)) LOOP
(
  A[I] = I
  I = A.NEXT(I,-1)
)
```

Here's an example of how you should not group statements::

```
I = A.FIRST
WHILE (A.EXISTS(I)) LOOP
  A[I] = I
  I = A.NEXT(I,-1) /* This is not executed as part of the loop. */
```

### Related Topics

- [Overview of Calling a Formula from a Formula](#)
- [Overview of Working Storage Area](#)

## Naming Schemes for Variables

You can use naming schemes for variables when adding them to formulas; however, you must avoid using reserved words as variable names.

## Naming Schemes

You can use one of these naming schemes:

- Variable names comprise one or more words, joined by underscores. The words must each start with a letter and can be followed by a combination of letters and digits.
- Variable names begin and end with double quotes (""). Between the quotes, you can use any printable characters, such as "This is a quoted variable name".

**Note:** Any word consisting of only digits could be mistaken for numbers.

Formulas aren't case sensitive. For example, the variable named EMPLOYEE\_NAME is the same as the variable employee\_name.

**Tip:** The maximum length of a variable name is 255 characters.

## Reserved Words

Ensure that you don't use these reserved words as variable names:

Source of Reserved Words	Reserved Words
Statements	<code>ALIAS AND ARE AS CHANGE_CONTEXTS DEFAULT DEFAULT_DATA_VALUE DEFAULTED ELSE EXIT FOR IF INPUTS IS LIKE LOOP NEED_CONTEXT NOT OR RETURN THEN USING WAS WHILE</code>
Array Types	<code>EMPTY_DATE_NUMBER EMPTY_NUMBER_NUMBER EMPTY_TEXT_NUMBER EMPTY_DATE_TEXT EMPTY_NUMBER_TEXT EMPTY_TEXT_TEXT</code>
Formula Data Types	<code>DATE DATE_NUMBER DATE_TEXT NUMBER NUMBER_NUMBER NUMBER_TEXT TEXT TEXT_NUMBER TEXT_TEXT</code>
Array Methods	<code>COUNT DELETE EXISTS FIRST LAST NEXT PREVIOUS PRIOR</code>
Built-in Calls	<code>CONTEXT_IS_SET EXECUTE</code>

Source of Reserved Words	Reserved Words
	GET_CONTEXT GET_OUTPUT IS_EXECUTABLE SET_INPUT WSA_DELETE WSA_EXISTS WSA_GET WSA_SET

*Related Topics*

- [Overview of Using Fast Formula Components](#)

## Database Items

Database items exist in the application database and have a computer code associated with them. The application uses this code to find data. All database items are read-only variables.

You can't change database item values within a formula. If you try to write a value to a database item, you receive a compilation error.

You can use database items in formulas and HCM extracts.

Database items are of these types:

- Static
- Dynamic

### Static Database Items

Static database items are predefined. They include generic information about an employee, such as sex, birth date, and work location. They also include data about other objects, such as the start and end dates of a payroll period.

### Dynamic Database Items

The application creates dynamic database items when you create these objects. In the case of flexfields, you must run the Generate Flexfield Database Items process to create the database items.

Object	Description
Elements	The element name is the database item name prefix.
Defined Balances	The balance name followed by the balance dimension name is the database item name.
Formula global values	The global value name is the database item name.
Input values	The element and input value names are the database item name prefix.

Object	Description
Flexfields	The Generate Flexfield Database Items process creates database items for the contexts and segments of your registered HCM flexfields.

## Array Database Items

Array database items have an index type of NUMBER with indexes that start at 1 and increment by 1 without gaps. Here's an example:

```
/* 1 is the starting index for an array database item. */
I = 1
WHILE DBI.EXISTS(I)  LOOP
(
  V = DBI[I] /* Do some processing with element at index I. */
  I = I + 1 /* Array database items indexes go up in steps of 1. */
)
```

You can use the DEFAULT\_DATA\_VALUE FOR statement to set a default value in scenarios where an array database item returns a NULL value for an element. There can be only one DEFAULT\_DATA\_VALUE FOR statement for each array database item and it must appear at the start of the formula.

Here's an example of a DEFAULT\_DATA\_VALUE FOR statement:

```
DEFAULT_DATA_VALUE FOR A IS 0
INPUTS ARE B, C
```

Here's an example of an array database item usage error:

```
/* Array database item A. */
A[1] = 1
A = B
A.DELETE(1)
A.DELETE
```

### Related Topics

- [Overview of Generating Flexfield Database Items](#)

## Overview of Generating Flexfield Database Items

You configure registered HCM flexfields to add contexts and segments for your business requirements.

After you deploy the flexfield, you can generate database items for the flexfield for use in your formulas and extracts by submitting the Generate Flexfield Database Items process from the Payroll Checklist or Payroll Administration work areas.

You can generate database items for the following flexfields:

- Descriptive flexfields
- Extensible flexfields for single and multiple row routes
- Key flexfields

The process generates database items at the enterprise level only. As a best practice, when you submit the process, skip the legislative data group parameter so that the process generates database items for use by any legislative data group.

You can determine which database items to generate by specifying or skipping the flexfield and context parameters as shown in the following table.

Flexfield Parameter	Context Parameter	Result
Specify parameter	Skip parameter	Generate database items for all the contexts and related segments for a specified flexfield
Skip parameter	Skip parameter	Generate database items for all registered flexfields and their contexts.

The process creates database item names with this following structure:

```
de<FLEXFIELD_CODE><CONTEXT_CODE><SEGMENT_CODE>
```

When you include the database item in a formula or extract, the application returns a value for the database item, based on the flexfield context, for the segments column in the underlying flexfield table. After you generate database items, compile any formulas using these database items.

Periodically, you may need to update a flexfield structure, for example to add a segment to capture additional data. If you previously generated database items for a flexfield, submitting the process deletes and regenerates the associated database items. After the process regenerates the database items, be sure to compile any formulas using them.

#### Related Topics

- [Database Items](#)
- [Extract Components](#)
- [Example of Generating Flexfield Database Items](#)

## Example of Generating Flexfield Database Items

Let's look at an example that shows you how to add segments and contexts to a registered HCM flexfield, and how to generate database items for the flexfield for later use in formulas and extracts.

Here are the key decisions when deciding on the HCM flexfields and contexts that require database items.

Decisions to Consider	In this Example
Which registered HCM flexfield requires database items?	Organization Information flexfield
What is the name of the flexfield code?	PER_ORGANIZATION_INFORMATION_EFF

Decisions to Consider	In this Example
How many contexts include segments that require database items?	Two contexts: HCM_CN_PSU_TERMINATION_INFO, HRX_CN_TRU_TERMINATION_INFO
Should I give all legislative data groups access to the generated database items, so that they can use them in their formulas and extracts?	Yes

In this example, Joe plans to create formulas for an implementation in China based on information captured in the organization information flexfield. Joe configures the flexfield to add the contexts and segments. He writes a formula to calculate severance pay that returns results for the leave compensation factor based on the tax reporting unit.

## Configure the Flexfield

1. In the Setup and Maintenance work area, navigate to:
  - o Offering: Workforce Deployment
  - o Functional Area: Organization Structures
  - o Task: Manage Extensible Flexfields
2. Search for the Organization Information EFF and click **Edit**.
3. Click **Manage Contexts**.
4. Click **Create** and create two contexts: HCM\_CN\_TRU\_TERMINATION\_INFO and HRX\_CN\_PSU\_TERMINATION\_INFO.
5. For each context, create two segments: LEAVE\_COMPENSATION\_FACTOR and MONTHLY\_SALARY\_PAID\_DAYS.
6. Deploy the flexfield.

## Submit the Generate Flexfield Database Process

1. In the Payroll Checklists or Payroll Administration work area, select the **Submit a Process or Report** task.
2. Complete the following parameters.

This table lists which parameters you can skip and those that you can complete, and where the parameter is displayed.

Page	Parameter	Data
Select Flow Pattern	Legislative Data Group	Skip this parameter
Select Flow Pattern	Process or Report	Generate Flexfield Database Items
Enter Parameters	Payroll Flow	Organization flexfield database items
Enter Parameters	Name	PER_ORGANIZATION_INFORMATION_EFF

Page	Parameter	Data
Enter Parameters	Context	Skip this field to generate database items for all contexts.

3. Click **Next** and skip the Enter Interaction page.
4. Click **Next** and skip the Schedule page.
5. Click **Next** and review the submitted parameters on the Review page.
6. Click **Submit** to create the payroll flow.

The submitted process creates database items for each context for the flexfield segments:

- o PER\_ORGANIZATION\_INFORMATION\_HRX\_CN\_TRU\_TERMINATION\_LEAVE\_COMPENSATION\_FACTOR
- o PER\_ORGANIZATION\_INFORMATION\_HRX\_CN\_TRU\_TERMINATION\_MONTHLY\_SALARY\_PAID\_DAYS
- o PER\_ORGANIZATION\_INFORMATION\_HRX\_CN\_PSU\_TERMINATION\_LEAVE\_COMPENSATION\_FACTOR
- o PER\_ORGANIZATION\_INFORMATION\_HRX\_CN\_PSU\_TERMINATION\_MONTHLY\_SALARY\_PAID\_DAYS

## Create a Formula

1. Create a formula for calculating severance pay that returns a segment for LEAVE\_COMPENSATION\_FACTOR, if the context code is the one specified for the database item, HRX\_CN\_TRU\_TERMINATION\_INFO.

## Formula Operators

Formula operators are expressions that can contain arithmetic operators. These operators determine how the formula manipulates variables and literals. For example, the plus operator (+) indicates that two items are added together. You can also use operators to concatenate strings.

## Types of Operators

Here's a list of the operator types:

Operator	Description	Example
+	Addition	$A = B + 1$
+	String concatenation	$A = 'Hello ' + 'World'$ $B = 'Hello '    'World'$
-	Subtraction	$A = B - 1$
-	Unary minus	$A = -B$

Operator	Description	Example
*	Multiplication	$A = B * C$
/	Division	$A = B / C$

## Using Operators

You can use the arithmetic operators, subtraction, multiplication, and division only with numeric operands. You can use the addition operator with numeric or text operands. The operands can be variables, literals, or sub-expressions. A formula error occurs in these scenarios:

- The second operand of a division equals zero
- The result of multiplication is too large

Whether a number is too large or not is determined by the normal limits in the database. For string concatenation, if the result is longer than 255 characters, a formula error occurs.

Expressions are evaluated in order from left to right. The unary minus has precedence over the other operators because it applies directly to a single sub-expression. The multiplication and division operators take precedence over addition and subtraction. For example, the expression  $1 + 2 * 3$  evaluates to 7 rather than 9. You can use brackets to change the precedence of the operators. For example,  $(1 + 2) * 3$  evaluates to 9.

### Related Topics

- [Overview of Using Fast Formula Components](#)

## Literals

A literal is a piece of information that you manipulate or use in a formula.

Literals are of these types:

- Numeric
- Text
- Date
- Array

## Numeric Literals

Follow these rules when entering numeric literals:

- Don't use quotes to enclose the literal.
- Don't use commas or spaces in the number.
- Don't use exponents and floating point scientific notations.
- You can enter numbers that have a decimal component after a decimal point.

- To enter a negative number, precede it with a minus sign (-).

Here are a few examples of numeric literals:

- 63
- 3.55
- -2.3
- -.033
- -.2
- 10000

## Text Literals

When you enter text literals, enclose them in single quotes. Text literals can contain spaces. To enter a single quote character in a text constant, enter two single quotes (for example, 'P O'Donnell'). Note that this isn't the same as the double quotes ("").

Here are a few examples of text literals:

- 'J. Smith'
- '1234'
- 'Manager'
- '12 Union Road'
- 'The Bonus this year is 23%'

## Date Literals

When you enter a date literal, enclose dates in single quotes and follow immediately with the word date in brackets.

Here are a few examples of date literals:

- '2010-11-04T00:00:00.000Z' (DATE)
- '1989-03-12 00:00:00' (DATE)
- '12-MAR-1989' (DATE)

## Array Literals

An array holds multiple values that the formula can access using the corresponding index values. You define array literals only for an empty array of each type.

Array literals are of these types:

- Array of date values indexed by a numeric index (EMPTY\_DATE\_NUMBER)
- Array of number values indexed by a numeric index (EMPTY\_NUMBER\_NUMBER)
- Array of text values indexed by a numeric index (EMPTY\_TEXT\_NUMBER)
- Array of date values indexed by a text index (EMPTY\_DATE\_TEXT)
- Array of numeric values indexed by a text index (EMPTY\_NUMBER\_TEXT)
- Array of text values indexed by a text index (EMPTY\_TEXT\_TEXT)

# How Formulas Determine Variable Data Types

Variables can be of these data types: numeric, text or date. The data type determines the type of information the variable holds.

You don't have to specify the variable type. Formulas determine the type based on how you use the variable. For example, if you set a variable to 'J. Smith', the formula interprets it as a text variable.

**CAUTION:** Inconsistent or incorrect use of variables, such as trying to add a number to a text string, causes formula compilation errors.

## Determine Variable Data Types

Formulas process the rules that determine the variable data type in the order listed:

1. The variable can be an input you name in the input statement. For example:

```
INPUTS ARE SALARY_AMOUNT,  
START_DATE (DATE),  
FREQUENCY (TEXT)
```

If you don't specify the variable data type in the statement, the formula assumes it's a number.

You can use the DEFAULT\_FOR statement to determine the variable data type:

```
DEFAULT FOR B IS 0 /* B is a NUMBER variable. */
```

For an array database item, the DEFAULT FOR statement determines the index type and value type:

```
DEFAULT FOR A IS EMPTY_NUMBER_NUMBER /* A is a NUMBER_NUMBER array variable. */
```

You can use the DEFAULT\_DATA\_VALUE FOR statement to determine the variable data type:

```
DEFAULT_DATA_VALUE FOR A IS 0 /* A could be a NUMBER_NUMBER or NUMBER_TEXT variable. */
```

2. The formula searches the list of database items. If the variable is in the list, the data type is known.
3. If the variable appears in a context handling statement, then the formula searches the list of contexts. If the variable is in the list, then the formula knows the data type, otherwise it returns an error.
4. If the variable isn't a database item or a context, then the formula treats it as a local variable and determines the data type based on how you use the variable. For example:

```
A = 'abc' /* A is a TEXT variable. */
```

# Array Variables

You can use arrays for input, output, and local formula variables. These array variables can store date, number, or text values. Arrays are similar to PL/SQL index-by tables.

## Array Indexes

Here are some aspects of array indexes that you should be aware of:

- The index types are either text or number.
- Text indexes are upper case unique.
- Gaps in index value sequences are permitted.
- Number indexes are truncated to remove any fractional part.
- You may iterate an array in an index either forwards or backward.

## Array Data Types

You specify array types as DATA\_TYPE\_INDEX\_TYPE. Arrays are of these data types:

- NUMBER\_NUMBER
- NUMBER\_TEXT
- DATE\_NUMBER
- DATE\_TEXT
- TEXT\_NUMBER
- TEXT\_TEXT

## Rules for Using Arrays

Formula functions can't return arrays or take array parameters. Contexts can't be array types. If you try to reference an array value at a nonexistent index, the application returns a formula execution error. However, you can delete an array value at a nonexistent index.

## Array Methods

Array methods enable you to get the first and last indexes, and to get the next or prior index. These methods return the index data type. You can specify a default value for these methods, if the required indexes don't exist. You can use array methods to determine whether an index exists or not.

**Note:** The array method syntax doesn't work directly with the array literal values. For example, you can't use a construct such as EMPTY\_DATE\_NUMBER.COUNT.

Here's a list of the array methods, including their descriptions and usage examples:

Array Method	Description	Usage Example
<name> [ <index value> ]	Get the value for an index.	V = A[1]
<name> . FIRST( <default value> )	Get the first index for an array. The default value is returned if the array is empty.	I = A.FIRST(-1)
<name> . LAST( <default value> )	Get the last index for an array.	L = B.LAST(' ')
<name> . EXISTS( <index value> )	Conditional checking if a value exists at an index. The default value is returned if the array is empty.	IF A.EXISTS(1) THEN
<name> . NEXT( <index value> , <default index value> )	Get the next index given an index position. The default value is returned if there is no next index.	N = A.NEXT(1)
<name> . PRIOR( <index value> , <default index value> )	Get the prior index given the index position. The default value is returned if there is no prior index.	P = B.PRIOR('Two')
<name> , COUNT	Numeric method to count the array elements.	C = A.COUNT
<name> , DELETE( <index value> )	Delete the element at an index position.	B.DELETE('three')
<name> , DELETE()	Delete all elements.	B.DELETE()

## Iterating Through an Array

In this example, *A* is an array variable with a NUMBER index. -1234 is known to be an invalid index for *A*, so it's used as a default value when the FIRST and NEXT calls can't find an index.

```
/* -1234 is not a valid index for A in this instance, so use as default. */
NI = A.FIRST(-1234)
WHILE A.EXISTS(NI) LOOP
(
  VA = A[NI] /* Do some processing with element at index NI. */
  NI = A.NEXT(NI,-1234) /* Go to next index. */
)
```

In this example, *B* is an array variable with a TEXT index. -1234 is known to be an invalid index for *B*, so it's used as a default value when the FIRST and NEXT calls can't find an index.

```
/* 'No Index' is not a valid index for B in this instance, so use as default. */
TI = B.FIRST('No Index')
WHILE B.EXISTS(TI) LOOP
(
  VB = B[TI] /* Do some processing with element at index TI. */
  TI = B.NEXT(TI, 'No Index') /* Go to next index. */
)
The following example iterates backwards from through an array C with a NUMBER index.
/* -1234 is not a valid index for C in this instance, so use as default. */
NI = C.LAST(-1234)
```

```
WHILE C.EXISTS(NI) LOOP
(
  VC = C[NI] /* Do some processing with element at index NI. */
  NI = C.PRIOR(NI,-1234) /* Go to prior index. */
)
```

*Related Topics*

- [Overview of Using Fast Formula Components](#)

## Formula Contexts

Formulas run within an application-specific execution context, which determines the context variables available to the formula. Context values act as SQL bind values when the formula fetches database item values from the database. Formulas can also pass context values into formula function calls.

Here are some examples of the execution contexts:

- **EFFECTIVE\_DATE**: The date on which the formula becomes effective.
- **PAYROLL\_ID**: The payroll run for which the formula is processed.
- **PERSON\_ID**: The person for whom the formula is processed.

## Context Value Setting

The application code that's calling a formula usually sets all the context values. For some complex applications, such as the payroll run, the code sets only the contexts necessary to meet general processing requirements.

Here's how formulas for payroll work:

- A payroll run sets contexts for the legislative data group, date earned, the payroll being processed, the payroll relationship, payroll actions, and the person being processed.
- Additional, country-specific contexts are also available. For example, the jurisdiction area and tax code context values are country-specific.

## Formula Context-Handling Statements

If you use a variable in a context-handling statement, the formula searches the list of contexts. The variable must appear in the contexts list; otherwise the formula returns an error. The data type is stored along with the context list entry.

This table describes the formula context-handling statements.

Statement	Purpose of the Statement	Example
<code>CHANGE_CONTEXTS (assignment [, ...])</code>	Changes context values within the context changing block. Inside this block, the formula function calls, the database items, and the called formulas use the new context values. Outside the block, the formula uses the original values.	<pre>/*  * Nested Context changes: DB1  depends upon SOURCE_ID and SOURCE_ TEXT. */ CHANGE_CONTEXTS(SOURCE_TEXT = 'A') ( /* SOURCE_TEXT = 'A' */</pre>

Statement	Purpose of the Statement	Example
	You can nest context changing blocks to apply context changes in stages.	<pre> X = DBI1  /* Nesting used to change Contexts in stages. */ CHANGE_CONTEXT(SOURCE_ID = 2) ( /* SOURCE_TEXT = 'A', SOURCE_ID = 2 */ Y = DBI1  /* Overriding a Context change. */ CHANGE_CONTEXTS(SOURCE_TEXT = 'B', SOURCE_ID = 3) ( /* SOURCE_TEXT = 'B', SOURCE_ID = 3 */ Z = DBI1 ) ) </pre>
CONTEXT_IS_SET(context)	Tests whether or not a context value is set.	<p>This code tests whether or not the AREA3 context is set.</p> <pre> IF CONTEXT_IS_SET(AREA3) THEN </pre>
GET_CONTEXT(context, default value)	Returns a context's value if the context is set, otherwise it returns the default value specified in its second argument.	<pre> /* AREA1 is a context of type TEXT. */ AREA1_VALUE = GET_CONTEXT(AREA1, ' ') </pre>

## Overview of Working Storage Area

The working storage area is a mechanism for storing global values across formulas.

Using the different call methods, you can test whether or not an item exists in the storage area, delete an item, set the value for an item, and get a value for an item. You can access the values by name. The names are case-independent.

This table lists the working storage area methods.

Method	Description
WSA_EXISTS(item [, type])	<p>Tests whether or not the item called <code>item</code> exists in the storage area. If <code>type</code> is specified, then the item must be of the same type. These are the valid values for <code>type</code>:</p> <ul style="list-style-type: none"> <li>• <code>DATE</code></li> <li>• <code>DATE_NUMBER</code></li> <li>• <code>DATE_TEXT</code></li> <li>• <code>NUMBER</code></li> <li>• <code>NUMBER_NUMBER</code></li> <li>• <code>NUMBER_TEXT</code></li> </ul>

Method	Description
	<ul style="list-style-type: none"> <li>• <code>TEXT</code></li> <li>• <code>TEXT_NUMBER</code>,</li> <li>• <code>TEXT_TEXT</code></li> </ul>
<code>WSA_DELETE([item])</code>	Deletes the item called <code>item</code> . If you don't specify a name, then all the storage area data is deleted.
<code>WSA_SET(item, value)</code>	Sets the value for the item called <code>item</code> . Any existing item of the same name is overwritten.
<code>WSA_GET(item, default-value)</code>	Retrieves a value for the item called <code>item</code> . If there is no item called <code>item</code> , then the method returns the default value. The data type of <code>default-value</code> is the expected data type for <code>item</code> .

## Overview of Calling a Formula from a Formula

A formula can call another formula. This enables some modularity in formula organization. You specify the called formula name and any formula input or output names as TEXT values. The names are case-independent.

You can call a formula in any of these ways:

- Using a single call
- Using separate calls

As you write your formula, consider these points:

- Validation of the Called Formula
- Passing Contexts
- Call a Formula using any of these methods:
  - Using Separate Calls
  - Using a Single Self-Contained Call
- Use Cases to Compare Methods

## Validation of the Called Formula

When a formula runs, it validates whether the called formula can be run, and whether the specified input and output data types are correct. You can use the `IS_EXECUTABLE` call to determine whether an executable formula with a specified name exists. You must compile the formula to make it available for the specified legislative data group. Also, the formula must be valid on the effective date on which the calling formula runs. In addition, the payroll code imposes extra restrictions based on formula type combinations.

## Passing Contexts

Context values are inherited from the calling formula. You can also set or unset the context values explicitly in the nested formula call.

## Alternative Methods to Call a Formula

You can call a formula from within another formula in any of these ways:

- Using a series of separate calls
- Using a single self-contained call

## Using Separate Calls

You can use three separate calls:

### 1. Set the Inputs

Use the `SET_INPUT` call for each formula input and context that you need to explicitly set for the formula call. You don't need to specify all formula inputs and contexts. To explicitly unset a context value, use the `SET_INPUT` call without passing the optional value parameter. Any extra inputs specified in `SET_INPUT` calls are ignored.

### 2. Call the Formula

Use the `EXECUTE` call to call a formula.

### 3. Get the Formula Outputs

Use one or more `GET_OUTPUT` calls to fetch outputs from the last formula call.

When the formula runs, it returns an execution error in these scenarios:

- The called formula isn't executable.
- The called formula is already running.
- The data type of an input variable (specified using `SET_INPUT`) or an output variable (specified using `GET_OUTPUT`) doesn't match its actual data type within the called formula.

This table summarizes the methods for calling a formula using separate calls.

Method	Use To
<code>SET_INPUT(input [,value])</code>	Sets the value of an input or a context to the value specified in the value parameter. The value parameter is optional. If you specify the value parameter, the input is set to this value. If you don't specify the value parameter, the input or context is passed as unset to the formula. The data type of the value is the expected data type for the input.  If you use a context in a <code>SET_INPUT</code> call, its value will be set to the value in the calling formula.
<code>EXECUTE(formula)</code>	Executes the called <code>formula</code> .
<code>GET_OUTPUT(output, default-value)</code>	Gets the value of the output parameter after calling a formula. If there is no formula output called 'output' or it's not set, the formula returns the value specified in the default value parameter. The data type of default value is the expected data type for the output.

**Note:** Formula inputs that are set using `SET_INPUT` persist as long as no `EXECUTE` or `GET_OUTPUT` calls are made. Output values from a called formula persist as long as no `SET_INPUT` or new `EXECUTE` calls are made. When the calling formula exits, the process removes any saved input or output values.

## Using a Single Self-Contained Call

Here's what a single self-contained call does:

- Clears the input values at the start to not use prior `SET_INPUT` call values.
- Discards the outputs at the end so that the subsequent `GET_OUTPUT` calls only return the default values.

Here's how you can use the `CALL_FORMULA` method:

```
CALL_FORMULA(formula, [set statement, get statement])
```

A `SET` statement is a `SET_INPUT` call. A `GET` statement assigns the result of a `GET_OUTPUT` call to a variable in the calling formula. The calls execute in this order:

1. `SET_INPUT` calls
2. EXECUTE call
3. `GET_OUTPUT` assignments

The compiler generates code to execute in this order, even if SET and GET statements are interspersed.

### Related Topics

- [Examples of Calling a Formula from a Formula](#)

## Examples of Calling a Formula from a Formula

Use these examples to understand how to call a formula from another formula using these methods:

- A series of separate calls
- A single self-contained call

The examples include two versions of a wage formula, and a table comparing the two methods using a few use cases.

The first two examples show different versions of the wage formula. These points apply to both examples:

- The formula calls `RATE_FORMULA` to get a value for `HOURLY_RATE`.
- The `RATE_FORMULA` enters text to call `UNIT`.
- The formula call sets to enter the `UNIT` to 'Hourly'.
- The `RATE_FORMULA` returns the rate in the output variable called `RATE`.
- The `GET_OUTPUT` call returns 0.0 if the `RATE_FORMULA` doesn't return `RATE`.

### Wage Formula Using Separate Calls

This example shows how to call a formula using separate calls.

```
SET_INPUT('UNIT', 'Hourly')
EXECUTE('RATE_FORMULA')
HOURLY_RATE = GET_OUTPUT('RATE', 0.0)
WAGE = HOURS_WORKED * HOURLY_RATE
RETURN WAGE
```

## Wage Formula Using a Self-Contained Call

This example shows how to call a formula using a self-contained call.

```
CALL_FORMULA
('RATE_FORMULA','Hourly' > 'UNIT'
/* SET_INPUT('UNIT', 'Hourly') */
,HOURLY_RATE < 'RATE' DEFAULT 0.0
/* HOURLY_RATE = GET_OUTPUT('RATE',0.0) */
)
WAGE = HOURS_WORKED * HOURLY_RATE
RETURN RATE
```

## Sample Expressions to Compare Methods

Here's what you can do using sample expressions in the two methods:

- Call a formula
- Set inputs and context values
- Unset context values
- Get output values into a variable or array variable
- Provide a default output value

**Note:** SET\_INPUT or > statements have no effect if the calling formula has no formula input or context of the same name.

This table shows a few use cases that compare the two methods using sample expressions.

Use Case	Using Separate Calls	Using a Self-Contained Call
Execute a formula where the formula <b>GET_RATES</b> is executed	<b>EXECUTE ('GET_RATES')</b>	Use within a <b>CALL_FORMULA</b> statement <b>'GET_RATES'</b>
Set an input value in the called formula where you round off EXTRA_HOURS to 2 decimal places and set the input OVERTIME in the called formula.	<b>SET_INPUT ('OVERTIME', ROUNDUP(EXTRA_HOURS, 2))</b>	Use within a <b>CALL_FORMULA</b> statement <b>ROUNDUP(EXTRA_HOURS, 2) &gt; 'OVERTIME'</b>
Leave a formula input value unset inside the called formula, where <b>RATE</b> isn't a formula context.	A <b>SET_INPUTS</b> statement isn't required, but you can this:  <b>SET_INPUT ('RATE')</b>	A <b>SET</b> statement isn't required, but you can use this:  <b>&gt; 'RATE'</b>
Inherit a context value from the called formula. For example, both the calling formula and the called formula support the AREA1 context. The called formula inherits the AREA1 context value from the calling formula.	No statements are required to do this.	No statements are required to do this.
Set a context value inside a called formula, where the called formula supports the AREA1	<b>SET_INPUT</b>	<b>'London' &gt; 'AREA1'</b>

Use Case	Using Separate Calls	Using a Self-Contained Call
context and you must set AREA1 to 'London' in the called formula.	( 'AREA1' , 'London' )	
Call a formula with an unset context value, where the called formula supports the AREA1 context and AREA1 has to be unset in the called formula.	SET_INPUT('AREA1')	> 'AREA1'
Get a formula output from the called formula.  Get BONUS_RATE output value into the RATE variable using the default value 0.0 if the BONUS_RATE output doesn't exist or wasn't set.	RATE = GET_OUTPUT ( 'BONUS_RATE' , 0.0 )	RATE <'BONUS_RATE' DEFAULT 0.0
Get a formula output from a called formula into an array  Get the BONUS_RATE output value into the RATES array variable at index position 'BONUS'. Use the default value 0.0 if the BONUS_RATE output doesn't exist or wasn't set.	RATES[ 'BONUS' ] = GET_OUTPUT ( 'BONUS_RATE' , 0.0 )	RATES[ 'BONUS' ] <'BONUS_RATE' DEFAULT 0.0

# 4 Fast Formula Functions

## Overview of Fast Formula Functions

Fast formula functions manipulate data in different ways and always return a value. Functions are of these data types: date, number, text, and array (date\_number, number\_number, text\_number, date\_text, number\_text, and text\_text). Function operands are of these data types: date, number, and text. A fast formula function is identified by its name, return data type, data types, and usage behavior.

Here's how the general form of a fast formula function looks like:

**NAME-OF-FUNCTION (operand, operand, ...)**

Operands can be optional or mandatory. You can use operands any number of times in a statement; for example, with the **GREATEST** function. The formula compiler resolves functions by matching function calls against function specifications. You can use multiple functions with the same name within a fast formula, provided that they have different return or data types.

Some fast formula functions return values that are useful in specific formula types, such as absence management, benefits, or compensation. The more generic functions fall into these categories:

Category	Purpose of Functions
Text	Manipulate text data
Number	Manipulate numeric data
Date	Manipulate dates
Data Conversion	Convert data to a different data type
Message Handling	Return messages

## Text Formula Functions

Here are the formula functions that you can use to manipulate text data:

### CHR(n)

Returns the character having the binary equivalent to a number operand **n** in the ASCII character set.

## **GREATEST(expr, expr [,expr]....)**

Compares the values of all the text string operands. It returns the value of the last string in alphabetic order.

## **INITCAP(expr)**

Returns the expression `expr` with the first letter of each word in uppercase. Delimits the words or characters that aren't alphanumeric by a white space.

## **INSTR(expr1, expr2 [,n [,m]])**

Searches `expr1` beginning with its `n`th character for the `m`th occurrence of `expr2` and returns the character position in `expr1` for the first character of this occurrence. If `n` is negative, `INSTR` counts and searches backward from the end of `expr1`. The value of `m` must be positive. The default values of both `n` and `m` are 1, meaning `INSTR` begins searching at the first character of `expr1` for the first occurrence of `expr2`. The return value is relative to the beginning of `expr1`, regardless of the value of `n`, and is expressed in characters. If the search is unsuccessful (`expr1` doesn't appear `m` times after the `n`th character of `expr1`), the return value is 0.

## **INSTRB(expr1, expr2 [,n [,m]])**

Works in the same way as `INSTR`, except that `n` and the return values are expressed in bytes, rather than in characters. For a single-byte character set, `INSTRB` is equivalent to `INSTR`.

## **LEAST(expr, expr [,expr]...)**

Compares the values of all the text string operands. Returns the first string in alphabetic order from among its operands.

## **LENGTH(expr)**

Returns the number of characters in the text string operand `expr`.

## **LENGTHB(expr)**

Returns the length of `expr` in units of bytes.

## **LOWER(expr)**

Converts a text string to lowercase.

## **LPAD(expr, n [,pad])**

Returns the text string operand `expr` left-padded to length `n` with the sequence of characters in `pad`. The default value for `pad` is a blank. If `expr` is longer than `n`, then `LPAD` returns the portion of `expr` that fits in `n`.

These are a few examples of this expression:

```
/* A is set to 'XYXYXHello' */
A = LPAD ('Hello', 10, 'XY')
/* A is set to 'hell' */
A = LPAD ('Hello', 4 )
```

## LTRIM(*expr* [,set])

Returns the text string operand *expr* with all the left-most characters that appear in *set* removed. The default for *set* is a blank. If none of the left-most characters of *expr* appear in *set*, then **LTRIM** returns *expr*.

Examples:

```
/* A is set to 'def' */
A = LTRIM ('abcdef', 'abc')
/* A is set to 'abcdef' */
/A = LTRIM ('abcdef', 'bc')
```

## REPLACE(*expr*, *search* [,replacement])

Returns the text string operand *expr* with every occurrence of *search* replaced with *replacement*. If you omit *replacement*, it removes all occurrences of *search*. Use **REPLACE** to substitute one string for another or to remove character strings.

Example:

```
/* Set A to 'BLACK and BLUE'. */
A = REPLACE('JACK and JUE', 'J', BL')
```

## RPAD(*expr*, *n* [,pad])

Returns the text string operand *expr* right-padded to length *n* with the sequence of characters in *pad*. The default value for *pad* is a blank. If *expr* is longer than *n*, then **RPAD** returns the portion of *expr* that fits in *n*.

Examples:

```
/* A is set to 'helloXYXX' */
A = RPAD ('hello', 10, 'XY')
/* A is set to 'hell' */
A = RPAD ('hello', 4 )
```

## RTRIM(*expr* [,set])

Returns the text string operand *expr* with all the right-most characters that appear in *set* removed. The default value for *set* is a blank. If none of the right-most characters of *expr* appear in *set*, then *expr* is returned.

Examples:

```
/* A is set to 'abc' */
A = RTRIM ('abcdef', 'def')
/* A is set to 'abcdef' */
A = RTRIM ('abcdef', 'de')
```

## SUBSTR(*expr*, *m* [,*n*]) or SUBSTRING(*expr*, *m* [,*n*])

**SUBSTRING** returns a sub string of the text string operand *expr* of length *n* characters beginning at the *m*th character. If *n* is negative, **SUBSTR** counts backward of *expr*. If you omit the *n*, the sub string starts from *m* and finishes at the end of *expr*.

Example:

```
/* Check that the tax code starts with GG */
IF length(Tax_code) <= 2
THEN
  (message = 'Tax code is too short'
  RETURN message
)IF substr( Tax_code, 1, 2) = 'GG' THEN ...
```

## SUBSTRB(*expr*, *m* [*n*])

The same as `SUBSTR`, except that the arguments *m* and *n* are expressed in bytes, rather than in characters. For a single-byte database character set, `SUBSTRB` is equivalent to `SUBSTR`.

## TRANSLATE(*expr*,*from*,*to*)

Returns the text string operand *expr* with all occurrences of each character in *from* replaced by its corresponding character in *to*. Characters in *expr* that aren't in *from* aren't replaced. The argument *from* can contain more characters than *to*. In this case, the extra characters at the end of *from* have no corresponding characters in *to*. If these extra characters appear in *expr*, they're removed from the return value.

## TRIM(*expr*)

Trims leading and trailing spaces from a character string.

## UPPER(*expr*)

Converts a text string to uppercase.

# SPLIT FUNCTIONS

These Text Formula Functions return TEXT\_NUMBER arrays. The arrays contain the result of splitting the string by a separator.

## SPLIT\_BY\_SPACES

The separator is a sequence of one or more consecutive spaces. Here "space" includes tab character, newline in addition to space ''.

### SPLIT\_BY\_SPACES(*expr*)

Here's an example of SPLIT\_BY\_SPACES:

```
SPLIT_BY_SPACES(' Hello World !! ')
```

The return value is an array containing 'Hello', 'World', '!!' in that order.

## SPLIT

### SPLIT(*expr*, *separator*, *default\_string*)

Splits *expr* into strings, separated by *separator*, and returns the strings in a TEXT\_NUMBER array. If there is nothing between two consecutive separators then *default\_string* is returned in the array.

#### SPLIT Example 1:

```
SPLIT('Hello|World', '|', '_')
```

The return value is an array containing 'Hello', 'World' in that order.

#### SPLIT Example 2 Multiple Character Separator:

```
SPLIT('Hello||World', '||', '_')
```

The return value is an array containing 'Hello', 'World' in that order.

### **SPLIT Example 3 Showing default string use:**

```
SPLIT('|Hello||World|', '|', '_')
```

The return value is an array containing '\_', 'Hello', '\_', 'World', '\_' in that order. The default string, '\_', replaces "gaps" between separators i.e. before first '|', after second '|' and before 3rd '|', and after last '|'.

### **SPLIT Example 4 Input String Without Separator:**

```
SPLIT('Hello', '|', '_')
```

The return value is an array just containing 'Hello'.

## Numeric Formula Functions

Here are the formula functions that you can use to manipulate numeric data:

### **ABS(n)**

Returns the magnitude of a numeric operand  $n$  as a positive numeric value. If the value of the operand is positive, its value returns unchanged. If the operand is negative, then the value's sign inverts and the value returns as a positive number.

Example:

```
ABS (-17)
```

It returns 17.

### **FLOOR(n)**

Returns the integer part of a numeric operand  $n$ . If the value of the operand contains information after the decimal point, **FLOOR** discards that information and returns a whole number.

Example:

```
FLOOR (35.455)
```

It returns 35.

### **GREATEST(n, n [, n] ...)** or **GREATEST\_OF(n, n [, n] ...)**

Compares all the operands and returns the greatest value.

### **LEAST(n, n [, n] ...)** or **LEAST\_OF(n, n [, n] ...)**

Compares all the operands and returns the smallest value.

### **MOD(m, n)**

Returns the remainder after dividing  $m$  by  $n$ .

## POWER(m, n)

Returns  $m$  raised to the  $n$ th power.

## ROUND(m [,n])

Rounds  $m$  to  $n$  decimal places. The default number of decimal places is 0.

Examples:

```
ROUND (2.3401, 2)
```

It returns 2.34.

```
ROUND (2.3461, 2)
```

It returns 2.35.

## ROUNDUP(m [,n]) or ROUND\_UP(m [,n])

Rounds off  $m$  up to  $n$  decimal places. The default number of places is 0.

Examples:

```
ROUND_UP (2.3401, 2)
```

It returns 2.35.

```
ROUND_UP (2.3400, 2)
```

It returns 2.34.

## TRUNC(n [,m]) or TRUNCATE(n [,m])

Truncates  $m$  down to  $n$  decimal places. The default number of places is 0.

Examples:

```
TRUNC (2.3401, 2)
```

It returns 2.34.

# Date Formula Functions

Here are the formula functions that you can use to manipulate date data:

## ADD\_DAYS(date, n)

Adds  $n$  whole days to `date`.

Example:

```
ADD_DAYS ('30-DEC-1990' (date), 6)
```

It returns 5 JAN 1991.

## ADD\_MONTHS(date, n)

Adds *n* whole months to *date*.

## ADD\_YEARS(date, n)

Adds *n* whole years to *date*.

## DAYS\_BETWEEN(date1, date2)

Returns the number of days between *date1* and *date2*. If *date1* is later than *date2*, then the result is a positive number. If *date1* is earlier than *date2*, then the result is a negative number.

Here's an example of this function:

```
DAYS_BETWEEN('1995/06/27 00:00:00' (date), '1995/07/03 00:00:00' (date))
```

It returns -6.

```
Similarly;
DAYS_BETWEEN('2016/12/31 00:00:00' (DATE), '2017/01/01 00:00:00' (DATE))
```

It return -1.

## GET\_SYSDATE()

Returns the current system date value according to an internal notion of system date.

## LAST\_DAY(date)

Returns the last day of the month containing *date*.

## LEAST(date, date [, date] ...)

Compares the operands and returns the earliest *date*.

## MONTHS\_BETWEEN(date1, date2)

Returns the number of months between *date1* and *date2*. If *date1* is later than *date2*, the result is a positive number. If *date1* is earlier than *date2*, the result is a negative number. The return value has a numeric data type that can contain a fraction if the dates don't differ by a whole number of months.

## NEW\_TIME(date, zone1, zone2)

Returns the date and time in *zone2* when the date and time in *zone1* are *date*.

Here are the standard text strings that you can store in arguments *zone1* and *zone2*:

Time Zone	Description
AST	Atlantic Standard Time

Time Zone	Description
BST	Bering Standard Time
CST or CDT	Central Standard or Daylight Time
EST or EDT	Eastern Standard or Daylight Time
GMT	Greenwich Mean Time
HST or HDT	Alaska-Hawaii Standard Time or Daylight Time
MST or MDT	Mountain Standard or Daylight Time
NST	Newfoundland Standard Time
PST or PDT	Pacific Standard or Daylight Time
YST	Yukon Standard Time

## NEXT\_DAY(d, *expr*)

Returns the first date following a of the weekday named by *expr*.

You can specify the week day as a string that has a 3-letter prefix or a number. The prefix is case-independent.

Day	Prefix	Number
Monday	MON	1
Tuesday	TUE	2
Wednesday	WED	3
Thursday	THU	4
Friday	FRI	5
Saturday	SAT	6
Sunday	SUN	7

## ROUND(date [,format])

Returns the result of rounding `date` according to `format`. The default format is `DDD`, which represents the nearest day.

## TRUNC(date [,format])

Returns the result of truncating `date` according to `format`. The default format is `DDD`, which represents a whole day.

# Data Conversion Formula Functions

Here are the formula functions that you can use to perform data conversions:

## DATE\_TO\_TEXT(date [,format]), TO\_CHAR(date [,format]), and TO\_TEXT(date [,format])

Converts `date` to a character string in the format specified by `format`. The default format is the application canonical format.

## NUM\_TO\_CHAR(n, format)

Converts the number `n` to a character string in the specified format. This function is equivalent to the SQL `TO_CHAR` function.

## TO\_CHAR(n) and TO\_TEXT(n)

Converts the number `n` to a character string in canonical number format.

## TO\_DATE (expr [, format])

Converts the character string `expr` to a date in the specified format. If no format is specified, then `expr` must be in canonical format.

## TO\_NUMBER(expr) and TO\_NUM(expr)

Converts the character string `expr` to a number. The character string must be in canonical number format. A period is used for the decimal point; for example, 1.234. Negative numbers are preceded with a minus; for example, -1.234.

# Miscellaneous Formula Functions

Here are the formula functions that you can use to manipulate messaging data or retrieve values from user-defined tables:

## ESS\_LOG\_WRITE

ESS\_LOG\_WRITE(log\_message)

Outputs a log message to the Enterprise Scheduler log file.

Refer to **Formula Logging** section in *Improve Formula Performance*, when considering using the ESS\_LOG\_WRITE function. This function should be used sparingly as it can impact system performance.

## GET\_LOOKUP\_MEANING

GET\_LOOKUP\_MEANING(lookup\_type, lookup\_code [, default\_meaning])

Returns the lookup meaning for the lookup type and code. The default meaning can be provided to return a value if the lookup type and code combination isn't valid. An error will be raised if the lookup type and code combination isn't valid, and no default meaning is provided.

## GET\_MESG, GET\_FND\_MESG

GET\_MESG(appname, msgname [, token1, value1] [, token2, value2] [, token3, value3] [, token4, value4] [, token5, value5] )

GET\_FND\_MESG(appname, msgname [, token1, value1] [, token2, value2] [, token3, value3] [, token4, value4] [, token5, value5] )

Returns an expanded version of the application message specified using `appname`, `msgname`, and up to five pairs of message tokens and their corresponding values.

## GET\_RATE

GET\_RATE(from\_currency, to\_currency, rate\_type[,default\_rate])

Returns the rate between the two currencies for a given conversion date and rate type. The return type is NUMBER.

Contexts: EFFECTIVE\_DATE (text) is the conversion date.

Parameters:

- FROM\_CURRENCY (text): An example would be USD.
- TO\_CURRENCY (text): An example would be GBP.
- RATE\_TYPE (text): The source of a currency conversion rate, such as user defined, spot, corporate, or fixed.

**Note:** During journal entry, the conversion rate is provided automatically by the General Ledger based on the selected conversion rate type and currency, unless the rate type is user. For user rate types, you must enter the conversion rate. Define additional rate types as needed. Set your most frequently used rate type as the default.

- DEFAULT\_RATE (number): The rate that's returned if GET\_RATE can't get a rate. A typical value for this would be 1.0.

GET\_RATE calls a General Ledger product API. The API retrieves the rate from the General Ledger daily rates or fixed rates, where appropriate.

## GET\_TABLE\_VALUE

GET\_TABLE\_VALUE(table\_name, column\_name, row\_value [,default\_value])

GET\_TABLE\_VALUE(table\_name, column\_name, row\_value, effective date)

Returns the value of a cell in a user-defined table on the effective date of the session or process. The first three text operands identify the cell. An optional fourth parameter does one of the following, depending on its data type:

- Text: Returns a text default value if no data is found.
- Date: Returns the value of the cell on the specified date.

Here's an example of this function:

GET\_TABLE\_VALUE('WAGE\_RATES', 'Wage\_Rate', Rate\_Code, 'DEFAULT'): Returns the row\_value for Wage Rate or DEFAULT if it doesn't find a row.

## GET\_VALUE\_SET

GET\_VALUE\_SET(value\_set\_code, bind)

Returns the first record of the given value set for the bind passed, whereby the bind needs to be in this format: <separator character> <equal character><first parameter name><equal character><value><Separator character><second parameter name><equal character><value><Repeat the same for more bind parameters>

The return value is up to 100 characters long and is a value set id column value.

GET\_VALUE\_SET\_ID is an alternative name for GET\_VALUE\_SET.

Here's an example of this function:

ASG\_NUMBER=GET\_VALUE\_SET('SAMPLE\_GET\_ASG\_NUM', '|=PERSON\_ID='||POSITION3||') Passes data to a fast formula function to retrieve assignment number given PERSON\_ID whose value is set from the position3 variable.

**Note:** Use the GET\_VALUE\_SET function to retrieve information when a database item isn't available. This function supports Value Set with Validation Type = Table, Value Data Type = Character and no aliases.

## GET\_VALUE\_SET\_VALUE

GET\_VALUE\_SET\_VALUE(value\_set\_code, bind)

The return value is up to 255 characters long and is a value set value column value. The parameters and value set configuration are the same as for GET\_VALUE\_SET.

## GET\_VALUE\_SET\_IDS

GET\_VALUE\_SET\_IDS(value\_set\_code, bind)

GET\_VALUE\_SET\_IDS corresponds to GET\_VALUE\_SET\_ID, but returns a TEXT\_NUMBER array of values instead of a TEXT value. The parameters and value set configuration are the same as for GET\_VALUE\_SET.

GET\_VALUE\_SET\_CODES is an alternative name for GET\_VALUE\_SET\_IDS.

## GET\_VALUE\_SET\_VALUES

`GET_VALUE_SET_VALUES(value_set_code, bind)`

`GET_VALUE_SET_VALUES` corresponds to `GET_VALUE_SET_VALUE`, but returns a `TEXT_NUMBER` array of values instead of a `TEXT` value. The parameters and value set configuration are the same as for `GET_VALUE_SET`.

`GET_VALUE_SET_MEANINGS` is an alternative name for `GET_VALUE_SET_VALUES`.

## HR\_TRACE(`expr`)

Outputs a trace message.

**Note:** It's more efficient to use an application-specific logging function than `HR_TRACE`.

# 5 Formulas for Payroll

## Overview of Payroll Formulas

Here are the tasks you can do using payroll formulas in your application:

- Prorate payroll results
- Convert rate periodicities
- Validate element entry values
- Control preferences for payment methods
- Use HR database items in formulas
- Set default organization payment methods
- Return salary amounts
- Restrict the payroll run process to a specific set of employees
- Control when the application submits the current flow
- Return reference values for comparison in balance exception reports
- Control how the application loads a batch from a file

You can copy and edit the predefined formulas or create your own.

This table describes the types of formulas that you can use within your payroll application.

Formula or Type	Description	Example
Proration Formula	Controls how the payroll calculation prorates an element entry. Proration occurs when the formula encounters an event, such as a change to an element entry value.	Return a proration factor or .25 for absences based on an employee type, such as a system analyst on temporary assignment.
Rate Conversion Formula	Creates a formula to convert rates for proration when you require different values for periodicity than the ones used in the predefined formulas.	Specify a different number of working hours to use when converting annual values into hourly rates, such as 2080 annual hours.
Element Input Validation Formula Type	Here's what you can do with this formula: <ul style="list-style-type: none"><li>• Validate element entry values</li><li>• Provide a default value for an element entry value</li><li>• Calculate entry values based on other entry values</li></ul>	Multiply the pay rate by 1.5 if the employee works more than 40 hours a week. Use this formula at the element or element eligibility level.
User Table Validation Formula Type	Validates entries in user-defined tables.	Return an error message if a user enters a value less than 10 or greater than 50 in the Dues column of the Union Dues user table.

Formula or Type	Description	Example
Flow Schedule Formula Type	Creates a scheduling formula when the scheduling options for submitting a process, report, or flow don't cover your requirements. You can also determine how often the formula submits future instances of the flow.	Load time card batches daily. Or, write a formula to increase the number of batch loads at the end of a payroll period when workers typically submit their time cards.
Payroll User Interface Configuration Formula Type	Create formulas to control preferences for payment methods for the enterprise if you don't want to use the default configuration. After you create your formulas, you attach them to the appropriate rows in the Payroll User Interface Configuration user-defined table.	Set the default organization payment method for each legislative data group.
Payroll Access to HR Formula Type	Calls a payroll formula when your payroll formula requires an HR database item.	Return salary amounts and then use the Payroll Access to HR formula to return a full-time equivalent (FTE) database item for a specific group of employees.
Payroll Relationship Group Formula Type	Returns Yes or No values to indicate whether a person is part of a payroll relationship group.	Restrict the payroll run process to a specific set of employees, based on assignment and person level attributes.
Flow Schedule Formula Type	Controls when the application submits the current flow and how often it submits future instances of the flow.	Create a formula that schedules the frequency with which an extract process checks for new starter details.
Balance Exception Formula Type	Creates formulas to return reference values for comparison in balance exception reports.	Multiply the year-to-date gross earnings by 5 percent and return the value when the balance exception report to which it's associated is run.
Element Skip Formula Type	Uses conditional processing to define when the run should process the element, and when it should skip it. If you specify a skip rule for the element, payroll runs process the element only when the conditions of the formula are met. Otherwise, it skips the element. You select skip rules on the Manage Elements page.	Specify to process the Union Fees element in every run, unless the Union_Fees_Paid balance is greater than 10,000.

## Create Conversion Formulas for Proration

The predefined proration formula GLB\_EARN\_PRORATION controls how the payroll calculation prorates an element entry when it encounters an event. This could happen when there's a change to an element entry value.

You can copy and edit a predefined proration formula to modify the calculation. Then, you can select the user-defined formula as the proration formula for your element.

## Create a Formula

You must create a modified rate conversion before you create its related proration conversion rule.

1. Select the formula type **Payroll Run Proration**.
2. Search for and copy the predefined **Rate Conversion Proration** formula.
3. Add the suffix underscore **\_PRORATE** to the name.
4. Update these formula inputs:
  - PRORATE\_START\_DATE (date)
  - PRORATE\_END\_DATE (date)
  - SOURCE\_PERIODICITY (text)
  - DAYS\_WORKED (number)
  - RATE\_CONV\_FORMULA (text)
  - HOURS\_WORKED (number)
  - IN\_AMOUNT (number)
  - UNIT\_TYPE (text)
  - PRORATION\_UNIT (text)
5. Add the formula outputs for the element input values.
6. Save, submit, and compile the formula.

Some countries or territories supply predefined proration formulas that you can use as the basis for your modified version.

### Related Topics

- [Set Up Element Proration](#)
- [How Prorated Earnings and Deductions are Calculated](#)
- [Periodicity Conversion](#)

## Configure Periodicity Conversion Rules

You can use the predefined periodicity conversion formulas when converting rates for hours multiplied by rate calculations of an element run result. You can also use these formulas for rates based on rate definitions and proration unless you specify another proration formula.

The conversion rule applies to the Flat Amount, Hours \* Rate, and Days \* Rate calculation rules.

Here's what you can do to use a different value instead of the predefined value for periodicity:

- Override the default periodicity for the element definition at the element entry level.
- Create your own periodicity conversion formula.

For example, you can create a formula to specify a different number of working hours when converting annual values into hourly rates.

**Note:** As a best practice, if you configure a periodicity rate formula, you should also create a related proration formula.

## Configure a Formula

Complete these steps to configure a rate conversion formula:

1. On the Home page, click the **Fast Formulas** task under the **My Client Groups** tab to search for formulas with the Rate Conversion formula type.
2. Search for and display the rate conversion formula you want to copy.
3. Create a new formula with the formula type Rate Conversion.
4. Copy the formula text into your new formula.
5. Edit the periodicity values.

For example, if you create a rate conversion formula that uses 7.5 hours instead of 8 hours for the number of work hours in a day, you would edit the periodicity as follows:

```
else if (out_periodicity = 'WORKHOUR') then
  l_amt = (l_year_amt / 260) / 7.5
```

6. Save, submit, and compile the formula.
7. Use the Elements task to search for the element and identify the formula used to calculate the element.
8. Use the Fast Formulas task to edit the formula.

Edit the rate converter call, replacing the rate conversion formula name with the name of your new formula.

```
CALL_FORMULA( 'RATE_CONVERTER',
  l_rate > 'SOURCE_AMOUNT',
  l_source_periodicity > 'SOURCE_PERIODICITY',
  l_target_periodicity > 'TARGET_PERIODICITY',
  'ANNUALIZED RATE CONVERSION' > 'method',
  /* replace with the name of new formula*/
  l_actual_amount < 'TARGET_AMOUNT' DEFAULT 0)
```

9. Save, submit, and recompile the formula.

### Related Topics

- [Overview of Using Formulas](#)
- [Periodicity Conversion](#)

## Element Input Validation Formula Type

You can use an element input validation formula to validate one or more element entry values. You can also use this formula type to provide a default value for an element entry value.

Also, you can calculate entry values based on the user's entries in other entry values.

You select the formula on the Element Summary page in these fields:

Page Section	Field	Purpose	When Does the Formula Run?
Element Details or Element Eligibility	Validation Formula	Validates one or more entry values for the element based on entries in other entry values.	When you save the element entry.
Element Details or Element Eligibility	Calculation Formula	Provides values for one or more entry values using a calculation formula that takes input from these entry values or other entry values.	When you save the element entry.
Element Details or Element Eligibility	Defaulting Formula	Provides default values for one or more entry values.	When you create the element entry.
Input Value	Validation Formula	Validates one entry value independent of others.	When you enter a value.

**Note:** A formula at the element eligibility level always overrides an equivalent formula at the element level.

Here are the contexts that are available to all formulas of this type:

- LEGISLATIVE\_DATA\_GROUP\_ID
- DATE\_EARNED
- EFFECTIVE\_DATE

These contexts are available to formulas only at element or element eligibility level; they aren't available to validation formulas at the input value level:

- PERSON\_ID
- PAYROLL\_RELATIONSHIP\_ID
- PAYROLL\_TERM\_ID
- PAYROLL\_ASSIGNMENT\_ID
- HR\_RELATIONSHIP\_ID
- HR\_TERM\_ID
- HR\_ASSIGNMENT\_ID

Here are the input variables that are available to formulas of this type.

Formula Usage	Input Variables	Comments
Validation formula at input value level	<code>entry_value</code>	Passes the value to be validated. You must declare the input variable as the appropriate type for the element input value.
Validation formula at element or element eligibility level	Any element input value name that corresponds to an entry value.	Replace spaces in the input value name with underscores in the input variable name. It doesn't matter whether you use uppercase or lowercase for the name.

Formula Usage	Input Variables	Comments
Defaulting formula	None	Use database items or other logic instead.
Calculation formula	Any element input value name of an entry value.	Replace spaces with underscores. You need not provide all the available entry values.

Here are the return values that are available to formulas of this type.

Formula Usage	Return Values	Comments
Validation formula at any level.	<b>formula_status</b>	Must be either 'S' (success) or 'E' (error). Required.
Validation formula at any level.	<b>formula_message</b>	The text of the message is passed to the user if the validation fails. Optional.
Defaulting formula	Any element input value name of an entry value.	A return value overrides any default value that's provided on the input value in the element or element eligibility record.
Calculation formula	Any element input value name of an entry value.	You don't need to return all the available entry values. You can return the entry values that were passed as input variables or other entry values.

Here are a few sample formula that you can use:

- Validation formula at input value level
- Validation formula at element or element eligibility level
- Calculation formula at element or element eligibility level
- Defaulting formula at element or element eligibility level

Validation formula at input value level:

```
inputs are entry_value(date)
if(entry_value = '01-APR-2008' (date)) then
(
formula_message = 'Valid date'
formula_status = 'S'
)
else(formula_message = 'Invalid date'
formula_status = 'E'
)
return formula_message, formula_status
```

Validation formula at element or element eligibility level:

```
inputs are hours_worked, rate, earning_date(date), comment(text)
if(hours_worked > 80) then
```

```

(
formula_message = 'You are within the working limit.
'formula_status = 'S'
)
else
(
formula_message = 'You have worked too many hours.
'formula_status = 'E'
)
return formula_message, formula_status

```

Calculation formula at element or element eligibility level:

```

inputs are hours_worked, rate, comment(text)
if(hours_worked > 80) then
(
rate = rate * 1.2
comment = 'Your rate has been increased'
)
return rate, comment

```

Defaulting formula at element or element eligibility level:

```

if(CATEGORY = 'S') then
(
rate = 20
)
else
(
rate = 30
)
rate_code = 'B'
return rate, rate_code

```

## User Table Validation Formula Type

The User Table Validation formula type validates entries in user-defined tables. When you create or edit user-defined tables, select the formula in the Formula field for the user-defined columns.

You can use this formula type to ensure that entries are:

- Between a specified range
- Don't contain a negative value

You can use the EFFECTIVE\_DATE (text) context for formulas of this type.

You must use one input variable called ENTRY\_VALUE of data type `text`.

These return values are available to formulas of this type:

Return Value	Data Type	Required	Description
FORMULA_MESSAGE	Text	N	Returns a text message for either or both statuses. The message is displayed on the Create User-Defined Table: User-Defined Table Values page.

Return Value	Data Type	Required	Description
FORMULA_STATUS	Text	Y	Returns the value S (success) or E (error).

This formula checks whether the deduction value entered in the Union A column of the Union Dues table is between 10.00 and 20.00:

```
/* Formula Name: Union A Dues Validation */
/* Formula Type: User Table Validation */
INPUTS ARE entry_value (text)
IF TO_NUMBER(entry_value) < 10.00 OR
TO_NUMBER(entry_value) > 20.00
THEN
(
formula_status = 'e'
formula_message = 'Error: Union A dues must be between $10.00 and $20.00.'
)
ELSE
(
formula_status = 's'
formula_message = ' '
)
RETURN formula_status, formula_message
```

## Payroll User Interface Configuration Formula Type

You can use the Payroll User Interface Configuration formula type to control the configurable preferences for payment methods. On the Fast Formulas page, you can create formulas to override the default values by creating new values for these preferences.

After you create your formulas, you can attach them to the appropriate rows in the PAYROLL\_USER\_INTERFACE\_CONFIGURATION user-defined table.

This table lists the configurable preferences, their predefined row name in the user-defined table, purpose, and default values.

User-Defined Row Name	Purpose	Default Value
Default Organization Payment Method	Sets the default organization payment method in the simplified UI for each legislative data group.	(Not applicable)
Execute Personal Payment Method Validation	Validates the customer personal payment method criteria. For example, you can create only one payment method with a specific account type.	No
Maximum Number of Personal Payment Methods	Sets the maximum allowed number of personal payment methods.	No limit

User-Defined Row Name	Purpose	Default Value
Payment Types Available to Workers	Limits the creation of personal payment methods to a specific payment type.	All available organization payment methods
Prevent Edit Personal Payment Method	Prevents employees from modifying any personal payment method details that meet the criteria set in the formula.	No
Show Percentage or Amount	Displays only Amount or Percentage amount types on the Manage Personal Payment Methods page.	Both

**Note:** Each preference that you configure must have its own formula. You can't combine different preferences into a single formula. For example, you can't create a formula that sets the default organization payment method and also sets the maximum number of allowed personal payment methods.

The Payroll User Interface Configuration formula type doesn't support database items. As with other formula types, to retrieve information when a database item isn't available, use the GET\_VALUE\_SET function.

**Note:** When using the GET\_VALUE\_SET function, ensure that the **Value Attributes Table Alias** field for the value set has no value. The function fails if you provide an alias.

These input values are available to all formulas of the Payroll User Interface Configuration formula type:

Input Variable	Type	Description
EFFECTIVE_DATE	Text	The effective date on which the formula validation will be applied.
LEGISLATIVE_DATA_GROUP_ID	Text	The number that identifies the legislative data group for the variable.
PAYROLL_RELATIONSHIP_ID	Text	The number that identifies the payroll relationship for the variable.

These input values are available to the two personal payment method validation formulas:

Input Variable	Type	Description
AMOUNT	Text	Total amount to be paid to the personal payment method.
BANK_ACCOUNT_NUMBER	Text	Bank account number for the external bank account.

Input Variable	Type	Description
BANK_ACCOUNT_TYPE	Text	Type of the external bank account for the personal payment method.
BANK_NAME	Text	Name of the bank associated with the personal payment method.
BRANCH_NAME	Text	Name of the bank branch associated with the personal payment method.
CURRENCY_CODE	Text	The currency code of the personal payment method.
ORGANIZATION_PAYMENT_METHOD_NAME	Text	Name of the organization payment method.
PAYMENT_AMOUNT_TYPE	Text	Percentage or amount attributed to the personal payment method.
PAYMENT_TYPE_NAME	Text	Payment type, such as check or direct deposit for a particular organization payment method.
PERCENTAGE	Text	The percentage amount attributed to the personal payment method.
PERSONAL_PAYMENT_METHOD_NAME	Text	The name of the personal payment method.
PRIORITY	Text	The priority order of the personal payment method.

This context is available only for formulas mapped to the Default Organization Payment Method row in the user-defined table.

- PAYMENT\_TYPE\_NAME (text)

Text representing the payment type in the expected format, for example, EFT or Check.

Unlike other formulas, the return values for the Payroll User Interface Configuration formula type are variables that you declare in your formulas. Refer to the sample formulas for examples.

This sample formula sets default organization payment methods in LDG\_A (ID 300100001) based on the payment type. To use this rule, attach your formula to the **Default Organization Payment Method** row in the PAYROLL\_USER\_INTERFACE\_CONFIGURATION user-defined table. In this example, the valid return values for **DEFAULT\_OPM** are the exact names of organization payment methods.

```
*****
FORMULA NAME: Default OPM Formula
FORMULA TYPE: Payroll User Interface Configuration/
*****
/* Configuration */
IF (LEGISLATIVE_DATA_GROUP_ID = '300100001'
AND PAYMENT_TYPE_NAME = 'EFT')
```

```

THEN DEFAULT_OPM = 'NACHA_OPM_A'
' ELSE IF (LEGISLATIVE_DATA_GROUP_ID = '300100002
'AND PAYMENT_TYPE_NAME = 'Check')
THEN DEFAULT_OPM = 'CHECK_OPM_A'
ELSE DEFAULT_OPM = 'NODATA'
RETURN DEFAULT_OPM
/* End Formula Text */

```

This sample formula limits personal payment methods to be based only on organization payment methods of EFT (electronic funds transfer) or check payment types. To use this rule, attach your formula to the **Payment Types Available to Workers** row in the PAYROLL\_USER\_INTERFACE\_CONFIGURATION user-defined table. In this example, the valid return values are the base payment type names defined in the table PAY\_PAYMENT\_TYPES\_VL.

```

*****
FORMULA NAME: Worker Payment Types Formula
FORMULA TYPE: Payroll User Interface Configuration/
*****
/* Configuration */
PAYMENT_TYPE[1] = 'EFT'
PAYMENT_TYPE[2] = 'Check'
RETURN PAYMENT_TYPE
/* End Formula Text */

```

This sample formula limits the number of personal payment methods for employees in LDG\_A (ID 300100001) to **3**, and employees in LDG\_B (ID 300100002) to **1**. To use this rule, attach your formula to the **Maximum Number of Personal Payment Methods** row in the PAYROLL\_USER\_INTERFACE\_CONFIGURATION user-defined table. In this example, the valid return values for **MAX\_NUM\_PPMS** are integers.

```

*****
FORMULA NAME: Maximum PPM Formula
FORMULA TYPE: Payroll User Interface Configuration
*****
/* Configuration */
IF LEGISLATIVE_DATA_GROUP_ID = '300100001'
THEN MAX_NUM_PPMS = '3'
IF LEGISLATIVE_DATA_GROUP_ID = '300100002'
THEN MAX_NUM_PPMS = '1'
ELSE
MAX_NUM_PPMS = 'NO DATA'
RETURN MAX_NUM_PPMS
/* End Formula Text */

```

This sample formula sets a restriction to display only the Percentage amount type and field on the Manage Personal Payment Methods page. To use this rule, attach your formula to the **Show Percentage or Amount** row in the PAYROLL\_USER\_INTERFACE\_CONFIGURATION user-defined table. In this example, the valid return values for **PAYMENT\_AMOUNT\_TYPE** are **AMOUNT OR PERCENTAGE**.

```

*****
FORMULA NAME: Show Percentage Formula
FORMULA TYPE: Payroll User Interface Configuration
*****
/* Configuration */
PAYMENT_AMOUNT_TYPE = 'PERCENTAGE'
RETURN PAYMENT_AMOUNT_TYPE
/* End Formula Text */

```

This sample formula validates that the first personal payment method is Pay Card and the payment type is either EFT or IAT. And the account type is Pay Card. This formula ensures that the first personal payment method meets these criteria and that an employee has only one personal payment method of this type.

```

*****
FORMULA NAME: Execute Personal Payment Method Validation
FORMULA TYPE: Payroll User Interface Configuration

```

```
*****
IF (PAGE_NAME = 'DETAILS') {
//PPM Validation
//check if there is PPM for PAYROLL_RELATIONSHIP_ID
COUNT = NUMBER_OF_PPMS

//This returns the number of PPMs for a payroll
relationship ID using a value set
//If this is first PPM
IF(COUNT==0) {
IF((PAYMENT_TYPE_NAME==EFT OR PAYMENT_TYPE_NAME==IAT)
AND BANK_ACCOUNT_TYPE==PAYCARD) {
//If first PPM and is of type PAYCARD, PPM can be created
RETURN_VALUE[1]= 'N'
}
ELSE{
//If first PPM is not PAYCARD, throw error message
RETURN_VALUE[1]='Y'
RETURN_VALUE[2]='ANY_VALID_ERROR_MESSAGE'
}
}
//This is not the first PPM.
ELSE{
IF((PAYMENT_TYPE_NAME==EFT OR PAYMENT_TYPE_NAME==IAT)

AND BANK_ACCOUNT_TYPE==PAYCARD) {
//If multiple PAYCARD PPMs, throw error message
RETURN_VALUE[1]='Y'
RETURN_VALUE[2]='ANY_VALID_ERROR_MESSAGE'
}
ELSE{
RETURN_VALUE[1]='N'
}
}
}
//When changing the priority of a PPM
ELSE IF(PAGE_NAME == 'SUMMARY') {
{
//Get the highest priority for PAYROLL_RELATIONSHIP_ID
IF (PRIORITY == HIGHEST_PRIORITY) {
//Gets the highest priority among the PPMs for a particular payroll relationship ID
//by executing a value set. Change in priority throws error message.
RETURN_VALUE[1]='Y'
RETURN_VALUE[2]='ANY_VALID_ERROR_MESSAGE'
}
ELSE
RETURN_VALUE[1]='N'
}
}
}

*****
```

This sample formula prevents a self-service user from editing personal payment methods that are associated with an organization payment method of payment type Check.

```
*****
FORMULA NAME: Prevent Edit Personal Payment Method
FORMULA TYPE: Payroll User Interface Configuration
Expected Behavior : If value returned is N, no change in functionality
If value returned is Y, Save/Submit buttons will be disabled in edit flow to prevent the edit of personal
payment method.
The Delete buttons will also be disabled for the personal payment methods

Sample Input File Format:
User Interface|Legislative Data Group|Effective Date|Payroll Relationship|Payment Type
*****
/* inputs */
```

```
INPUTS ARE PAGE_NAME (text), LEGISLATIVE_DATA_GROUP_ID (text), PAYROLL_RELATIONSHIP_ID (text),  
EFFECTIVE_DATE(text), PERSONAL_PAYMENT_METHOD_NAME (text), ORGANIZATION_PAYMENT_METHOD_NAME (text),  
PAYMENT_TYPE_NAME (text), CURRENCY_CODE (text), PAYMENT_AMOUNT_TYPE (text), PERCENTAGE (text), AMOUNT  
(text), PRIORITY (text), BANK_NAME (text), BRANCH_NAME (text), BANK_ACCOUNT_NUMBER (text), BANK_ACCOUNT_TYPE  
(text)  
  
/* Configuration */  
  
IF ORGANIZATION_PAYMENT_METHOD_NAME='Check' THEN  
(  
OUTPUT_VALUE = 'Y'  
)  
ELSE  
(  
OUTPUT_VALUE = 'N'  
)  
  
RETURN OUTPUT_VALUE  
  
/* End Formula Text */
```

*Related Topics*

- [Configure Payment Method Preferences](#)

## Payroll Access to HR Formula Type

Use the Payroll Access to HR formula type to call a payroll formula when your payroll formula requires an HR database item.

For example, you can use the payroll formula to return the salary amount, and then use the Payroll Access to HR formula to return a full-time equivalent (FTE) database item for a specific group of employees.

These contexts are available to all formulas of this type:

- HR\_RELATIONSHIP\_ID
- HR\_TERM\_ID
- GRADE\_RATE\_ID
- LEGAL\_EMPLOYER\_ID
- AREA1
- LOCATION\_ID
- HR\_ASSIGNMENT\_ID
- PERSON\_ID
- JOB\_ID
- GRADE\_ID
- ADDRESS\_ID
- ADDRESS\_TYPE
- ACCRUAL\_PLAN\_ID
- EFFECTIVE\_DATE

- LEGISLATIVE\_DATA\_GROUP\_ID
- ORGANIZATION\_ID

These input variables are available to formulas of this type.

Enter Data	Data Type
ACTUAL_END_DATE	Date
ACTUAL_START_DATE	Date
EFF_DATE	Date
END_DATE	Date
ENTRY_LEVEL	Text
HR_ASSG_ID	Number
HR_ASSIGN_ID	Number
HR_EFFECTIVE_DATE	Date
HR_ENTRY_LEVEL	Text
HR_ID	Number
HR_ID2	Number
HR_TRM_ID	Number
MODE	Text
START_DATE	Date
UNIT_TYPE	Text

These return values are available to formulas of this type:

Return Value	Data Type	Description
L_SALARY	Number	Salary ID of employee
L_SALARY_BASIS_CODE	Text	Salary basis of employee

Return Value	Data Type	Description
L_UNITS	Number	Number of units worked
X_OUTPUT	Date	Output date
X_OUTPUT	Text	Output text
X_OUTPUT_NUMBER	Number	Output number
X_OUTPUT_TEXT	Text	Output text

This sample Oracle Payroll formula returns the salary amount for employees based on their full-time equivalent (FTE), which is an HR database item. Because HR database items aren't accessible through Oracle Payroll formulas, you must create a Payroll Access to HR formula to retrieve the database item. And then call the formula that retrieves the database item from the formula that returns the salary amount.

```
*****
FORMULA TYPE: Payroll Access to HR
DESCRIPTION: Retrieve FTE DBI
*****
/* Default Statement Section */
Default for PER_ASG_FTE_VALUE is 1/
* Calculation Section */
HR_FTE = PER_ASG_FTE_VALUE
/* Return Statement Section */
Return HR_FTE
*****
FORMULA NAME: GB Salary by FTE
FORMULA TYPE: Oracle Payroll
DESCRIPTION: Calculate the salary amount for OK employees based on their FTE.
*****
/* Alias Statement Section */
ALIAS ANNUAL_SALARY_UK_AMOUNT_ASG_ENTRY_VALUE AS UK_ASG_SAL
/* Default Statement Section */
Default for ASG_HR_ASG_ID is 1
Default for UK_ASG_SAL is 0
Default for Salary_UK is 0
/* Calculation Section */
Salary_UK = UK_ASG_SAL
SET_INPUT('HR_ASSIGNMENT_ID', ASG_HR_ASG_ID)
EXECUTE('Call HR FTE DBI')
FTE = GET_OUTPUT('HR_FTE',1)
l_amount = round((FTE * Salary_UK),2)
Message = 'Salary Value is'||to_Char(l_Amount)
/* Return Statement Section */
RETURN l_Amount,Message
```

# Payroll Relationship Group Formula Type

The Payroll Relationship Group formula type returns Yes or No to indicate whether a person is part of a payroll relationship group. The formula contains the criteria that define the group.

You can use these groups to define a set of people for payroll processing, data entry, or reporting. On the Object Groups page, select the formula when creating a payroll relationship group.

For example, you can use the Payroll Relationship Group formula type to restrict the payroll run process to a specific set of employees based on assignment and person-level attributes.

By default, you create formulas of this type using the Expression editor on the Create Fast Formula page. However, you can use the text editor to create more complex formulas, if required.

These contexts are available to all formulas of this type:

- PAYROLL\_RELATIONSHIP\_ID
- PAYROLL\_TERM\_ID
- DATE\_EARNED
- OBJECT\_GROUP\_ID
- EFFECTIVE\_DATE
- PAYROLL\_ASSIGNMENT\_ID
- PAYROLL\_ID
- LEGISLATIVE\_DATA\_GROUP\_ID

You must have at least one input value called LEVEL\_NAME. The data type is text and these are the valid values:

- PAY\_REL
- PAY\_TERM
- PAY\_ASG

This return value is available to the INCLUDE\_FLAG formula type. The data type is text.

This formula calls the summary formula for each of the levels in the person group until the membership is established.

```
/*
DEFAULT_DATA_VALUE FOR PERSON_GROUP_LEVEL_SUMMARY_FORMULA_NAME is 'null'
NEED_CONTEXT(PAYROLL_RELATIONSHIP_ID) NEED_CONTEXT(PAYROLL_TERM_ID)
NEED_CONTEXT(PAYROLL_ASSIGNMENT_ID)
NEED_CONTEXT(LEGISLATIVE_DATA_GROUP_ID)
NEED_CONTEXT(PAYROLL_ID) NEED_CONTEXT(DATE_EARNED)
NEED_CONTEXT(EFFECTIVE_DATE)

in_group = 'N'

SET_INPUT('level_name', 'PAY_REL')
EXECUTE (PERSON_GROUP_LEVEL_SUMMARY_FORMULA_NAME[1])
in_group = GET_OUTPUT('INCLUDE_FLAG', 'N')

IF (in_group = 'N') THEN
(
SET_INPUT('level_name', 'PAY_TERM')
EXECUTE (PERSON_GROUP_LEVEL_SUMMARY_FORMULA_NAME[1])
in_group = GET_OUTPUT('INCLUDE_FLAG', 'N')
```

```
)  
IF (in_group = 'N') THEN (SET_INPUT('level_name', 'PAY_ASG')  
EXECUTE (PERSON_GROUP_LEVEL_SUMMARY_FORMULA_NAME[1])  
in_group = GET_OUTPUT('INCLUDE_FLAG', 'N')  
)IF in_group = 'Y' THEN  
INCLUDE_FLAG = 'Y'  
  
RETURN INCLUDE_FLAG
```

## Flow Schedule Formula Type

Use Flow Schedule formulas to control when to submit the current flow and how often to submit future instances of the flow. You create scheduling formulas on the Manage Fast Formulas page when the predefined formulas don't meet your requirements.

Let's say you create a formula that loads time card batches daily, and increases to four times a day at the end of a payroll period when workers typically submit their time cards. You can create a formula that schedules the frequency with which an extract process checks for new starter details.

Here are a few points to consider when creating or updating a scheduling formula:

- Specify a meaningful name to assist the person selecting the formula.
- Review the formula to ensure it doesn't contain negative numbers that might produce an error condition, such as running a process continually.
- After updating the formula, cancel any scheduled flows that use the formula. Resubmit the flow to apply the updated definition.

The SCHEDULED\_DATE (scheduled date) context is available to formula of this type.

Here's the list of database items that are available to Flow Schedule formulas.

Database Item	Data Type	Description
FF_ADD_DAYS	Date	Function to add days to a date.
FF_ADD_MONTHS	Date	Function to add months to a date.
NEXT_SCHEDULED_DATE	Date	Calculated value for the date to schedule the next flow.
SCHEDULED_DATE	Date	Date used to schedule the flow.

These are the input variables available to Flow Schedule formulas.

Input Variables	Data Type	Required	Description
SCHEDULED_DATE(DATE)	Date	Y	Date on which to schedule the flow. The date is passed to the formula

Input Variables	Data Type	Required	Description
			when it calculates the next date to schedule the flow.

Use predefined names for return variables. These are the return values available to Flow Schedule formulas.

Return Values	Data Type	Required	Description
NEXT_SCHEDULED_DATE	Date	Y	The date calculated by the formula to schedule the next flow.

This predefined formula schedules a flow so that it's submitted weekly from the date the flow owner initially submitted it.

```
*****
FORMULA NAME: Weekly
FORMULA TYPE: Flow Schedule
DESCRIPTION: Formula to return a date time.
Returns NEXT_SCHEDULED_DATE;
Formula Results :
NEXT_SCHEDULED_DATE This is a date time value with yyyy-MM-dd HH:mm:ss format.
*****
/* Inputs */
INPUTS ARE SUBMISSION_DATE(DATE), SCHEDULED_DATE(DATE)
/* Calculations */
NEXT_SCHEDULED_DATE = ADD_DAYS(SCHEDULED_DATE,7)
/* Returns */
RETURN NEXT_SCHEDULED_DATE
/* End Formula Text */
```

You can calculate units smaller than a day by replacing the calculation portion of the formula text using a decimal or a fraction. Let's look at some examples of submitting a flow several times a day.

Flow Submission	Formula Text for Calculation
Twice a day	NEXT_SCHEDULED_DATE =ADD_DAYS(SCHEDULED_DATE,0.5)
	<p><b>Note:</b> For accuracy, enter a value with at least ten decimal places. The formula text supports a maximum of 14 decimal places.</p>
Hourly	NEXT_SCHEDULED_DATE =ADD_DAYS(SCHEDULED_DATE,1/24)

## Balance Exception Formula Type

Use the Balance Exception formula type to create formulas to return reference values for comparison in Balance Exception report.

On the Create Balance Exception page, enter **Formula** in the **Variance Type** field, and then enter the name of the formula that you create using this formula type in the **Formula Name** field.

These contexts are available to all formulas of this type:

- EFFECTIVE\_DATE
- PERSON\_ID
- HR\_ASSIGNMENT\_ID
- TAX\_UNIT\_ID
- PAYROLL\_RELATIONSHIP\_ID
- PAYROLL\_REL\_ACTION\_ID
- PAYROLL\_TERM\_ID
- PAYROLL\_ASSIGNMENT\_ID
- LEGISLATIVE\_DATA\_GROUP\_ID
- PAYROLL\_ID
- CALC\_BREAKDOWN\_ID
- AREA\_1
- AREA\_2
- AREA\_3
- AREA\_4
- AREA\_5
- AREA\_6
- PAYROLL\_STAT\_UNIT\_ID
- INSURANCE\_TYPE
- PENSION\_TYPE
- DEDUCTION\_CARD\_ID

You can reference any database item in the fast formula that uses any of the listed contexts.

You can't use input variables for this formula type. This formula is run by the balance exception report for each person record. It returns the reference value for that employee for comparison with the balance value that's configured in the respective balance exception.

These return values are available to formulas of this type:

Return Value	Data Type	Required	Description
REFERENCE_VALUE	Text	Y	Returns a reference value for comparing in the balance exception report.
REFERENCE_INFO	Text	Y	Returns the text that's displayed in the <b>Reference</b> field in the balance exception report.

This formula returns 100 as a static value when the balance exception report it's associated with is run.

```
*****
FORMULA NAME: SAMPLE_BEX_FORMULA_1
FORMULA TYPE: Balance Exception
DESCRIPTION: This is a sample formula that returns a static value.
Formula Results:
reference_value - Reference Value for comparison in the Balance Exception Report.
reference_info - Reference Value for reporting in the Balance Exception Report.
*****
reference_value = 100
reference_info = 'Ref Info'
RETURN REFERENCE_VALUE,REFERENCE_INFO
/* End Formula Text */
```

This formula multiplies the year-to-date gross earnings by 75 percent and returns the value when the balance exception report it's associated with is run.

```
*****
FORMULA NAME: SAMPLE_BEX_FORMULA_3
FORMULA TYPE: Balance Exception
DESCRIPTION: This sample formula returns a Balance Value Formula
Results:
reference_value - Reference Value for comparison in the Balance Exception Report.
reference_info - Reference Value for reporting in the Balance Exception Report.
*****
ytd_value = GROSS_EARNINGS_ASG_YTD
reference_value = 0.75* ytd_value
reference_info = 'Ref Info'
RETURN REFERENCE_VALUE,REFERENCE_INFO
/* End Formula Text */
```

#### Related Topics

- [Balance Exceptions for the US](#)

## Element Skip Formula Type

If your payroll policies require conditional processing of an element, you can write a formula to define when the run should process the element, and when it should skip it.

For example, your formula can specify to process the Union Fees element in every run, unless the Union\_Fees\_Paid balance is greater than 10,000.

Your skip rule formula must be consistent with other processing rules defined for the element, such as frequency rules, which determine in which period the element is normally processed. A skip rule cannot contravene any other processing rules in place for the element. You can associate only one element skip rule formula with each element. You must write and validate the formula before you define the element, so that you can select the formula from a list on the **Element** window.

Here's how you can write a formula defining a skip rule:

1. Select the **Element Skip** formula type in the **Formulas** window.

The formula must set and return a local variable of type text, and named as skip\_flag.

The following example of a skip rule formula defines that the Union Fees element isn't processed if the Union\_Fees balance is greater than or equal to 10,000:

```
DEFAULT FOR UNION_FEES_REL_NOCB_YTD IS 0

l_amount = 0
l_amount = UNION_FEES_REL_NOCB_YTD

SKIP_FLAG = 'N'
IF l_amount >= 10000 THEN
(
SKIP_FLAG = 'Y'
)
RETURN SKIP_FLAG
```

2. Associate this formula to the element in the **Skip Formula** field.

#### Related Topics

- [What's an element's skip rule?](#)

## Get the Rate Amount for an Assignment Base Rate

To get the rate amount for a specific rate of the salary, use the GET\_SALARY\_RATE\_DATA function. You can use this function in payroll processing and to interface with other internal and external apps.

For payroll processing, you can use the rates-based ORA\_CMP\_ASSIGNMENT\_SALARY\_RATES\_UE and ORA\_CMP\_ASSIGNMENT\_SALARY\_RATES\_RGE\_UE user entities to extract rate-based salary components for a particular date or date range.

## Parameters

Parameter	Data Type	Comments
P_BASE_RATE_NAME	TEXT	Name of the base rate given in the rate definition, to get the rate amount
P_AMOUNT_TYPE	TEXT	If the input value is null or RATE_AMOUNT, returns the rate amount If the input value is ANNUAL_AMOUNT, returns the annual amount for the rate

Parameter	Data Type	Comments
		If the input value is ANNUAL_FT_AMOUNT, returns the annual full-time amount for the rate If the input value is RATE_FACTOR, returns the factor for the rate

*Related Topics*

- [User Entities for Extracting Rate-Based Salary Components](#)

# 6 Formulas for Absence Management

## Options to Define Qualification Plans

Configure the following rules when you create an absence qualification plan in accordance with the leave policy of your enterprise:

- Plan term
- Plan eligibility
- Enrollment and termination
- Payments

### Plan Term

A qualification plan term is an assessment period for which the Evaluate Absence process calculates entitlements for the total absent time recorded in that period. When you create an absence qualification plan, you must select the type of plan term. For example, you can limit the duration of the plan term to the duration of the absence.

### Plan Eligibility

Associate an eligibility profile with the qualification plan to determine the set of workers who are eligible to record an absence that belongs to that plan.

### Enrollment and Termination

Decide when to enroll workers in the qualification plan. Also, decide whether ongoing payments under this plan must continue if a worker is terminated or loses eligibility for the plan.

### Payments

Use an entitlement band matrix to determine the payment percentages that apply for specific time periods during an absence. Decide how you want to calculate the payment rate of a single unit of absence. You can use a rate definition to include the calculation rules, or use a formula. For example, you want workers who have completed a particular tenure to receive specific percentage of pay for a specific absence period.

The following table shows a sample scenario:

Length of Service	Payment Rule
5 to 10 years	75 percent up to 10 absent days.
10 to 20 years	75 percent up to 20 absent days.

Decide how you want to calculate the payment rate of a single unit of absence. You can use a rate definition to include the calculation rules, or use a formula.

*Related Topics*

- [Create a Maternity Plan](#)

## Options to Define Accrual Plans

Configure the following rules when you create an absence accrual plan in accordance with the leave policy of your enterprise:

- Plan term
- Plan eligibility
- Enrollment and termination
- Transfer and rollover
- Prior balance reinstatement
- Vesting period
- Plan limits
- Balance Updates
- Payments
- Disbursement
- Donation

### Plan Term

An accrual term is a period of time during which workers accrue time. You must specify the type of accrual term to use for the plan. For example, you can define one of these term types:

- An accrual term of one calendar year that restarts on January 1
- An accrual term that starts on the worker's annual hire date and restarts on every anniversary

### Plan Eligibility

Associate an eligibility profile with the accrual plan to determine the set of workers who can enroll in that plan.

### Enrollment and Termination

Decide when to enroll workers in the accrual plan. Indicate how to handle negative and positive balances where only plan enrollment ends, or both plan enrollment and employment ends.

### Transfer, Rollover, and Carryover

Define the limits for transfer of some or all of the accrual balance to a new plan when the existing plan is terminated due to loss of eligibility. The Transfer Rules section is enabled only when you select **Transfer positive balance** in the Balance Disposition section.

Additionally, for year end processing, you can define:

- Rollover limits so that employees can transfer remaining balance to a new plan. Select **Unlimited** if you don't want to define a limit for rollover.
- Carryover limits to determine the maximum unused leave time that employees can take over to the next term for the same plan.

If the unused absence balance at the end of the term is above the defined rollover and carryover limit, you can choose to pay out the remaining balance. If you don't select the **Disburse remaining balance** check box, the unused accrual balance expires.

**Note:** During the year end processing, the application first processes the rollover, and only then processes the carryover with the leftover balance.

## Prior Balance Reinstatement

Configure accrual plans so that when a worker is terminated or loses plan eligibility, the remaining balance can be optionally held for reinstatement upon rehire or return to plan eligibility.

Absence plan setup options allow you to define the balance amount eligible for reinstatement along with the time frame for which the withheld balance can be reinstated. You can configure this option in the Re-enrollment Rules section. You can find this section in the Plan Participation tab when you create or edit an absence plan.

If you select **Allow Prior Balance Reinstatement**, a new row in accruals called "Plan Balance close transaction" is created that deducts the remaining balance on the last day of the employee. This is the transaction that will act as a reference to calculate the reinstated balance if the employee is re-hired within the time frame limit configured in the plan setup.

However, if you select **Allow Prior Balance Reinstatement**, and leave the **Balance Reinstatement Limit** and **Time Frame Limit** fields blank, the entire closing balance can be reinstated any time.

## Vesting Period

Define if you want newly enrolled workers to accrue time, but not use it until after a specific amount of time.

## Plan Limits

Define rules for the maximum leave time that workers can accrue. For incremental plans, Absence Management applies the limit to each accrual period. When there are accrual transactions such as adjustments, balance transfers, and disbursements, which impact the ceiling or carryover value, then the periodic accrual balance calculation takes precedence and readjusts the balance accordingly.

## Balance Updates

You can enable the following types of adjustments that HR specialists can make during maintenance of absence records and entitlements:

- Balance transfer across plans
- Other adjustments
- Elective disbursements

## Payments

Decide how you want to calculate payment of accrual balances for the following scenarios:

- When workers must be paid a different rate during the absence period
- When a part of the accrual balance must be disbursed to workers as cash
- When the cost of accrual balance must be calculated to determine employer liability
- When the accrual balance must be paid to workers when their plan participation ends

## Disbursement

Determine whether workers are eligible for cash disbursement requests. Decide who can initiate the disbursement. Additionally, define disbursement rules and the number of hours that can be disbursed.

## Donation

Define whether workers are eligible to donate accrual balances to a coworker. Decide who can initiate the donation. Additionally, define donation rules and the number of hours that can be donated.

### Related Topics

- [Create a Vacation Accrual Plan](#)

## Formulas for Accrual Plans

Use the Absence Plan page to apply delivered accrual plan rules in the plan. However, if you want to define other special rules to suit your requirement, you can write your own formulas.

## Formulas for Accrual Plans

The following table lists the aspects of an accrual plan for which you can write a formula and identifies the formula type for each.

Rule	Description	Formula Type to Use
Enrollment Start	Date when eligible workers are enrolled in the plan. If a worker is already enrolled in an existing accrual plan, you can't use this formula to change the enrollment start date.	Global Absence Plan Enrollment Start
Enrollment End	Date when workers are disenrolled from the plan. This formula works only if there is an eligibility profile associated with the plan.	Global Absence Plan Enrollment End
Conversion Formula	Method to override the default absence plan entry duration.	Global Absence Plan Duration

Rule	Description	Formula Type to Use
	Example: You might have a requirement to consider only whole working days in a vacation absence to update the accrual balance. In such cases, you define logic in a formula to convert the absence duration to a value that excludes partial days.	
Anniversary Event	Method to determine the employment anniversary date on which you want the accrual plan to restart.	Global Absence Plan Period Anniversary Event Date
Accrual Event	<p>Method to capture information about events during a calendar year that affect the accrual band of a worker.</p> <p>Example: An organization has a vacation plan in which enrolled workers accrue a certain number of days every year based on their grade. When the grade of a worker changes in the middle of the calendar year, the organization might want to prorate their total accrual balance. You can configure this proration rule using the global absence accrual event formula to capture the dates when such changes occur.</p> <p>This formula can't be attached to the plan definition at any point. When you create this formula, the formula name needs to be the same as the absence plan name. This automatically links the formula to the plan.</p>	Global Absence Accrual Event
Accrual Vesting	A period during which workers accrue time, but can't use it.	Global Absence Vesting
Accrual Proration	Method to calculate the time workers accrue if they enroll in the middle of an accrual period.	Global Absence Proration
Ceiling	The maximum time that a worker can accrue.	Global Absence Ceiling
Ceiling Proration	Method to return a multiplying factor to prorate the defined ceiling limit.	Global Absence Ceiling Proration
Carryover	The maximum unused time that a worker can transfer to the next accrual term.	Global Absence Carryover
Carryover Proration	Method to return a multiplying factor to prorate the defined carryover amount.	Global Absence Carryover Proration
Accrual Definition	Method to determine the paid time, eligible workers accrue over the course of an accrual term.	Global Absence Accrual

Rule	Description	Formula Type to Use
Accrual Formula	Range of eligibility criteria that identify how much paid time eligible workers accrue over the course of an accrual term. The criteria may be years of service, grades, hours worked, or some other factor that you can define.	Global Absence Accrual Matrix
Partial Accrual Period	Method to determine the prorated accrual amount for workers who enroll or disenroll from a plan during the plan period.	Global Absence Partial Accrual Period Rate
Absence Payment Rate	Method to calculate payment during absence period.	Global Absence Plan Use Rate
Discretionary Disbursement Rate	Method to calculate payment when paying out part of the accrual balance.	Global Absence Plan Use Rate
Disbursement Rule	Method to determine the minimum, maximum, and increment amounts that the worker is eligible for to request a cash disbursement.	Global Absence Discretionary Disbursement Rule
Final Disbursement Rate	Method to calculate payment of accruals when plan participation ends.	Global Absence Plan Use Rate
Liability Rate	Method to calculate cost of accrual balance to determine employer liability.	Global Absence Plan Use Rate

When you schedule an absence for a worker using an accrual plan with formulas defined, the absence doesn't appear in the Absence Records section of the Manage Absences and Entitlements page. Instead, you can view the absence in the Plan Participation section. To view the absence:

1. In the Accrual Plans section, click the accrual plan to open the Accrual Plan Balance dialog box.
2. Enter the **Balance Calculation Date** to view the details of the absence. The details are displayed in the Summary and Details tabs.

## Global Absence Accrual Matrix Formula

The Global Absence Accrual Matrix fast formula can be used in conjunction with the accrual matrix to implement requirements such as band change proration, FTE proration<sup>1</sup> and so on.

An organization might have a vacation plan in which workers enrolled into the plan can accrue days every year based on their grade. If the grade changes mid-period, then the total accrual needs to be prorated based on the amount of time that the worker spends in each band. This can be achieved by defining an accrual matrix that is based on grades and using a combination of accrual event formula and accrual matrix formula.

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

## Navigation

1. In the **Absence Administration** work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Accrual**.
4. Click **Continue**.
5. On the Edit Absence Plan page Accruals tab there is a table in the Accrual Matrix section.
6. Select the defined formula in the Accrual Formula column in the table.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number

Context Name	Data Type
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_ACCRUAL	Number	Accrual value received from the accrual matrix.
IV_CARRYOVER	Number	Carryover limit received from accrual matrix.
IV_CEILING	Number	Ceiling limit received from the accrual matrix.
IV_ACCRUALPERIODSTARTDATE	Date	Start date of accrual period.
IV_ACCRUALPERIODENDDATE	Date	End date of accrual period.
IV_CALENDARSTARTDATE	Date	Start date of accrual calendar year.
IV_CALEDARENDDATE	Date	End date of accrual calendar year.
IV_PLANENROLLMENTSTARTDATE	Date	Start date of enrollment.
IV_PLANENROLLMENTENDDATE	Date	End date of enrollment.
IV_BAND_CHG_DT1*	Date	If the accrual matrix band definition consists of grades or derived factors length of service or age, then this input value fetches the date on which worker has had a change in any of these factors within the period.
IV_BAND_CHG_BEFVAL1*	Number	Accrual value as per the matrix before IV_BAND_CHG_DT1.
IV_BAND_CHG_AFTVAL1*	Number	Accrual value as per the matrix after IV_BAND_CHG_DT1.
IV_EVENT_DATES	Date_Number	Array of dates returned by the Accrual Event formula.
IV_ACCRUAL_VALUES	Number_Number	Array of accrual values fetched from the accrual matrix as of the dates contained in the accrual event dates array.
IV_ACCRUAL_CEILING	Number	Annual accrual limit from the accrual matrix.

## Return Variables

Variable Name	Data Type	Description
accrual	Number	Amount of accrual that the worker accrues in the specific period.
ceiling	Number	Maximum time that a worker can accrue.
carryover	Number	Maximum unused time that a worker can transfer to the next accrual term.
adjustmentvalues	Number_Number	Array of balance adjustments that when returned by the formula is applied against the accrual plan enrollment.
adjustmentdates	Date_Number	Array of dates associated with the array of balance adjustment values to specify the dates on which each of the balance adjustments need to be applied.
adjustmenttypes	Text_Number	Array of adjustment reasons used for the balance adjustments.
absvalues	Number_Number	Array of absence accrual usages that can be applied against an accrual plan enrollment. Useful for cases where the absence entry itself is not captured in Global Absence Management, but rather via other modules (such as payroll element entries) or even third party systems and the absence deduction against the accrual plan needs to happen in HCM Cloud. When using this, it should be noted that absence entries should not be made for the same dates.
absdates	Date_Number	Array of dates associated with the absence accrual usages on which the deduction is applied.
accrualCeiling	Number	Maximum time that a worker can accrue

## Sample Formula

**Requirement:** The organization has an accrual plan for Vacation where-in workers accrue a set number of days every year depending on the grade level that they are in. If the worker changes grade within a year, then the accrual needs to be prorated depending on the days spent in each grade. Also, the total accrual value needs to be prorated by FTE and rounded to two decimal places.

**Solution:** A Global Absence Accrual Matrix formula such as the one below can be used.

FORMULA NAME: ANC\_ACCMAT

**FORMULA TYPE:** Global Absence Accrual Matrix Formula**DESCRIPTION:** This formula returns the prorated accrual value for Vacation plan with band change proration.

```
DEFAULT FOR IV_CALENDARSTARTDATE IS '4712/12/31 00:00:00' (date)
DEFAULT FOR IV_CALEDARENDDATE IS '4712/12/31 00:00:00' (date)
DEFAULT FOR IV_EVENT_DATES IS EMPTY_DATE_NUMBER
DEFAULT FOR IV_ACCRUAL_VALUES IS EMPTY_NUMBER_NUMBER
DEFAULT FOR PER_ASG_FTE_VALUE IS 1

INPUTS ARE IV_ACCRUAL, IV_EVENT_DATES, IV_ACCRUAL_VALUES, IV_CALENDARSTARTDATE (date), IV_CALEDARENDDATE (date)

i = 1
j = 2
ln_accrual_total = 0
WHILE IV_EVENT_DATES.EXISTS(j)
LOOP
(
  ln_accrual_total = ln_accrual_total + ( (DAYS_BETWEEN(IV_EVENT_DATES[j], IV_EVENT_DATES[i]) + 1) *
  IV_ACCRUAL_VALUES[i] )
  i = i+1
  j = j+1
)

ln_term_duration = DAYS_BETWEEN(IV_CALEDARENDDATE, IV_CALENDARSTARTDATE) + 1
ln_accrual = PER_ASG_FTE_VALUE * (ln_accrual_total / ln_term_duration)

accrual = ROUND(ln_accrual, 2)

RETURN accrual
```

## Global Absences Annual Accrual Limit

The Global Absences Annual Accrual Limit fast formula is used to specify an annual accrual limit for each worker (each enrolment) based on custom criteria.

An organization might have an annual accrual limit rule that generally allows a worker in an accrual plan to accrue a maximum of 30 days. However, the workers in a particular department accrue an additional 5 days due to the nature of their work. In such cases, this logic can be composed into the fast formula so that when the accrual process determines the annual accrual limit, the application dynamically allocates different limits to different workers depending on their department.

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

## Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Accrual**.
4. Click **Continue**.
5. On the Edit Absence Plan page Accruals tab Plan Limits section, select **Formula** in the **Annual Accrual Limit Rule** field.

6. Select the defined formula from the **Limit Formula** field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_ACCRUAL	Number	Accrual value received from the accrual matrix.
IV_CARRYOVER	Number	Carryover limit received from accrual matrix.

Variable Name	Data Type	Description
IV_CEILING	Number	Ceiling limit received from the accrual matrix.
IV_ACCRUAL_CEILING	Number	Annual accrual limit received from the accrual matrix.
IV_ACCRUALPERIODSTARTDATE	Date	Start date of accrual period.
IV_ACCRUALPERIODENDDATE	Date	End date of accrual period.
IV_CALENDARSTARTDATE	Date	Start date of accrual calendar year.
IV_CALEDARENDDATE	Date	End date of accrual calendar year.
IV_PLANENROLLMENTSTARTDATE	Date	Start date of enrollment.
IV_PLANENROLLMENTENDDATE	Date	End date of enrollment.

## Return Variables

Variable Name	Data Type	Description
ACCUALCEILING	Number	Maximum time that a worker can accrue in annual year.

## Sample Formula

**Requirement:** The organization has an annual accrual limit rule, which allows workers to accrue up to a maximum of 30 days in an annual year. However, workers belonging to business unit 'OBU1' accrue up to a maximum of 40 days in an annual year.

**Solution:** A Global Absence Annual Accrual Limit formula such as the one below can be used.

FORMULA NAME: ANC\_ANNUALCEIL

FORMULA TYPE: Global Absences Annual Accrual Limit

DESCRIPTION: This formula returns the annual accrual limit for workers enrolled into Vacation plan differentiated based on their business unit

```

DEFAULT FOR PER_ASG_BUSINESS_UNIT_NAME IS ' '
accrualceiling = 30
IF (PER_ASG_BUSINESS_UNIT_NAME = 'OBU1')
THEN
(ACCUALCEILING = 40)
RETURN accrualceiling

```

# Global Absences Annual Accrual Limit Proration

The Global Absence Annual Accrual Limit Proration fast formula can be used in cases where a proration factor (or a multiplication factor) needs to be applied onto the annual accrual limit rule.

An organization has a rule that asks for the annual accrual limit to be prorated based on FTE value of the worker. In such a case, after defining the annual accrual limit rule, the annual accrual limit proration rule can be specified using this formula to return a proration factor, which is multiplied onto the annual accrual limit before returning the final value against the worker's enrollment data.

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

## Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Accrual**.
4. Click **Continue**.
5. On the Edit Absence Plan page Accruals tab Plan Limits section, select a value other than **None** in the **Annual Accrual Limit Rule** field.
6. After defining the Annual Accrual Limit Rule as desired, select **Formula** in the **Limit Proration Rule** field.
7. Select the defined formula from the **Limit Proration Formula** field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number

Context Name	Data Type
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_ACCRUAL	Number	Accrual value received from the accrual matrix
IV_CARRYOVER	Number	Carryover limit received from accrual matrix
IV_CEILING	Number	Ceiling limit received from the accrual matrix
IV_ACCRUAL_CEILING	Number	Annual accrual limit received from the accrual matrix
IV_ACCRUALPERIODSTARTDATE	Date	Start date of accrual period
IV_ACCRUALPERIODENDDATE	Date	End date of accrual period
IV_CALENDARSTARTDATE	Date	Start date of accrual calendar year
IV_CALEDARENDDATE	Date	End date of accrual calendar year
IV_PLANENROLLMENTSTARTDATE	Date	Start date of enrollment
IV_PLANENROLLMENTENDDATE	Date	End date of enrollment

## Return Variables

Variable Name	Data Type	Description
PRORATIONFACTOR	Number	Multiplication factor for annual accrual ceiling limit

## Sample Formula

**Requirement:** The organization has an annual accrual limit rule, which allows workers to accrue up to a maximum of 35 days. This Annual year accrual limit needs to be prorated by FTE of workers who have the jobs 'Wealth Management' or 'Trade Associate' assigned against their assignment record.

**Solution:** A Global Absences Annual Accrual Limit Proration fast formula such as the one below can be used.

FORMULA NAME: ANC\_ANNUALCEILPRORAT

FORMULA TYPE: Global Absences Annual Accrual Limit Proration

DESCRIPTION: This formula returns the proration factor for the annual accrual ceiling limit for the absence plan

```
DEFAULT FOR PER_ASG_FTE_VALUE IS 1
DEFAULT FOR PER_ASG_JOB_NAME IS ' '

PRORATIONFACTOR = 1

IF (PER_ASG_JOB_NAME = 'Wealth Management Consultant' OR PER_ASG_JOB_NAME = 'Trade Associate')
THEN
  (PRORATIONFACTOR = ROUND(PER_ASG_FTE_VALUE,2))

RETURN PRORATIONFACTOR
```

## Global Absence Carryover

The Global Absence Carryover fast formula can be used in cases where a single carryover rule does not apply to the entire population that belongs to the accrual plan.

An organization might have a carryover rule that generally allows a maximum of 5 days to be carried over. However, the workers in a particular department are allowed to carryover an additional 2 days due to the nature of their work. In such cases, this logic can be composed into the fast formula so that when carryover is calculated, the application dynamically allocates different carryover limits to different workers depending on their department.

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

## Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.

2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Accrual**.
4. Click **Continue**.
5. On the Edit Absence Plan page Accruals tab Plan Limits section, select **Formula** in the **Carryover Rule** field.
6. Select the defined formula from the Carryover Formula field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_ACCRUAL	Number	Accrual value received from the accrual matrix
IV_CARRYOVER	Number	Carryover limit received from accrual matrix
IV_CEILING	Number	Ceiling limit received from the accrual matrix
IV_ACCRUALPERIODSTARTDATE	Date	Start date of accrual period
IV_ACCRUALPERIODENDDATE	Date	End date of accrual period
IV_CALENDARSTARTDATE	Date	Start date of accrual calendar year
IV_CALEDARENDDATE	Date	End date of accrual calendar year
IV_PLANENROLLMENTSTARTDATE	Date	Start date of enrollment
IV_PLANENROLLMENTENDDATE	Date	End date of enrollment
IV_ACCRUAL_CEILING	Number	Annual accrual limit from the accrual matrix

## Return Variables

Variable Name	Data Type	Description
carryover	Number	Maximum unused time that a worker can transfer to the next accrual

## Sample Formula

**Requirement:** The organization has a carryover rule which allows workers to carry forward only 5 days of their vacation balance into the next year. Workers belonging to business unit 'OBU1' work in shifts and projects which prevent them from being able to utilize all their annual vacation days on time. Hence, as an exception, workers in this BU are allowed to carry forward an additional 2 days on top of the 5 days.

**Solution:** A Global Absence Carryover formula such as the one below can be used.

FORMULA NAME: ANC\_CRRYOV

FORMULA TYPE: Global Absence Carryover

DESCRIPTION: This formula returns the carryover limit for workers enrolled into Vacation plan differentiated based on their business unit

```
DEFAULT FOR PER_ASG_BUSINESS_UNIT_NAME IS ''  
carryover = 5
```

```
IF (PER_ASG_BUSINESS_UNIT_NAME = 'OBU1')
THEN
  (carryover = 7)

RETURN carryover
```

## Global Absence Carryover Proration

The Global Absence Carryover Proration fast formula can be used in cases where a proration factor (or a multiplication factor) needs to be applied onto the maximum carryover limit.

An organization might have a rule which asks for the carryover to be prorated based on FTE or even their job. In such a case, after the carryover rule is defined, the carryover proration rule can be composed to return a proration factor which will be multiplied onto the carryover amount before returning the final value against the worker's enrollment data.

### Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

### Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Accrual**.
4. Click **Continue**.
5. On the Edit Absence Plan page Accruals tab Plan Limits section, select a value other than **None** in the **Carryover Rule** field.
6. After defining the carryover rule as desired, select **Formula** in the **Carryover Proration** field.
7. Select the defined formula from the **Carryover Proration Formula** field

### Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number

Context Name	Data Type
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_ACCRUAL	Number	Accrual value received from the accrual matrix
IV_CARRYOVER	Number	Carryover limit received from accrual matrix
IV_CEILING	Number	Ceiling limit received from the accrual matrix
IV_ACCRUALPERIODSTARTDATE	Date	Start date of accrual period
IV_ACCRUALPERIODENDDATE	Date	End date of accrual period
IV_CALENDARSTARTDATE	Date	Start date of accrual calendar year
IV_CALEDARENDDATE	Date	End date of accrual calendar year
IV_PLANENROLLMENTSTARTDATE	Date	Start date of enrollment
IV_PLANENROLLMENTENDDATE	Date	End date of enrollment
IV_ACCRUAL_CEILING	Number	Annual accrual limit from the accrual matrix

## Return Variables

Variable Name	Data Type	Description
prorationFactor	Number	Multiplication factor for carryover

## Sample Formula

**Requirement:** The organization has a carryover rule which allows workers to carry forward only 5 days of their vacation balance into the next year. This carryover needs to be prorated by FTE for workers who have the jobs 'Wealth Management' or 'Trade Associate' assigned against their assignment record.

**Solution:** A Global Absence Carryover Proration formula such as the one below can be used.

FORMULA NAME: ANC\_CRRYOVPRORAT

FORMULA TYPE: Global Absence Carryover Proration

DESCRIPTION: This formula returns the proration factor for the carryover limit for the absence plan

```
DEFAULT FOR PER_ASG_FTE_VALUE IS 1
DEFAULT FOR PER_ASG_JOB_NAME IS ' '

prorationFactor = 1

IF (PER_ASG_JOB_NAME = 'Wealth Management Consultant' OR PER_ASG_JOB_NAME = 'Trade Associate')
THEN
  (prorationFactor = ROUND(PER_ASG_FTE_VALUE,2))

RETURN prorationFactor
```

## Global Absence Ceiling

The Global Absence Ceiling fast formula can be used in cases where a single ceiling rule does not apply to the entire population that belongs to the accrual plan.

An organization might have a ceiling rule that generally allows a maximum of 30 days to be accrued by a worker in a plan. However, the workers in a particular department are allowed to accrue an additional 5 days due to the nature of their work. In such cases, this logic can be composed into the fast formula so that when ceiling limit is determined, the application dynamically allocates different limits to different workers depending on their department.

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

## Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.

3. In the Plan Type field, select **Accrual**.
4. Click **Continue**.
5. On the Edit Absence Plan page Accruals tab Plan Limits section, select **Formula** in the **Ceiling Rule** field.
6. Select the defined formula from the **Ceiling Formula** field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_ACCRUAL	Number	Accrual value received from the accrual matrix

Variable Name	Data Type	Description
IV_CARRYOVER	Number	Carryover limit received from accrual matrix
IV_CEILING	Number	Ceiling limit received from the accrual matrix
IV_ACCRUALPERIODSTARTDATE	Date	Start date of accrual period
IV_ACCRUALPERIODENDDATE	Date	End date of accrual period
IV_CALENDARSTARTDATE	Date	Start date of accrual calendar year
IV_CALEDARENDDATE	Date	End date of accrual calendar year
IV_PLANENROLLMENTSTARTDATE	Date	Start date of enrollment
IV_PLANENROLLMENTENDDATE	Date	End date of enrollment
IV_ACCRUAL_CEILING	Number	Annual accrual limit from the accrual matrix

## Return Variables

Variable Name	Data Type	Description
ceiling	Number	Maximum time that a worker can accrue

## Sample Formula

**Requirement:** The organization has a ceiling rule which allows workers to accrue up to a maximum of 30 days. Workers belonging to business unit 'OBU1' work in shifts and projects which prevent them from always being able to utilize all their annual vacation days on time. Hence, as an exception, workers in this BU are allowed to accrue up to a maximum of 40 days in total.

**Solution:** A Global Absence Ceiling formula such as the one below can be used.

FORMULA NAME: ANC\_CEIL

FORMULA TYPE: Global Absence Ceiling

DESCRIPTION: This formula returns the ceiling limit for workers enrolled into Vacation plan differentiated based on their business unit

```
DEFAULT FOR PER_ASG_BUSINESS_UNIT_NAME IS ''
ceiling = 35
IF (PER_ASG_BUSINESS_UNIT_NAME = 'OBU1')
THEN
  (ceiling= 40)
RETURN ceiling
```

# Global Absence Ceiling Proration

The Global Absence Ceiling Proration fast formula can be used in cases where a proration factor (or a multiplication factor) needs to be applied onto the maximum ceiling limit.

An organization might have a rule which asks for the ceiling limit to be prorated based on FTE or even their job. In such a case, after the ceiling rule is defined, the ceiling proration rule can be composed to return a proration factor which will be multiplied onto the ceiling limit before returning the final value against the worker's enrollment data.

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

## Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Accrual**.
4. Click **Continue**.
5. On the Edit Absence Plan page Accruals tab Plan Limits section, select a value other than **None** in the **Ceiling Rule** field.
6. After defining the ceiling rule as desired, select **Formula** in the **Ceiling Proration** field.
7. Select the defined formula from the **Ceiling Proration Formula** field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number

Context Name	Data Type
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_ACCRUAL	Number	Accrual value received from the accrual matrix
IV_CARRYOVER	Number	Carryover limit received from accrual matrix
IV_CEILING	Number	Ceiling limit received from the accrual matrix
IV_ACCRUALPERIODSTARTDATE	Date	Start date of accrual period
IV_ACCRUALPERIODENDDATE	Date	End date of accrual period
IV_CALENDARSTARTDATE	Date	Start date of accrual calendar year
IV_CALEDARENDDATE	Date	End date of accrual calendar year
IV_PLANENROLLMENTSTARTDATE	Date	Start date of enrollment
IV_PLANENROLLMENTENDDATE	Date	End date of enrollment
IV_ACCRUAL_CEILING	Number	Annual accrual limit from the accrual matrix

## Return Variables

Variable Name	Data Type	Description
prorationFactor	Number	Multiplication factor for ceiling

## Sample Formula

**Requirement:** The organization has a ceiling rule which allows workers to accrue upto a maximum of 35 days. This ceiling limit needs to be pro-rated by FTE for workers who have the jobs 'Wealth Management' or 'Trade Associate' assigned against their assignment record.

**Solution:** A Global Absence Ceiling Proration formula such as the one below can be used.

FORMULA NAME: ANC\_CEILPRORAT

FORMULA TYPE: Global Absence Ceiling Proration

DESCRIPTION: This formula returns the proration factor for the ceiling limit for the absence plan

```
DEFAULT FOR PER_ASG_FTE_VALUE IS 1
DEFAULT FOR PER_ASG_JOB_NAME IS ''

prorationFactor = 1

IF (PER_ASG_JOB_NAME = 'Wealth Management Consultant' OR PER_ASG_JOB_NAME = 'Trade Associate')
THEN
  (prorationFactor = ROUND(PER_ASG_FTE_VALUE,2))

RETURN prorationFactor
```

## Global Absence Discretionary Disbursement Rule

This formula type can be used to define the limit of the amount of leave balance that a worker is eligible to request to be disbursed as cash. The formula returns the minimum and maximum amount of leave balance that can be disbursed, and also returns an increment, which limits the disbursement amount to specific values between the minimum and maximum. These attributes are defined as follows.

- The minimum amount of leave balance that the worker is eligible to request a cash disbursement.
- The maximum amount of leave balance that the worker can request, for instances where you want to limit the amount to a certain number of hours.
- An increment to prevent workers from entering decimal places, and thus avoid rounding issues later.

Note that the output should always be in the absence plan's UOM. Returning negative numbers is not recommended because this could lead to unexpected behavior while processing the transaction.

You have the option to enter minimum, maximum and increments as flat amounts. But if the limits vary by certain criteria like the worker's grade, years of service, hours, then this formula type can be used to meet these requirements.

The organization has a vacation plan from which employees can choose to disburse some of the accrued vacation balance as cash. For employees with 5 years of service or less, the maximum amount that can be disbursed is 5 days. For employees that have more than 5 years of service, the maximum amount that can be disbursed increases to 10 days.

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

## Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Accrual**.
4. Click **Continue**.
5. On the Edit Absence Plan page **Entries and Balances** tab **Discretionary Disbursements** section, select **Formula** in the **Disbursement Rule** field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number

Context Name	Data Type
START_DATE	Date

## Input Values

No input values.

## Return Variables

Variable Name	Data Type	Description
MIN	Number	The minimum amount that can be disbursed
MAX	Number	The maximum amount that can be disbursed
INCREMENT	Number	The increments that will be allowed

## Sample Formula

**Requirement:** The organization has a vacation plan from which employees can choose to disburse some of the accrued vacation balance as cash. For employees with 5 years of service or less, the maximum amount that can be disbursed is 5 days. For employees that have more than 5 years of service, the maximum amount that can be disbursed increases to 10 days.

**Solution:** A Global Absence Discretionary Disbursement Rule formula such as the one below can be used.

FORMULA NAME: ANC\_DISBURSEMENTRULE

FORMULA TYPE: Global Absence Discretionary Disbursement Rule

DESCRIPTION: This formula dictates what minimum / maximum / increments are allowed

```

DEFAULT FOR PER_PERSON_ENTERPRISE_HIRE_DATE IS '0001/01/01 00:00:00' (date)
DEFAULT FOR PER_REL_ORIGINAL_DATE_OF_HIRE IS '0001/01/01 00:00:00' (date)

MIN=1
MAX=5
INCREMENT=1

IF(PER_REL_ORIGINAL_DATE_OF_HIRE WAS DEFAULTED) THEN
  (L_Hire_Date = PER_PERSON_ENTERPRISE_HIRE_DATE)
ELSE
  (L_Hire_Date = PER_REL_ORIGINAL_DATE_OF_HIRE)

L_Eff_Date = GET_CONTEXT(EFFECTIVE_DATE, '4712/12/31 00:00:00' (date))
Length_of_service = DAYS_BETWEEN(L_Eff_Date,L_Hire_Date)/365
IF(Length_of_service > 5) THEN (
  MAX = 10
)

RETURN MIN,MAX,INCREMENT

```

# Global Absence Elective Disbursement Evaluation Formula

The Global Absence Elective Disbursement Evaluation formula is used to validate elective disbursements in an accrual plan and set payment percentages for the disbursement. The formula can also be used to override the disbursement amount elected by the worker.

An organization might have a rule such that if an employee's accrual balance as of the election date is below a certain minimum threshold limit, then the disbursement should not be allowed. Whereas, if the balance is between two other threshold values, then the payment percentage should be 75%. In such cases, you can compose a Global Absence Elective Disbursement Evaluation formula and attach it to the accrual plan.

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

## Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Accrual**.
4. Click **Continue**.
5. On the Edit Absence Plan page Accruals tab Plan Limits section, select a value other than **None** in the **Carryover Rule field**.
6. After defining the carryover rule as desired, select **Formula** in the **Carryover Proration field**
7. Select the defined formula from the **Carryover Proration Formula** field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number

Context Name	Data Type
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_ACCRUAL	Number	Accrual value received from the accrual matrix
IV_CARRYOVER	Number	Carryover limit received from accrual matrix
IV_CEILING	Number	Ceiling limit received from the accrual matrix
IV_ACCRUALPERIODSTARTDATE	Date	Start date of accrual period
IV_ACCRUALPERIODENDDATE	Date	End date of accrual period
IV_CALENDARSTARTDATE	Date	Start date of accrual calendar year
IV_CALEDARENDDATE	Date	End date of accrual calendar year
IV_PLANENROLLMENTSTARTDATE	Date	Start date of enrollment
IV_PLANENROLLMENTENDDATE	Date	End date of enrollment
IV_BAND_CHG_DT1*	Date	If the accrual matrix band definition consists of grades or derived factors length of service or age, then this input value fetches the date on which worker has had a change in any of these factors within the period.

Variable Name	Data Type	Description
IV_BAND_CHG_BEFVAL1*	Number	Accrual value as per the matrix before IV_BAND_CHG_DT1
IV_BAND_CHG_AFTVAL1*	Number	Accrual value as per the matrix after IV_BAND_CHG_DT1
IV_EVENT_DATES	Date_Number	Array of dates returned by the Accrual Event formula.
IV_ACCRUAL_VALUES	Number_Number	Array of accrual values fetched from the accrual matrix as of the dates contained in the accrual event dates array.
IV_ACCRUAL_CEILING	Number	Annual accrual limit from the accrual matrix
IV_DISB_TYPE	Text	The disbursement type for current record which is being processed. Possible values: CSH / ORA_ANC_CSH_YREND / ORA_ANC_ELECT_DISB
IV_DISB_VALUE	Number	The disbursement value for current record which is being processed
IV_DISB_ELECTED_AMT	Number	Elected amount for the elective disbursement
IV_DISB_ELECTION_DATE	Date	Election date for the elective disbursement
IV_DISB_PROCD_DATE	Date	Exact processed date for the elective disbursement
IV_DISB_ELECT_PENDING	Text	If the elective disbursement is marked as pending or not. "Y" stands for pending, "N" stands for not pending.

## Return Variables

Variable Name	Data Type	Description
paymentPercentage	Number	Payment percentage for the elective disbursement value
overrideValue	Number	Override value for elective disbursement

## Sample Formula

**Requirement:** If the current balance of an employee is below 8 hours, then 0 hours are to be disbursed.

**Solution:** A Global Absence Elective Disbursement Evaluation formula such as the one below can be used.

FORMULA NAME: ANC\_ELECTDISBEVAL

FORMULA TYPE: Global Absence Elective Disbursement Evaluation

DESCRIPTION: This formula validates the elective disbursement values for an accrual plan.

```
DEFAULT FOR IV_DISB_TYPE IS 'XX'  
DEFAULT FOR IV_DISB_VALUE IS -1  
INPUTS ARE IV_DISB_TYPE, IV_DISB_VALUE, IV_ACCRUAL_CEILING  
  
ln_plan_balance = GET_PLAN_BALANCE('Vision Accrual Plan')  
  
paymentPercentage = 100  
overrideValue = IV_DISB_VALUE  
  
IF (IV_DISB_TYPE = 'ORA_ANC_ELECT_DISB') THEN  
(  
IF (ln_plan_balance <5) THEN  
(  
paymentPercentage = 0  
overrideValue = 0  
)  
)  
RETURN paymentPercentage,overrideValue
```

## Global Absence Discretionary Donation Rule

This formula type can be used to define a limit for the amount of leave balance that a worker is eligible to donate to another worker. This formula returns the minimum and maximum amount of leave balance that can be donated, and also returns an increment, which limits the donation amount to specific values between the minimum and maximum. These attributes are defined as follows

- The minimum amount of leave balance that the worker is eligible to donate.
- The maximum amount of leave balance that the worker can donate, for instances where you want to limit the amount to a certain number of hours
- An increment to prevent workers from entering decimal places, and thus avoid rounding issues later

Note that the output should always be in the absence plan's UOM. Returning negative numbers is not recommended because this could lead to unexpected behavior while processing the transaction.

You have the option to enter minimum, maximum and increments as flat amounts. But if the limits vary by certain criteria like the worker's grade, years of service, hours, then this formula type can be used to meet these requirements.

The organization has a vacation plan from which employees can choose to donate some of the accrued vacation balance to another worker who has been enrolled into a donation plan. For employees with 5 years of service or less, the maximum amount that can be donated is 5 days. For employees that have more than 5 years of service, the maximum amount that can be donated increases to 10 days.

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

## Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Accrual**.
4. Click **Continue**.
5. On the Edit Absence Plan page **Entries and Balances** tab **Discretionary Disbursements** section, select **Formula** in the **Disbursement Rule** field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number

Context Name	Data Type
START_DATE	Date

## Input Values

No input values.

## Return Variables

Variable Name	Data Type	Description
MIN	Number	The minimum amount that can be disbursed
MAX	Number	The maximum amount that can be disbursed
INCREMENT	Number	The increments that will be allowed

## Sample Formula

**Requirement:** The organization has a vacation plan from which employees can choose to donate some of the accrued vacation balance to another worker who has been enrolled into a donation plan. For employees with 5 years of service or less, the maximum amount that can be donated is 5 days. For employees that have more than 5 years of service, the maximum amount that can be donated increases to 10 days.

**Solution:** A Global Absence Discretionary Donation Rule formula such as the one below can be used.

FORMULA NAME: ANC\_DONATIONRULE

FORMULA TYPE: Global Absence Discretionary Donation Rule

DESCRIPTION: This formula dictates what minimum / maximum / increments are allowed

```

DEFAULT FOR PER_PERSON_ENTERPRISE_HIRE_DATE IS '0001/01/01 00:00:00' (date)
DEFAULT FOR PER_REL_ORIGINAL_DATE_OF_HIRE IS '0001/01/01 00:00:00' (date)

MIN=1
MAX=5
INCREMENT=1

IF(PER_REL_ORIGINAL_DATE_OF_HIRE WAS DEFAULTED) THEN
  (L_Hire_Date = PER_PERSON_ENTERPRISE_HIRE_DATE)
ELSE
  (L_Hire_Date = PER_REL_ORIGINAL_DATE_OF_HIRE)

L_Eff_Date = GET_CONTEXT(EFFECTIVE_DATE, '4712/12/31 00:00:00' (date))
Length_of_service = DAYS_BETWEEN(L_Eff_Date,L_Hire_Date)/365
IF(Length_of_service > 5) THEN (
  MAX = 10
)

```

# Global Absence Partial Period Accrual Rate

The Global Absence Partial Period Accrual Rate fast formula is where any logic required for proration of accrual balance during enrollment year and disenrollment year needs to be entered.

If the annual accrual that a worker is eligible for every year is 20 days and the worker has enrolled into the plan mid year, the organization would like to grant the worker 10 days for the year of enrollment since the worker is eligible only for half the year. Similarly, if a worker disenrolls from a plan mid year, the total accrual for that year would need to be reduced from 20 to 10 because the worker was enrolled into the plan for only half the year.

This formula is invoked when enrollment or disenrollment dates fall within the repeating period for which the accrual is being processed.

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

## Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Accrual**.
4. Click **Continue**.
5. On the Edit Absence Plan page Accruals tab Accrual Attributes section, select the desired formula in the **Partial Accrual Period Formula** field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number

Context Name	Data Type
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_ACCRUAL	Number	Accrual value received from the accrual matrix
IV_CARRYOVER	Number	Carryover limit received from accrual matrix
IV_CEILING	Number	Ceiling limit received from the accrual matrix
IV_ACCRUALPERIODSTARTDATE	Date	Start date of accrual period
IV_ACCRUALPERIODENDDATE	Date	End date of accrual period
IV_CALENDARSTARTDATE	Date	Start date of accrual calendar year
IV_CALEDARENDDATE	Date	End date of accrual calendar year
IV_PLANENROLLMENTSTARTDATE	Date	Start date of enrollment
IV_PLANENROLLMENTENDDATE	Date	End date of enrollment
IV_BAND_CHG_DT1*	Date	If the accrual matrix band definition consists of grades or derived factors length of service or age, then this input value fetches the date on which worker has had a change in any of these factors within the period.
IV_BAND_CHG_BEFVAL1*	Number	Accrual value as per the matrix before IV_BAND_CHG_DT1
IV_BAND_CHG_AFTVAL1*	Number	Accrual value as per the matrix after IV_BAND_CHG_DT1
IV_ACCRUAL_CEILING	Number	Annual accrual limit from the accrual matrix

\*the sequence for these input values can be extended to 5.

## Return Variables

Variable Name	Data Type	Description
accrual	Number	Partial period prorated accrual value returned for the particular partial repeating period. Value is returned gets added as an accrual line. For processing termination of enrollment for front loaded plans, the variable should return a negative value, to deduct prorated amount from the existing annual accrual value.

## Sample Formula

**Requirement:** The organization has a partial period rule where accruals are prorated based on months enrolled for the plan.

**Solution:** A Global Absence Partial Period Accrual Rate formula such as the one below can be used.

FORMULA NAME: ANC\_PARACC

FORMULA TYPE: Global Absence Partial Period Accrual Rate Formula

DESCRIPTION: This formula returns the accrual for mid period enrollments and unenrollments for Vacation absence plan

```

DEFAULT FOR IV_ACCRUAL IS 0
DEFAULT FOR IV_ACCRUALPERIODSTARTDATE IS '4712/12/31 00:00:00' (date)
DEFAULT FOR IV_ACCRUALPERIODENDDATE IS '4712/12/31 00:00:00' (date)
DEFAULT FOR IV_CALENDARSTARTDATE IS '4712/12/31 00:00:00' (date)
DEFAULT FOR IV_CALEDARENDDATE IS '4712/12/31 00:00:00' (date)
DEFAULT FOR IV_PLANENROLLMENTSTARTDATE IS '4712/12/31 00:00:00' (date)
DEFAULT FOR IV_PLANENROLLMENTENDDATE IS '4712/12/31 00:00:00' (date)

INPUTS ARE IV_ACCRUAL, IV_ACCRUALPERIODSTARTDATE (date), IV_ACCRUALPERIODENDDATE (date), IV_CALENDARSTARTDATE (date), IV_CALEDARENDDATE (date), IV_PLANENROLLMENTSTARTDATE (date), IV_PLANENROLLMENTENDDATE (date)

ln_same_year_duration = 0
lc_first_month_flag = 'N'
ln_first_month_accrual = 0
ln_participation_duration = 0
ln_term_duration = 1 + (TO_NUMBER(TO_CHAR(IV_CALEDARENDDATE, 'yy')) - TO_NUMBER(TO_CHAR(IV_CALENDARSTARTDATE, 'yy'))) * 12 + (TO_NUMBER(TO_CHAR(IV_CALEDARENDDATE, 'mm')) - TO_NUMBER(TO_CHAR(IV_CALENDARSTARTDATE, 'mm'))))
ld_start_date = GREATEST (IV_PLANENROLLMENTSTARTDATE, IV_CALENDARSTARTDATE)
ld_end_date = LEAST (IV_PLANENROLLMENTENDDATE, IV_CALEDARENDDATE)

/*Calculating the duration for which the enrollment was active*/
ln_participation_duration = (TO_NUMBER(TO_CHAR(ld_end_date, 'yy')) - TO_NUMBER(TO_CHAR(ld_start_date, 'yy'))) * 12 + (TO_NUMBER(TO_CHAR(ld_end_date, 'mm')) - TO_NUMBER(TO_CHAR(ld_start_date, 'mm')))) + 1

/*Capturing this in case un-enrollment occurred in the same month as calendar start month*/
ln_first_month_accrual = IV_ACCRUAL * (ln_participation_duration / ln_term_duration)

/*Checking if un-enrollment happened before end of calendar year*/
IF IV_PLANENROLLMENTENDDATE < IV_CALEDARENDDATE
THEN
(

```

```

/*Checking if enrollment also happened within the same year as unenrollment*/
IF (IV_PLANENROLLMENTSTARTDATE > IV_CALENDARSTARTDATE)
THEN
(
/*If enrollment happened in the same year as the un-enrollment, calculating the duration from calendar
start date to enrollment start date so that this can be subtracted from total*/
ln_same_year_duration = (TO_NUMBER(TO_CHAR(IV_PLANENROLLMENTSTARTDATE, 'yy')) -
TO_NUMBER(TO_CHAR(IV_CALENDARSTARTDATE, 'yy'))) * 12 + (TO_NUMBER(TO_CHAR(IV_PLANENROLLMENTSTARTDATE, 'mm')) -
TO_NUMBER(TO_CHAR(IV_CALENDARSTARTDATE, 'mm')))

)
/*The termination pro-rata has to return a negative value - the amount that needs to be subtracted from
the current balance*/
ln_participation_duration = -1 * (ln_term_duration - ln_participation_duration - ln_same_year_duration)
/*Checking if un-enrollment happened in the first month of the calendar year*/
IF ( (TO_CHAR(IV_CALENDARSTARTDATE, 'mm') = TO_CHAR(IV_PLANENROLLMENTENDDATE, 'mm')) AND
(TO_CHAR(IV_CALENDARSTARTDATE, 'yyyy') = TO_CHAR(IV_PLANENROLLMENTENDDATE, 'yyyy')) )
THEN
(lc_first_month_flag = 'Y')
)

prorated_accrual = IV_ACCRUAL * (ln_participation_duration / ln_term_duration)
/*If un-enrollment happened in the first month of the calendar year, then return the pro-rata for the
single month*/
IF (lc_first_month_flag = 'Y')
THEN
(prorated_accrual = ln_first_month_accrual)

accrual = ROUND(prorated_accrual,2)

RETURN accrual

```

## Global Absence Plan Duration

The Global Absence Plan Duration fast formula is used to override the default duration calculation logic for daily accrual duration entries against accrual plans. For example, if the accrual deduction to be considered for an absence entry in an accrual plan of an organization depends on the location of the worker, then this formula can be leveraged to specify this dynamic calculation logic. This formula is invoked once for each day of absence. If there are multiple shifts in a day, then the formula is invoked once for each shift. If the shift spans over a day divide, then the shift is split into two and the formula is invoked once for each shift block.

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

## Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Accrual**.
4. Click **Continue**.
5. On the Edit Absence Plan page **Plan Attributes** tab General Attributes section, select the defined formula from the **Conversion Formula** field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_ABS_START_DATE	Date	Start date: time of absence entry
IV_ABS_END_DATE	Date	End date: time of absence entry
IV_ABS_START_DURATION	Number	Start date duration of absence entry. Applicable when absence entry is being recorded against an Elapsed based work schedule

Variable Name	Data Type	Description
IV_ABS_END_DURATION	Number	End date duration of absence entry. Applicable when absence entry is being recorded against an Elapsed based work schedule
IV_START_DATE	Date	Start date of shift block within absence
IV_END_DATE	Date	End date of shift block within absence
IV_START_DURATION	Number	Start date duration of the shift block within absence entry. Applicable when absence entry is being recorded against an Elapsed based work schedule
IV_END_DURATION	Number	End date duration of the shift block within absence entry. Applicable when absence entry is being recorded against an Elapsed based work schedule
IV_START_TIME	Text	Start time on the start date of the shift block within absence entry. Applicable when absence entry is being recorded against a Time based work schedule. For elapsed work schedules it will be 00:00. If there are no shifts on the day being processed, then absence start time or 00:00 is passed, whichever is greater.
IV_END_TIME	Text	End time on the start date of the shift block within absence entry. Applicable when absence entry is being recorded against a Time based work schedule. For elapsed work schedules it will be 00:00. If there are no shifts on the day being processed, then absence end time or 00:00 is passed, whichever is greater.
IV_UOM	Text	Unit of measure
IV_ABSENCE_TYPE_ID	Number	Absence type ID for the absence being recorded.
IV_USER_MODE	Text	Identifies the role of the person recording the absence.

## Return Variables

Variable Name	Data Type	Description
Duration	Number	Duration of plan entries

## Sample Formula

**Requirement:** The organization has an accrual plan where the accrual usage is only 50% for workers who work in an office whose location is in California, whereas the rest of the workers the accrual usage is same as the absence duration.

**Solution:** A Global Absence Plan Duration formula such as the one below can be used.

FORMULA NAME: ANC\_PLN\_DUR

## FORMULA TYPE: Global Absence Plan Duration

DESCRIPTION: This formula evaluates the entitlement usage based on worker location

```
DEFAULT FOR IV_START_DATE IS '4712/12/31 00:00:00' (date)
DEFAULT FOR IV_END_DATE IS '4712/12/31 00:00:00' (date)
DEFAULT FOR IV_START_TIME IS '00:00'
DEFAULT FOR IV_END_TIME IS '23:59'
DEFAULT FOR PER_ASG_LOC_REGION2 IS 'XXXXXXXXXX'
INPUTS ARE IV_START_DATE (date), IV_END_DATE (date), IV_START_TIME(text), IV_END_TIME(text)

ln_entry_duration_d = 0
ln_unrounded_duration_d = 0

ld_period_start_date = to_date(to_char(IV_START_DATE, 'DD/MM/YYYY') || ' ' || IV_START_TIME, 'DD/MM/YYYY
HH24:MI:SS')
ld_period_end_date = to_date(to_char(IV_END_DATE, 'DD/MM/YYYY') || ' ' || IV_END_TIME, 'DD/MM/YYYY HH24:MI:SS')

ln_entry_duration_d = GET_PAY_AVAILABILITY ('ASSIGN',
ld_period_start_date,
ld_period_end_date,
'Y',
'Y',
'Y',
'Y',
'D')

IF (PER_ASG_LOC_REGION2 = 'CA')
THEN
(
  ln_entry_duration_d = ln_entry_duration_d * 0.5
)
DURATION = ROUND(ln_entry_duration_d,2)
RETURN DURATION
```

## Global Absence Plan Enrollment End

The Global Absence Plan Enrollment End fast formula can be used to over-ride the default enrollment end date rule for the absence plan when workers are being terminated from the organization or when the Update Accrual Plan Enrollments batch job is being run.

In an organization the absence plan un-enrollment rule could be such that for termination, workers have to serve a notice period of one month during which time the worker should not be enrolled into any absence plan. In such a case a Global Absence Plan Enrollment End formula can be composed to derive this alternate enrollment end date.

### Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

### Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Accrual**.
4. Click **Continue**.

5. On the Edit Absence Plan page Participation tab, Termination Rules section, select **Formula** in the **Enrollment End Rule** field.
6. Select the defined formula from the **End Date Formula** field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Return Variables

Variable Name	Data Type	Description
enrollmentEndDate	Date	End date of enrollment

## Sample Formula

**Requirement:** Upon entering of termination information, the worker needs to be un-enrolled from the plan, one month before the actual termination date (in this case, the event date).

**Solution:** A Global Absence Plan Enrollment End formula such as the one below can be used.

FORMULA NAME: ANC\_ENRMNTEND

FORMULA TYPE: Global Absence Plan Enrollment End

DESCRIPTION: This formula returns the Enrollment End Date for absence plan enrollments by subtracting 1 month from the termination date

```
enrollmentEndDate = GET_CONTEXT(EFFECTIVE_DATE, '4712/12/31 00:00:00' (date))
enrollmentEndDate = ADD_MONTHS(enrollmentEndDate, -1)
RETURN enrollmentEndDate
```

## Global Absence Plan Enrollment Start

The Global Absence Plan Enrollment Start fast formula can be used to override the default enrollment start date rule for the absence plan when workers are being hired into the organization or when the Update Accrual Plan Enrollments batch job is being run.

In an organization the absence plan enrollment rule can be configured such that only workers are allowed to enroll into the plan from the hire date, whereas Interns and Graduates have to complete a waiting period of 1 month before being enrolled into the plan. In such cases, the Plan Enrollment Start formula can be used to derive the alternate enrollment date (one that is different from the hire date or the date passed into the parameter when submitting the Update Accrual Plan Enrollments batch job).

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

## Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Accrual**.
4. Click **Continue**.
5. On the Edit Absence Plan page Participation tab, Enrollment Rules section, select **Formula** in the **Enrollment Start Rule** field.
6. Select the defined formula from the **Start Date Formula** field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Return Variables

Variable Name	Data Type	Description
enrollmentStartDate	Date	Start date of enrollment

## Sample Formula

**Requirement:** All new Interns and Graduates who are hired into the organization should get enrolled into the plan only after one month of employment. All other workers should get enrolled into the plan as of hire date.

**Solution:** A Global Absence Plan Enrollment Start formula such as the one below can be used.

FORMULA NAME: ANC\_ENRMNTSTART

## FORMULA TYPE: Global Absence Plan Enrollment Start

DESCRIPTION: This formula returns the Enrollment Start Date for absence plan enrollments by adding 1 month to the event date for Interns and Graduates

```
DEFAULT FOR PER_ASG_USER_PERSON_TYPE IS 'Worker'

enrollmentStartDate = GET_CONTEXT(EFFECTIVE_DATE, '4712/12/31 00:00:00' (date))

IF (PER_ASG_USER_PERSON_TYPE != 'Worker')
  THEN (enrollmentStartDate = ADD_MONTHS(enrollmentStartDate, 1))
RETURN enrollmentStartDate
```

## Global Absence Plan Period Anniversary Event Date

The Global Absence Plan Period Anniversary Event Date fast formula can be used to specify the date to be used as the calendar leave year start date for accrual plans whose term is of type Anniversary year.

If the calendar leave year for a vacation plan in an organization is based on the original hire date of a worker, then a formula of this type can be composed to return the required date to the plan calculations.

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

## Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Accrual**.
4. Click **Continue**.
5. On the Edit Absence Plan page Plan Attributes tab, Plan Term section, select **Anniversary year** in the **Type** field.
6. Then select **Formula** in the **Anniversary Event Rule** field.
7. Select the defined formula from the **Anniversary Event Formula** field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date

Context Name	Data Type
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Return Variables

Variable Name	Data Type	Description
anniversaryDate	Date	Date used to determine the leave year start date for anniversary based accrual plan term

## Sample Formula

**Requirement:** The organization has an accrual plan term that is based on the anniversary of the worker. If the person is a regular worker, then the leave year is to be based on the earliest start date of the Worker. If the person is an Intern or Graduate, then the leave year is to be based on the latest legal employer hire date.

**Solution:** A Global Absence Plan Period Anniversary Event Date formula such as the one below can be used.

FORMULA NAME: ANC\_PLN\_PRD\_ANNIV\_DATE

FORMULA TYPE: Global Absence Plan Period Anniversary Event Date

DESCRIPTION: This formula returns the calendar leave year start date for Anniversary year based on the person type

```

DEFAULT FOR PER_ASG_USER_PERSON_TYPE IS 'Worker'
DEFAULT FOR PER_PERSON_ENTERPRISE_HIRE_DATE IS '4712/12/31 00:00:00' (date)
DEFAULT FOR PER_ASG_REL_DATE_START IS '4712/12/31 00:00:00' (date)

IF (PER_ASG_USER_PERSON_TYPE = 'Worker')

```

```
THEN
(
anniversaryDate = PER_PERSON_ENTERPRISE_HIRE_DATE
)
ELSE
(
anniversaryDate = PER_ASg_REL_DATE_START
)

RETURN anniversaryDate
```

## Global Absence Plan Use Rate

The Global Absence Plan Use Rate fast formula can be used to dynamically specify the rate definition associated with the qualification plan depending on custom conditions.

If the rate definition associated with the same qualification plan varies depending on the location of the worker being evaluated, a Global Absence Plan Use Rate can be composed to associate the corresponding rate definition to the Worker.

### Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

### Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Qualification**.
4. Click **Continue**.
5. On the Edit Absence Plan page Entries and Balances tab, Rates section, choose **Formula** in the **Absence Payment Rule** field.
6. Select the defined formula in the **Formula** field.

### Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_CATEGORY_ID	Number
ABSENCE_ENTRY_ID	Number
ABSENCE_MATERNITY_ID	Number
ABSENCE_REASON_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number

Context Name	Data Type
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_ATTRIBUTE_CATEGORY	Text	Context of the DFF row for Absence Recordings DFF
IV_ATTRIBUTE_1*	Text	First text segment of the Absence Recordings DFF
IV_ATTRIBUTE_NUMBER1*	Number	First number segment of the Absence Recordings DFF
IV_ATTRIBUTE_DATE1*	Date	First date segment of the Absence Recordings DFF
IV_ATTRIBUTE_ARR	Text_Number	Array of text segments of the Absence Recordings DFF. The index indicates the segment number. Eg: IV_ATTRIBUTE_ARR[1] holds the value of segment ATTRIBUTE1 in

Variable Name	Data Type	Description
		ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_ATTRIBUTE_ARR.exists(1))
IV_ATTRIBUTE_NUMBER_ARR	Number_Number	Array of number segments of the Absence Recordings DFF. The index indicates the segment number. Eg: IV_ATTRIBUTE_NUMBER_ARR[1] holds the value of segment ATTRIBUTE_NUMBER1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_ATTRIBUTE_NUMBER_ARR.exists(1))
IV_ATTRIBUTE_DATE_ARR	Date_Number	Array of date segments of the Absence Recordings DFF. The index indicates the segment number. Eg: IV_ATTRIBUTE_DATE_ARR[1] holds the value of segment ATTRIBUTE_DATE1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_ATTRIBUTE_DATE_ARR.exists(1))
IV_INFORMATION_CATEGORY	Text	Context of the DDF row for ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_1*	Text	First text segment of the ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_NUMBER1*	Number	First number segment of the ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_DATE1*	Date	First date segment of the ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_ARR	Text_Number	Array of text segments of the ANC_PER_ABS_ENTRIES DDF. The index indicates the segment number. Eg: IV_INFORMATION_ARR[1] holds the value of segment INFORMATION1 in ANC_

Variable Name	Data Type	Description
		PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_INFORMATION_ARR.exists(1))
IV_INFORMATION_NUMBER_ARR	Number_Number	Array of number segments of the ANC_PER_ABS_ENTRIES DDF. The index indicates the segment number. Eg: IV_INFORMATION_NUMBER_ARR[1] holds the value of segment INFORMATION_NUMBER1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_INFORMATION_NUMBER_ARR.exists(1))
IV_INFORMATION_DATE_ARR	Date_Number	Array of date segments of the ANC_PER_ABS_ENTRIES DDF. The index indicates the segment number. Eg: IV_INFORMATION_DATE_ARR[1] holds the value of segment INFORMATION_DATE1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_INFORMATION_DATE_ARR.exists(1))
IV_PAYMENT_DTL_BAND	Number	Identifier of the payment band chosen for the absence linked to an agreement
IV_NOTIFICATION_DATE	Date	Date of illness/injury notification
IV_MATCHING_DATE	Date	Date on which child is matched with the worker for adoption

\*The sequence for this input value can be extended up to 5

## Return Variables

Variable Name	Data Type	Description
RATECODE	Text	Rate code for rate associated with qualification plan
DEBUG_MESSAGE	Text	Text passed into this output variable would get logged if application logging is enabled. Useful for debugging

## Sample Formula

**Requirement:** The organization has an absence payment rate rule for the qualification plan that depends on the hourly/salaried nature of the Worker. Separate rate definition rules exist for each category of workers.

**Solution:** A Global Absence Plan Use Rate formula such as the one below can be used.

FORMULA NAME: ANC\_PLN\_RATE

FORMULA TYPE: Global Absence Plan Use Rate

DESCRIPTION: This formula returns the rate code for the rate definition that needs to be associated with the qualification plan

```
DEFAULT FOR PER_ASG_HOURLY_SALARIED_CODE IS 'H'

IF (PER_ASG_HOURLY_SALARIED_CODE = 'H')
THEN
(
RATECODE = 'HOURLY_RATE'
)
ELSE
(
RATECODE = 'ANNUAL_RATE'
)

RETURN RATECODE
```

## Global Absence Proration

The Global Absence Proration fast formula can be used to apply a proration factor (or a multiplication factor), onto the final accrual calculated and returned by the accrual calculation rules in an accrual based absence plan.

If an organization has an accrual plan where the accrual rate varies based on Worker grades, and on top of that if a multiplication factor such as 0.75 needs to be applied depending on the Worker work location, then the band based on grades can be defined in the accrual matrix and the multiplication factor of 0.75 based on work location can be defined in the Global Absence Proration formula.

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

## Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Accrual**.
4. Click **Continue**.
5. On the Edit Absence Plan page Accruals tab, Accrual Attributes section, select **Formula** in the **Accrual Proration Rule** field.
6. Select the defined formula from the **Accrual Proration Formula** field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number

Context Name	Data Type
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_ACCRUAL	Number	Accrual value received from the accrual matrix
IV_CARRYOVER	Number	Carryover limit received from accrual matrix
IV_CEILING	Number	Ceiling limit received from the accrual matrix
IV_ACCRUALPERIODSTARTDATE	Date	Start date of accrual period
IV_ACCRUALPERIODENDDATE	Date	End date of accrual period
IV_CALENDARSTARTDATE	Date	Start date of accrual calendar year
IV_CALEDARENDDATE	Date	End date of accrual calendar year
IV_PLANENROLLMENTSTARTDATE	Date	Start date of enrollment
IV_PLANENROLLMENTENDDATE	Date	End date of enrollment
IV_ACCRUAL_CEILING	Number	Annual accrual limit from the accrual matrix

## Return Variables

Variable Name	Data Type	Description
prorationFactor	Number	Multiplication factor used to pro-rate final accrual amount

## Sample Formula

**Requirement:** The organization has an accrual matrix based on grade of Workers. On top of the values defined by the matrix, Workers who belong to jobs 'Wealth Management' or 'Trade Associate' should have their annual accrual amount pro-rated by a factor of 0.75 as they have flexible working arrangements and do not fall under the standard absence accrual benefit policy of the organization.

**Solution:** Once the accrual rules are defined in the accrual matrix, a Global Absence Proration formula such as the one below can be used.

FORMULA NAME: ANC\_PRORAT

FORMULA TYPE: Global Absence Proration

**DESCRIPTION:** This formula returns the proration factor accrual absence plan

```
DEFAULT FOR PER_ASG_JOB_NAME IS ''  
  
prorationFactor = 1  
  
IF (PER_ASG_JOB_NAME = 'Wealth Management Consultant' OR PER_ASG_JOB_NAME = 'Trade Associate')  
THEN  
  (prorationFactor = 0.75)  
  
RETURN prorationFactor
```

## Global Absence Rollover

The Global Absence Rollover fast formula can be used in cases where a single rollover limit rule does not apply to the entire population that belongs to the accrual plan.

An organization might have a rollover rule that generally allows a maximum of 30 days to be rolled over to another plan at the end of plan year. However, the workers in a particular department are allowed to rollover an additional 10 day. In such cases, this logic can be composed into the fast formula so that when rollover limit is determined, the application dynamically allocates different limits to different workers depending on their department.

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

## Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Accrual**.
4. Click **Continue**.
5. On the Edit Absence Plan page Accruals tab Year End Processing section, select **Limited by formula** in the **Rollover Rule** field.
6. Select the defined formula from the **Limit Formula** field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date

Context Name	Data Type
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_ACCRUAL	Number	Accrual value received from the accrual matrix
IV_CARRYOVER	Number	Carryover limit received from accrual matrix
IV_CEILING	Number	Ceiling limit received from the accrual matrix
IV_ACCRUAL_CEILING	Number	Annual accrual limit from the accrual matrix
IV_ACCRUALPERIODSTARTDATE	Date	Start date of accrual period
IV_ACCRUALPERIODENDDATE	Date	End date of accrual period
IV_CALENDARSTARTDATE	Date	Start date of accrual calendar year

Variable Name	Data Type	Description
IV_CALEDARENDDATE	Date	End date of accrual calendar year
IV_PLANENROLLMENTSTARTDATE	Date	Start date of enrollment
IV_PLANENROLLMENTENDDATE	Date	End date of enrollment

## Return Variables

Variable Name	Data Type	Description
ROLLOVER	Number	Maximum amount that a worker can rollover at the end of year

## Sample Formula

**Requirement:** The organization has a rollover limit rule, which allows workers to rollover up to a maximum of 30 days to another plan at the end of year. However, workers belonging to business unit 'OBU1' are allowed to rollover up to a maximum of 40 days at the end of year.

**Solution:** A Global Absence Rollover formula such as the one below can be used.

FORMULA NAME: ANC\_ROLLOVER

FORMULA TYPE: Global Absence Rollover

DESCRIPTION: This formula calculates and returns the rollover limit amount for the vacation plan

```
DEFAULT FOR PER_ASG_BUSINESS_UNIT_NAME IS ''
ROLLOVER=30
IF (PER_ASG_BUSINESS_UNIT_NAME = 'OBU1')
THEN
(ROLLOVER = 40)

RETURN ROLLOVER
```

## Global Absence Rollover Proration

The Global Absence Rollover Proration fast formula can be used in cases where a proration factor (or a multiplication factor) needs to be applied onto the rollover limit.

An organization might have a rule that requires the rollover limit to be prorated based on FTE of a worker. In such a case, after the rollover rule is defined, the rollover proration rule can be composed to return a proration factor, which will be multiplied onto the rollover limit before returning the final value against the worker's enrollment data.

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

## Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Accrual**.
4. Click **Continue**.
5. On the Edit Absence Plan page Accruals tab Year End Processing section, select a value other than **Not Applicable** in the **Rollover Rule** field.
6. After defining the ceiling rule as desired, select **Formula** in the **Limit Proration Rule** field.
7. Select the defined formula from the **Limit Proration Formula** field

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number

Context Name	Data Type
PERSON_ID	Number
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_ACCRUAL	Number	Accrual value received from the accrual matrix
IV_CARRYOVER	Number	Carryover limit received from accrual matrix
IV_CEILING	Number	Ceiling limit received from the accrual matrix
IV_ACCRUAL_CEILING	Number	Annual accrual limit from the accrual matrix
IV_ACCRUALPERIODSTARTDATE	Date	Start date of accrual period
IV_ACCRUALPERIODENDDATE	Date	End date of accrual period
IV_CALENDARSTARTDATE	Date	Start date of accrual calendar year
IV_CALEDARENDDATE	Date	End date of accrual calendar year
IV_PLANENROLLMENTSTARTDATE	Date	Start date of enrollment
IV_PLANENROLLMENTENDDATE	Date	End date of enrollment

## Return Variables

Variable Name	Data Type	Description
ROLLOVERPRORATIONFACTOR	Number	Multiplication factor for Rollover Limit

## Sample Formula

**Requirement:** The organization has a rollover limit rule at the end of year, which allows workers to rollover to a maximum of 30 days to another plan. This rollover limit needs to be pro-rated by FTE for workers who have the jobs 'Wealth Management' or 'Trade Associate' assigned against their assignment record.

**Solution:** A Global Absence Rollover formula such as the one below can be used.

FORMULA NAME: ANC\_ROLLOVER\_PRORATION

FORMULA TYPE: Global Absence Rollover Proration

DESCRIPTION: This formula calculates and returns the rollover amount proration for the vacation plan

```
DEFAULT FOR PER_ASG_FTE_VALUE IS 1
DEFAULT FOR PER_ASG_JOB_NAME IS ' '

ROLLOVERPRORATIONFACTOR = 1
IF (PER_ASG_JOB_NAME = 'Wealth Management Consultant' OR PER_ASG_JOB_NAME = 'Trade Associate')
THEN
(ROLLOVERPRORATIONFACTOR = ROUND(PER_ASG_FTE_VALUE,2))

RETURN ROLLOVERPRORATIONFACTOR
```

## Global Absence Target Plan

For automated enrollment transfer feature, same category needs to be set across transfer source plan and target plan. But there is a chance that there are multiple plans using the same category, and employees are eligible for more than one plan. In this case, Global Absence Target Plan fast formula needs to be defined to identify the exact target plan name.

There are 3 plans – Plan A, Plan B and Plan C. These 3 plans belong to same plan category (Category One). Now employee is about to lose eligibility of Plan A, meanwhile employee is enrolled into Plan B and Plan C. If transfer rule is defined, system will not transfer any balance without this particular Absence Target Plan fast formula, because it doesn't know which one to transfer to (Plan B or Plan C).

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

## Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Accrual**.
4. Click **Continue**.
5. On the Edit Absence Plan page Participation tab Transfer Rules section, check the box **Transfer positive balance**, then select a value in the **Limit Rule** field. This is a required field.
6. Select the defined formula from the **Target Plan Formula** field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date

Context Name	Data Type
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_ACCRUAL	Number	Accrual value received from the accrual matrix
IV_CARRYOVER	Number	Carryover limit received from accrual matrix
IV_CEILING	Number	Ceiling limit received from the accrual matrix
IV_ACCRUAL_CEILING	Number	Annual accrual limit from the accrual matrix
IV_ACCRUALPERIODSTARTDATE	Date	Start date of accrual period
IV_ACCRUALPERIODENDDATE	Date	End date of accrual period
IV_CALENDARSTARTDATE	Date	Start date of accrual calendar year
IV_CALEDARENDDATE	Date	End date of accrual calendar year
IV_PLANENROLLMENTSTARTDATE	Date	Start date of enrollment
IV_PLANENROLLMENTENDDATE	Date	End date of enrollment

## Return Variables

Variable Name	Data Type	Description
TARGET_PLAN_NAME	Text	Target plan name which a worker can transfer to when loss of plan eligibility happens

## Sample Formula

**Requirement:** The organization has an enrollment transfer rule, and there are multiple plans using same plan category. If employee is about to lose eligibility of Plan\_A, meanwhile they will be enrolled for Plan\_B and Plan\_C that shares same plan category. Now we want to choose different target plan based on different business unit.

**Solution:** A Global Absence Target Plan formula such as the one below can be used.

FORMULA NAME: ANC\_ENRT\_TARGET\_FF

FORMULA TYPE: Global Absence Rollover

DESCRIPTION: This formula calculates and returns the rollover limit amount for the vacation plan

```
DEFAULT FOR PER_ASG_BUSINESS_UNIT_NAME IS ''  
TARGET_PLAN_NAME='Plan_B'  
IF (PER_ASG_BUSINESS_UNIT_NAME = 'OBU1')  
THEN  
TARGET_PLAN_NAME='Plan_C'  
RETURN TARGET_PLAN_NAME
```

## Global Absence Transfer

The Global Absence Transfer fast formula can be used in cases where a single automated enrollment transfer rule does not apply to the entire population that belongs to the accrual plan.

An organization might have an enrollment transfer rule that generally allows a maximum of 30 days to be transferred to another plan when current plan loses eligibility. However, the workers in a particular department are allowed to transfer an additional 10 days due to the nature of their work. In such cases, this logic can be composed into the fast formula so that when enrollment transfer limit is determined, the application dynamically allocates different limits to different workers depending on their department.

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

## Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Accrual**.

4. Click **Continue**.
5. On the Edit Absence Plan page Participation tab Transfer Rules section, check the box **Transfer positive balance**, then select **Limited byFormula** in the **Limit Rule** field.
6. Select the defined formula from the **Limit Formula** field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_ACCRUAL	Number	Accrual value received from the accrual matrix
IV_CARRYOVER	Number	Carryover limit received from accrual matrix
IV_CEILING	Number	Ceiling limit received from the accrual matrix
IV_ACCRUAL_CEILING	Number	Annual accrual limit from the accrual matrix
IV_ACCRUALPERIODSTARTDATE	Date	Start date of accrual period
IV_ACCRUALPERIODENDDATE	Date	End date of accrual period
IV_CALENDARSTARTDATE	Date	Start date of accrual calendar year
IV_CALEDARENDDATE	Date	End date of accrual calendar year
IV_PLANENROLLMENTSTARTDATE	Date	Start date of enrollment
IV_PLANENROLLMENTENDDATE	Date	End date of enrollment

## Return Variables

Variable Name	Data Type	Description
TRANSFER	Number	Maximum amount that a worker can transfer when loss of plan eligibility happens

## Sample Formula

**Requirement:** The organization has an enrollment transfer limit rule, which allows workers to transfer up to a maximum of 30 days when loss of plan eligibility happens. But workers belonging to business unit 'OBU1' are allowed to transfer up to a maximum of 40 days when loss of plan eligibility happens.

**Solution:** A Global Absence Transfer formula such as the one below can be used.

FORMULA NAME: ANC\_ENRT\_TRANSFER

FORMULA TYPE: Global Absence Transfer

DESCRIPTION: This formula calculates and returns the rollover limit amount for the vacation plan

```
DEFAULT FOR PER_ASG_BUSINESS_UNIT_NAME IS ''
TRANSFER=30
IF (PER_ASG_BUSINESS_UNIT_NAME = 'OBU1')
THEN
(TRANSFER = 40)
```

RETURN TRANSFER

## Global Absence Transfer Proration

The Global Absence Transfer Proration fast formula can be used in cases where a proration factor (or a multiplication factor) needs to be applied onto the maximum enrollment transfer limit.

An organization might have a rule which asks for the enrollment transfer limit to be prorated based on FTE or even their job. In such a case, after the enrollment transfer rule is defined, the transfer proration rule can be composed to return a proration factor which will be multiplied onto the enrollment transfer limit before returning the final value against the worker's enrollment data.

### Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

### Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Accrual**.
4. Click **Continue**.
5. On the Edit Absence Plan page Participation tab Transfer Rules section, check the box **Transfer positive balance**, then select a value other than **Unlimited** in the **Limit Rule** field.
6. Select the defined formula from the **Limit Proration Formula** field.

### Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number

Context Name	Data Type
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_ACCRUAL	Number	Accrual value received from the accrual matrix
IV_CARRYOVER	Number	Carryover limit received from accrual matrix
IV_CEILING	Number	Ceiling limit received from the accrual matrix
IV_ACCRUAL_CEILING	Number	Annual accrual limit from the accrual matrix
IV_ACCRUALPERIODSTARTDATE	Date	Start date of accrual period
IV_ACCRUALPERIODENDDATE	Date	End date of accrual period
IV_CALENDARSTARTDATE	Date	Start date of accrual calendar year
IV_CALEDARENDDATE	Date	End date of accrual calendar year
IV_PLANENROLLMENTSTARTDATE	Date	Start date of enrollment
IV_PLANENROLLMENTENDDATE	Date	End date of enrollment

## Return Variables

Variable Name	Data Type	Description
TRANSFERPRORATIONFACTOR	Number	Multiplication factor for Transfer Limit

## Sample Formula

**Requirement:** The organization has an enrollment transfer limit proration rule when loss of plan eligibility happens, which allows workers to transfer to a maximum of 30 days. This transfer limit needs to be pro-rated by FTE for workers who have the jobs 'Wealth Management' or 'Trade Associate' assigned against their assignment record.

**Solution:** A Global Absence Transfer Proration formula such as the one below can be used.

FORMULA NAME: ANC\_ENRT\_TRANSFERPRO

FORMULA TYPE: Global Absence Transfer Proration

DESCRIPTION: This formula returns the proration factor for the rollover limit for the absence plan

```
DEFAULT FOR PER_ASG_FTE_VALUE IS 1
DEFAULT FOR PER_ASG_JOB_NAME IS ''

TRANSFERPRORATIONFACTOR = 1

IF (PER_ASG_JOB_NAME = 'Wealth Management Consultant' OR PER_ASG_JOB_NAME = 'Trade Associate')
THEN
(TRANSFERPRORATIONFACTOR = ROUND(PER_ASG_FTE_VALUE,2))

RETURN TRANSFERPRORATIONFACTOR
```

## Global Absence Vesting Period

The Global Absence Vesting Period fast formula can be used to enforce a custom vesting period (a period during which the Worker is enrolled into the plan and accrues balance but cannot use them) logic while defining an absence plan.

An organization might have a vesting period rule for new workers who are hired as Interns or Graduates that they need to complete 30 days of employment before they can use their vacation balance. Here the vesting period formula can be composed to look at the person type to determine the period applicable for the particular enrollment.

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

## Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Accrual**.
4. Click **Continue**.
5. On the Edit Absence Plan page Accruals tab, Accrual Attributes section, select **Formula** in the **Accrual Vesting Rule** field.
6. Select the defined formula from the **Vesting Period Formula** field

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_ACCRUAL	Number	Accrual value received from the accrual matrix
IV_CARRYOVER	Number	Carryover limit received from accrual matrix
IV_CEILING	Number	Ceiling limit received from the accrual matrix

Variable Name	Data Type	Description
IV_ACCRUAL_CEILING	Number	Annual accrual limit from the accrual matrix
IV_ACCRUALPERIODSTARTDATE	Date	Start date of accrual period
IV_ACCRUALPERIODENDDATE	Date	End date of accrual period
IV_CALENDARSTARTDATE	Date	Start date of accrual calendar year
IV_CALEDARENDDATE	Date	End date of accrual calendar year
IV_PLANENROLLMENTSTARTDATE	Date	Start date of enrollment
IV_PLANENROLLMENTENDDATE	Date	End date of enrollment

## Return Variables

Variable Name	Data Type	Description
vestingUOM	Text	Unit of measure for the vesting period duration. UoM can be Calendar Days ('C'), Weeks ('W'), Months ('M') or Years ('Y')
vestingUnits	Number	Period for which time-off is accrued, but cannot be used by worker.

## Sample Formula

**Requirement:** The organization has a vesting rule which allows newly joined interns and graduates to avail their vacation balance only after 30 days from enrollment into the plan.

**Solution:** A Global Absence Vesting Period formula such as the one below can be used.

FORMULA NAME: ANC\_VESTPRD

FORMULA TYPE: Global Absence Vesting Period

DESCRIPTION: This formula calculates and returns the vesting period duration for the vacation plan

```

DEFAULT FOR PER_ASG_USER_PERSON_TYPE IS 'Worker'

vestingUOM = 'C'
vestingUnits = 0

IF (PER_ASG_USER_PERSON_TYPE != 'Worker')
THEN
(
vestingUOM = 'C'
vestingUnits=30
)

RETURN vestingUOM,vestingUnits

```

# Formulas for Qualification Plans

Use the Absence Plan page to incorporate qualification plan rules. However, if you want to define other special rules to suit your requirement, you can write your own formulas.

## Formulas for Qualification Plan Rules

The following table lists the aspects of a qualification plan for which you can write a formula and identifies the formula type for each.

Rule	Description	Formula Type to Use
Start Rule	When the rolling backward plan term starts. A rolling backward term is a specific time period that precedes the absence start date.	Global Absence Plan Roll Backward End
Qualification Date Rule	Date when eligible workers enroll in the plan.	Global Absence Plan Enrollment Start Date
Conversion Formula	<p>Method to calculate the absence duration differently.</p> <p>Example: You want to consider only whole working days in a sickness absence in the entitlement calculation. In such cases, you define logic in a formula to convert the absence duration to a value that excludes partial working days.</p>	Global Absence Plan Duration
Entitlement Definition Type	Determines payment percentages to apply during the absence period.	Global Absence Entitlement
Entitlement Formula (Qualification Details section)	A level that determines the payment that workers receive for a specific number of days during a long leave of absence based on their length of service.	Global Absence Band Entitlement
Absence Payment Rate Rule	Method to calculate payment during absence period.	Global Absence Plan Use Rate

## Global Absence Plan Duration

The Global Absence Plan Duration fast formula can be used to override the default duration calculation logic for daily entitlement duration entries against qualification plans.

If the entitlement to be considered against a qualification plan in an organization depends on the location of the Worker, then this formula can be leveraged to specify this dynamic calculation logic. This formula is invoked once for each day of absence. If there are multiple shifts in a day, then the formula is invoked once for each shift. If the shift spans over a day-divide, then the shift is split into two and the formula is invoked once for each shift block.

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition. This is currently available only for plans whose UoM is Days or Hours.

## Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Qualification**.
4. Click **Continue**.
5. On the Edit Absence Plan page Plan Attributes tab General Attributes section, select the defined formula from the **Conversion Formula** field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number

Context Name	Data Type
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_ABS_START_DATE	Date	Start date-time of absence entry
IV_ABS_END_DATE	Date	End date-time of absence entry
IV_ABS_START_DURATION	Number	Start date duration of absence entry. Applicable when absence entry is being recorded against an Elapsed based work schedule
IV_ABS_END_DURATION	Number	End date duration of absence entry. Applicable when absence entry is being recorded against an Elapsed based work schedule
IV_START_DATE	Date	Start date of shift block within absence
IV_END_DATE	Date	End date of shift block within absence
IV_START_DURATION	Number	Start date duration of the shift block within absence entry. Applicable when absence entry is being recorded against an Elapsed based work schedule
IV_END_DURATION	Number	End date duration of the shift block within absence entry. Applicable when absence entry is being recorded against an Elapsed based work schedule
IV_START_TIME	Text	Start time on the start date of the shift block within absence entry. Applicable when absence entry is being recorded against a Time based work schedule. For elapsed work schedules it will be 00:00. If there are no shifts on the day being processed, then absence start time or 00:00 is passed, whichever is greater
IV_END_TIME	Text	End time on the end date of the shift block within absence entry. Applicable when absence entry is being recorded against a Time based

Variable Name	Data Type	Description
		work schedule. For elapsed work schedules it will be 23:59. If there are no shifts on the day being processed, then absence end time or 23:59 is passed, whichever is lesser
IV_UOM	Text	Unit of measure
IV_ABSENCE_TYPE_ID	Number	Absence type ID for the absence being recorded.
IV_USER_MODE	Text	Identifies the role of the person recording the absence.

## Return Variables

Variable Name	Data Type	Description
DURATION	Number	Duration of plan entries

## Sample Formula

**Requirement:** The organization has a qualification plan where the entitlement usage is only 50% for workers who work in an office whose location is in California, whereas for the rest of the workers, the entitlement usage is same as the absence duration.

**Solution:** A Global Absence Plan Duration formula such as the one below can be used.

FORMULA NAME: ANC\_PLN\_DUR

FORMULA TYPE: Global Absence Plan Duration

DESCRIPTION: This formula evaluates the entitlement usage based on worker location

```

DEFAULT FOR IV_START_DATE IS '4712/12/31 00:00:00' (date)
DEFAULT FOR IV_END_DATE IS '4712/12/31 00:00:00' (date)
DEFAULT FOR IV_START_TIME IS '00:00'
DEFAULT FOR IV_END_TIME IS '23:59'
DEFAULT FOR PER_ASG_LOC_REGION2 IS 'XXXXXXXXXX'
INPUTS ARE IV_START_DATE (date), IV_END_DATE (date), IV_START_TIME(text), IV_END_TIME(text)

ln_entry_duration_d = 0
ln_unrounded_duration_d = 0

ld_period_start_date = to_date(to_char(IV_START_DATE, 'DD/MM/RRRR') || ' ' || IV_START_TIME, 'DD/MM/RRRR
HH24:MI:SS')
ld_period_end_date = to_date(to_char(IV_END_DATE, 'DD/MM/RRRR') || ' ' || IV_END_TIME, 'DD/MM/RRRR HH24:MI:SS')

ln_entry_duration_d = GET_PAY_AVAILABILITY ('ASSIGN',
ld_period_start_date,
ld_period_end_date,
'Y',
'Y',

```

```
'Y',
'Y',
'D')

IF (PER_ASG_LOC_REGION2 = 'CA')
THEN
(
  ln_entry_duration_d = ln_entry_duration_d * 0.5
)
DURATION = ROUND(ln_entry_duration_d,2)
RETURN DURATION
```

## Global Absence Band Entitlement

The Global Absence Band Entitlement fast formula can be used to define the bands of entitlement duration and percentage of payment that is applicable against a qualification plan entitlement.

An organization might have a rule that gives workers in a certain location additional fully paid days of maternity entitlement when compared to workers working in any other location.

### Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

### Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Qualification**.
4. Click **Continue**.
5. On the Edit Absence Plan page Entitlements tab, Qualification Band Matrix table, create a matrix line by clicking on the **Add** button.
6. With the matrix line selected, in the **Qualification Details** table, click on the **Add** button.
7. Choose the defined formula in the **Entitlement Formula** field.

### Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date

Context Name	Data Type
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_START_DATE	Date	Start date of absence entry
IV_END_DATE	Date	End date of absence entry
IV_TOTALDURATION	Number	Duration of absence entry
IV_START_DURATION	Number	Start date duration of the absence entry. Applicable when absence entry is being recorded against an Elapsed based work schedule
IV_END_DURATION	Number	End date duration of the absence entry. Applicable when absence entry is being recorded against an Elapsed based work schedule
IV_START_TIME	Text	Start time on the start date of the absence entry. Applicable when absence entry is being recorded against a Time based work schedule. For elapsed work schedules it will be 00:00.

Variable Name	Data Type	Description
IV_END_TIME	Text	End time on the end date for the absence being entered. Applicable when absence entry is being recorded against a Time based work schedule. For elapsed work schedules it will be 23:59.
IV_ACTUALCHILDBIRTHDATE	Date	Actual date of childbirth entered for the absence. Applicable for Childbirth/placement absence pattern
IV_ACTUALSTARTDATE	Date	Actual start date of absence. Applicable for Childbirth/placement absence pattern
IV_ACTUALENDDATE	Date	Actual end date of absence. Applicable for Childbirth/placement absence pattern
IV_EXPECTEDCHILDBIRTHDATE	Date	Expected date of childbirth entered for the absence. Applicable for Childbirth/placement absence pattern
IV_PLANNEDSTARTDATE	Date	Expected start date of absence. Applicable for Childbirth/placement absence pattern
IV_PLANNEDENDDATE	Date	Expected end date of absence. Applicable for Childbirth/placement absence pattern
IV_ABSENCE_REASON	Text	Absence reason in the language of the current session
IV_ATTRIBUTE_CATEGORY	Text	Context of the DFF row for Absence Recordings DFF
IV_ATTRIBUTE_1*	Text	First text segment of the Absence Recordings DFF
IV_ATTRIBUTE_NUMBER1*	Number	First number segment of the Absence Recordings DFF
IV_ATTRIBUTE_DATE1*	Date	First date segment of the Absence Recordings DFF
IV_ATTRIBUTE_ARR	Text_Number	Array of text segments of the Absence Recordings DFF. The index indicates the segment number. Eg: IV_ATTRIBUTE_ARR[1] holds the value of segment ATTRIBUTE1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using

Variable Name	Data Type	Description
		the “exists” command. Eg: if(IV_ATTRIBUTE_ARR.exists(1))
IV_ATTRIBUTE_NUMBER_ARR	Number_Number	Array of number segments of the Absence Recordings DFF. The index indicates the segment number. Eg: IV_ATTRIBUTE_NUMBER_ARR[1] holds the value of segment ATTRIBUTE_NUMBER1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_ATTRIBUTE_NUMBER_ARR.exists(1))
IV_ATTRIBUTE_DATE_ARR	Date_Number	Array of date segments of the Absence Recordings DFF. The index indicates the segment number. Eg: IV_ATTRIBUTE_DATE_ARR[1] holds the value of segment ATTRIBUTE_DATE1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_ATTRIBUTE_DATE_ARR.exists(1))
IV_INFORMATION_CATEGORY	Text	Context of the DDF row for ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_1*	Text	First text segment of the ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_NUMBER1*	Number	First number segment of the ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_DATE1*	Date	First date segment of the ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_ARR	Text_Number	Array of text segments of the ANC_PER_ABS_ENTRIES DDF. The index indicates the segment number. Eg: IV_INFORMATION_ARR[1] holds the value of segment INFORMATION1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value.

Variable Name	Data Type	Description
		This can be done using the “exists” command. Eg: if(IV_INFORMATION_ARR.exists(1))
IV_INFORMATION_NUMBER_ARR	Number_Number	Array of number segments of the ANC_PER_ABS_ENTRIES DDF. The index indicates the segment number. Eg: IV_INFORMATION_NUMBER_ARR[1] holds the value of segment INFORMATION_NUMBER1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_INFORMATION_NUMBER_ARR.exists(1))
IV_INFORMATION_DATE_ARR	Date_Number	Array of date segments of the ANC_PER_ABS_ENTRIES DDF. The index indicates the segment number. Eg: IV_INFORMATION_DATE_ARR[1] holds the value of segment INFORMATION_DATE1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_INFORMATION_DATE_ARR.exists(1))
IV_PAYMENT_DTL_BAND	Number	Identifier of the payment band chosen for the absence linked to an agreement
IV_NOTIFICATION_DATE	Date	Date of illness/injury notification
IV_MATCHING_DATE	Date	Date on which child is matched with the worker for adoption

\*The sequence for this input value can be extended up to 5

## Return Variables

Variable Name	Data Type	Description
BANDPAYFACTOR	Number	Pay factor for entitlement band value
BANDENTITLEMENT	Number	Entitlement duration that worker is eligible for

Variable Name	Data Type	Description
DEBUG_MESSAGE	Text	Text passed into this output variable would get logged if application logging is enabled. Useful for debugging.

## Sample Formula

**Requirement:** The organization has an entitlement payment band that is based on the location. Workers belonging to a location such as California receive an additional 10 days of fully paid entitlement for Maternity Leave.

**Solution:** A Global Absence Band Entitlement formula such as the one below can be used.

FORMULA NAME: ANC\_BND\_ENTLMNT

FORMULA TYPE: Global Absence Band Entitlement

DESCRIPTION: This formula evaluates the entitlement band based on location of Worker

```
DEFAULT FOR PER_ASG_LOC_REGION2 IS 'XXXXXXXXXX'

IF (PER_ASG_LOC_REGION2 = 'CA')
THEN
(
BANDPAYFACTOR = 100
BANDENTITLEMENT = 130
)
ELSE
(
BANDPAYFACTOR = 100
BANDENTITLEMENT = 120
)

RETURN BANDPAYFACTOR, BANDENTITLEMENT
```

## Global Absence Plan Enrollment Start Date

The Global Absence Plan Enrollment Start Date fast formula can be used to specify the Qualification date for the absence plan.

An organization might have a rule for Maternity entitlements according to which the qualification date is on the absence start date if actual dates are entered. If it is not entered, then the qualification date needs to be the event date (actual if available, or else, the planned date). For including such conditional logic to determine the qualification date, formulas of this type can be used.

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

## Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.

2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Qualification**.
4. Click **Continue**.
5. On the Edit Absence Plan page Participations tab, Enrollment Rules section, choose Formula in the **Qualification Date Rule** field.
6. Select the defined formula in the **Formula** field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_START_DATE	Date	Start date of absence entry
IV_END_DATE	Date	End date of absence entry
IV_TOTALDURATION	Number	Duration of absence entry
IV_START_DURATION	Number	Start date duration of the absence entry. Applicable when absence entry is being recorded against an Elapsed based work schedule
IV_END_DURATION	Number	End date duration of the absence entry. Applicable when absence entry is being recorded against an Elapsed based work schedule
IV_START_TIME	Text	Start time on the start date of the absence entry. Applicable when absence entry is being recorded against a Time based work schedule. For elapsed work schedules it will be 00:00.
IV_END_TIME	Text	End time on the end date for the absence being entered. Applicable when absence entry is being recorded against a Time based work schedule. For elapsed work schedules it will be 23:59.
IV_ACTUALCHILDBIRTHDATE	Date	Actual date of childbirth entered for the absence. Applicable for Childbirth/placement absence pattern
IV_ACTUALSTARTDATE	Date	Actual start date of absence. Applicable for Childbirth/placement absence pattern
IV_ACTUALENDDATE	Date	Actual end date of absence. Applicable for Childbirth/placement absence pattern
IV_EXPECTEDCHILDBIRTHDATE	Date	Expected date of childbirth entered for the absence. Applicable for Childbirth/placement absence pattern
IV_PLANNEDSTARTDATE	Date	Expected start date of absence. Applicable for Childbirth/placement absence pattern
IV_PLANNEDENDDATE	Date	Expected end date of absence. Applicable for Childbirth/placement absence pattern

Variable Name	Data Type	Description
IV_ABSENCE_REASON	Text	Absence reason in the language of the current session

## Return Variables

Variable Name	Data Type	Description
ENROLLMENTSTARTDATE	Date	Qualification date used by the absence plan

## Sample Formula

**Requirement:** The organization has a rule for Maternity entitlements according to which the qualification date is on the absence start date if actual dates are entered. If this is not provided, then the qualification date needs to be the event date (actual if available, or else, the planned date).

**Solution:** A Global Absence Enrollment Start Date formula such as the one below can be used.

FORMULA NAME: ANC\_ENROLMNT\_START

FORMULA TYPE: Global Absence Plan Enrollment Start Date

DESCRIPTION: This formula returns the qualification date as of which participation to qualification plan needs to be terminated

```

DEFAULT FOR IV_ACTUALCHILDBIRTHDATE IS '4712/12/31 00:00:00' (date)
DEFAULT FOR IV_ACTUALSTARTDATE IS '4712/12/31 00:00:00' (date)
DEFAULT FOR IV_EXPECTEDCHILDBIRTHDATE IS '4712/12/31 00:00:00' (date)
DEFAULT FOR PER_ASG_REL_ACTUAL_TERMINATION_DATE IS '4712/12/31 00:00:00' (date)

INPUTS ARE IV_ACTUALCHILDBIRTHDATE (date), IV_ACTUALSTARTDATE (date), IV_EXPECTEDCHILDBIRTHDATE (date)

ld_effective_date = GET_CONTEXT(EFFECTIVE_DATE, '4712/12/31 12:00:00' (date))

IF (IV_ACTUALSTARTDATE WAS NOT DEFAULTED)
THEN
(
ENROLLMENTSTARTDATE = IV_ACTUALSTARTDATE
)
ELSE
(
IF (IV_ACTUALCHILDBIRTHDATE WAS NOT DEFAULTED)
THEN
(
ENROLLMENTSTARTDATE = IV_ACTUALCHILDBIRTHDATE
)
ELSE
(
ENROLLMENTSTARTDATE = IV_EXPECTEDCHILDBIRTHDATE
)
)
)

RETURN ENROLLMENTSTARTDATE

```

# Global Absence Plan Entitlement

The Global Absence Plan Entitlement fast formula can be used to define the entire entitlement structure for a qualification plan for cases where matrix architecture doesn't fit the bill.

## Configuration Point in HCM Cloud

If you've created this formula, you can attach this formula to the absence plan definition.

## Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Qualification**.
4. Click **Continue**.
5. On the Edit Absence Plan page Entitlements tab, Entitlement Attributes section, choose **Formula** in the **Entitlement Definition Type** choice button field.
6. Select the defined formula in the **Entitlement Formula** field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_CATEGORY_ID	Number
ABSENCE_ENTRY_ID	Number
ABSENCE_MATERNITY_ID	Number
ABSENCE_REASON_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number

Context Name	Data Type
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_START_DATE	Date	Start date of absence entry
IV_END_DATE	Date	End date of absence entry
IV_TOTALDURATION	Number	Duration of absence entry
IV_START_DURATION	Number	Start date duration of the absence entry. Applicable when absence entry is being recorded against an Elapsed based work schedule
IV_END_DURATION	Number	End date duration of the absence entry. Applicable when absence entry is being recorded against an Elapsed based work schedule
IV_START_TIME	Text	Start time on the start date of the absence entry. Applicable when absence entry is being recorded against a Time based work schedule. For elapsed work schedules it will be 00:00.
IV_END_TIME	Text	End time on the end date for the absence being entered. Applicable when absence entry is being recorded against a Time based work schedule. For elapsed work schedules it will be 23:59.
IV_ACTUALCHILDBIRTHDATE	Date	Actual date of childbirth entered for the absence. Applicable for Childbirth/placement absence pattern

Variable Name	Data Type	Description
IV_ACTUALSTARTDATE	Date	Actual start date of absence. Applicable for Childbirth/placement absence pattern
IV_ACTUALENDDATE	Date	Actual end date of absence. Applicable for Childbirth/placement absence pattern
IV_EXPECTEDCHILDBIRTHDATE	Date	Expected date of childbirth entered for the absence. Applicable for Childbirth/placement absence pattern
IV_PLANNEDSTARTDATE	Date	Expected start date of absence. Applicable for Childbirth/placement absence pattern
IV_PLANNEDENDDATE	Date	Expected end date of absence. Applicable for Childbirth/placement absence pattern
IV_ABSENCE_REASON	Text	Absence reason in the language of the current session
IV_ATTRIBUTE_CATEGORY	Text	Context of the DFF row for Absence Recordings DFF
IV_ATTRIBUTE_1*	Text	First text segment of the Absence Recordings DFF
IV_ATTRIBUTE_NUMBER1*	Number	First number segment of the Absence Recordings DFF
IV_ATTRIBUTE_DATE1*	Date	First date segment of the Absence Recordings DFF
IV_ATTRIBUTE_ARR	Text_Number	Array of text segments of the Absence Recordings DFF. The index indicates the segment number. Eg: IV_ATTRIBUTE_ARR[1] holds the value of segment ATTRIBUTE1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the "exists" command. Eg: if(IV_ATTRIBUTE_ARR.exists(1))
IV_ATTRIBUTE_NUMBER_ARR	Number_Number	Array of number segments of the Absence Recordings DFF. The index indicates the segment number. Eg: IV_ATTRIBUTE_NUMBER_ARR[1] holds the value of segment ATTRIBUTE_NUMBER1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does

Variable Name	Data Type	Description
		not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_ATTRIBUTE_NUMBER_ARR.exists(1))
IV_ATTRIBUTE_DATE_ARR	Date_Number	Array of date segments of the Absence Recordings DFF. The index indicates the segment number. Eg: IV_ATTRIBUTE_DATE_ARR[1] holds the value of segment ATTRIBUTE_DATE1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_ATTRIBUTE_DATE_ARR.exists(1))
IV_INFORMATION_CATEGORY	Text	Context of the DDF row for ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_1*	Text	First text segment of the ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_NUMBER1*	Number	First number segment of the ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_DATE1*	Date	First date segment of the ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_ARR	Text_Number	Array of text segments of the ANC_PER_ABS_ENTRIES DDF. The index indicates the segment number. Eg: IV_INFORMATION_ARR[1] holds the value of segment INFORMATION1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_INFORMATION_ARR.exists(1))
IV_INFORMATION_NUMBER_ARR	Number_Number	Array of number segments of the ANC_PER_ABS_ENTRIES DDF. The index indicates the segment number. Eg: IV_INFORMATION_NUMBER_ARR[1] holds the value of segment INFORMATION_NUMBER1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error.

Variable Name	Data Type	Description
		Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_INFORMATION_NUMBER_ARR.exists(1))
IV_INFORMATION_DATE_ARR	Date_Number	Array of date segments of the ANC_PER_ABS_ENTRIES DDF. The index indicates the segment number. Eg: IV_INFORMATION_DATE_ARR[1] holds the value of segment INFORMATION_DATE1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_INFORMATION_DATE_ARR.exists(1))
IV_PAYMENT_DTL_BAND	Number	Identifier of the payment band chosen for the absence linked to an agreement
IV_NOTIFICATION_DATE	Date	Date of illness/injury notification
IV_MATCHING_DATE	Date	Date on which child is matched with the worker for adoption
IV_LNKG_ABS_ID	Number_Number	Array of absence entry identifier for which linkage is created
IV_LNKG_LNKD_ABS_ID	Number_Number	Array of absence entry identifier to which absence is linked
IV_LNKG_LNKD_ABS_START	Date_Number	Array of start date time of absences which are linked
IV_LNKG_LNKD_ABS_END	Date_Number	Array of end date time of absences which are linked
IV_LNKG_REASON	Text_Number	Array of linkage reasons
IV_LNKG_REASON_ID	Number_Number	Array of linkage reason identifiers
IV_LNKG_CHAIN_ID	Number_Number	Array of linkage chain identifiers
IV_UI_PER_CERT_ID	Text	Identifier of the certificate record added to the absence. Applicable for certificates acted up-on by admin.
IV_UI_ABS_CERT_ID	Text	Absence certificate identifier. Applicable for certificates acted up-on by admin.

Variable Name	Data Type	Description
IV_UI_CERT_TYPE	Text	Certification type. Applicable for certificates acted up-on by admin.
IV_UI_CERT_REVPAYSTART_DATE	Date	Revised pay start date for the certificate. Applicable for certificates acted up-on by admin.
IV_UI_CERT_REVPAYEND_DATE	Date	Revised pay end date for the certificate. Applicable for certificates acted up-on by admin.
IV_UI_CERT_REVPAY_FACTOR	Number	Revised payment percentage for the certificate. Applicable for certificates acted up-on by admin.
IV_UI_CERT_CREATION_TYPE	Text	Certification creation type. Applicable for certificates acted up-on by admin.
IV_UI_CERT_STATUS	text	Certification status. Applicable for certificates acted up-on by admin.

\*The sequence for this input value can be extended up to 5

## Return Variables

Variable Name	Data Type	Description
BAND1PAYFACTOR	Number	Pay factor for the entitlement BAND1ENTITLEMENT
BAND1ENTITLEMENT	Number	Entitlement duration that the worker is eligible for
BAND1RATEDEFINITION	Number	Identifier of the rate definition to be used for BAND1ENTITLEMENT
BAND1USEDENTITLEMENT	Number	Used entitlement value to be used for BAND1ENTITLEMENT
BAND2PAYFACTOR	Number	Pay factor for the entitlement BAND2ENTITLEMENT
BAND2ENTITLEMENT	Number	Entitlement duration that the worker is eligible for
BAND2RATEDEFINITION	Number	Identifier of the rate definition to be used for BAND2ENTITLEMENT

Variable Name	Data Type	Description
BAND2USEDENTITLEMENT	Number	Used entitlement value to be used for BAND2ENTITLEMENT
BAND3PAYFACTOR	Number	Pay factor for the entitlement BAND3ENTITLEMENT
BAND3ENTITLEMENT	Number	Entitlement duration that the worker is eligible for
BAND3RATEDEFINITION	Number	Identifier of the rate definition to be used for BAND3ENTITLEMENT
BAND3USEDENTITLEMENT	Number	Used entitlement value to be used for BAND3ENTITLEMENT
BAND4PAYFACTOR	Number	Pay factor for the entitlement BAND4ENTITLEMENT
BAND4ENTITLEMENT	Number	Entitlement duration that the worker is eligible for
BAND4RATEDEFINITION	Number	Identifier of the rate definition to be used for BAND4ENTITLEMENT
BAND4USEDENTITLEMENT	Number	Used entitlement value to be used for BAND4ENTITLEMENT
BAND5PAYFACTOR	Number	Pay factor for the entitlement BAND5ENTITLEMENT
BAND5ENTITLEMENT	Number	Entitlement duration that the worker is eligible for
BAND5RATEDEFINITION	Number	Identifier of the rate definition to be used for BAND5ENTITLEMENT
BAND5USEDENTITLEMENT	Number	Used entitlement value to be used for BAND5ENTITLEMENT
CERT_NAMES	Text_Number	Array of certificates (managed by entitlement formula certificates) to be associated with the absence entry
CERT_START_DATES	Date_Number	Array of certification start dates of certificates (managed by entitlement formula certificates) associated with the absence entry
CERT_END_DATES	Date_Number	Array of certification end dates of certificates (managed by entitlement formula certificates) associated with the absence entry

Variable Name	Data Type	Description
CERT_COMMENTS	Text_Number	Array of certification comments associated with the absence entry
DEBUG_MESSAGE	Text	Text passed into this output variable would get logged if application logging is enabled. Useful for debugging.
BAND1NAME	Text	Band name for the entitlement BAND1ENTITLEMENT
BAND2NAME	Text	Band name for the entitlement BAND2ENTITLEMENT
BAND3NAME	Text	Band name for the entitlement BAND3ENTITLEMENT
BAND4NAME	Text	Band name for the entitlement BAND4ENTITLEMENT

## Sample Formula

**Requirement:** The organization has an entitlement payment band structures that is based on the working location of the worker. Workers belonging to a location that is in California follow a different band structure as compared to workers in the rest of the country.

**Solution:** A Global Absence Plan Entitlement formula such as the one below can be used.

FORMULA NAME: ANC\_PLN\_ENTLMNT

FORMULA TYPE: Global Absence Plan Entitlement

DESCRIPTION: This formula evaluates the entitlement band based on location of Worker

```

DEFAULT FOR PER_ASG_LOC_REGION2 IS 'XXXXXXXXXX'

IF (PER_ASG_LOC_REGION2 = 'CA')
THEN {
  BAND1PAYFACTOR = 100
  BAND1ENTITLEMENT = 50
  BAND2PAYFACTOR = 75
  BAND2ENTITLEMENT = 50
  BAND3PAYFACTOR = 50
  BAND3ENTITLEMENT = 50
}
ELSE {
  BAND1PAYFACTOR = 100
  BAND1ENTITLEMENT = 40
  BAND2PAYFACTOR = 75
  BAND2ENTITLEMENT = 40
  BAND3PAYFACTOR = 50
  BAND3ENTITLEMENT = 40
}
RETURN BAND1PAYFACTOR, BAND1ENTITLEMENT, BAND2PAYFACTOR, BAND2ENTITLEMENT, BAND3PAYFACTOR, BAND3ENTITLEMENT

```

# Global Absence Plan Roll Backward End

The Global Absence Plan Roll Backward End fast formula is to be used to determine the start date of a plan term that uses the Roll Backward term rule.

If the start date for plan term in a roll backward period needs to be 365 days prior to the absence end date, required logic can be composed into this formula and the reference date returned.

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

## Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Qualification**.
4. Click **Continue**.
5. On the Edit Absence Plan page Plan Attributes tab, Plan Term section, choose **Rolling backward** in the **Type** field.
6. Select **Formula** in the **Start Rule** field.
7. Choose the defined formula in the **Start Formula** field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_CATEGORY_ID	Number
ABSENCE_ENTRY_ID	Number
ABSENCE_MATERNITY_ID	Number
ABSENCE_REASON_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number

Context Name	Data Type
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_START_DATE	Date	Start date of absence entry
IV_END_DATE	Date	End date of absence entry
IV_TOTALDURATION	Number	Duration of absence entry
IV_START_DURATION	Number	Start date duration of the absence entry. Applicable when absence entry is being recorded against an Elapsed based work schedule
IV_END_DURATION	Number	End date duration of the absence entry. Applicable when absence entry is being recorded against an Elapsed based work schedule
IV_START_TIME	Text	Start time on the start date of the absence entry. Applicable when absence entry is being recorded against a Time based work schedule. For elapsed work schedules it will be 00:00.
IV_END_TIME	Text	End time on the end date for the absence being entered. Applicable when absence entry is being recorded against a Time based work

Variable Name	Data Type	Description
		schedule. For elapsed work schedules it will be 23:59.
IV_ACTUALCHILDBIRTHDATE	Date	Actual date of childbirth entered for the absence. Applicable for Childbirth/placement absence pattern
IV_ACTUALSTARTDATE	Date	Actual start date of absence. Applicable for Childbirth/placement absence pattern
IV_ACTUALENDDATE	Date	Actual end date of absence. Applicable for Childbirth/placement absence pattern
IV_EXPECTEDCHILDBIRTHDATE	Date	Expected date of childbirth entered for the absence. Applicable for Childbirth/placement absence pattern
IV_PLANNEDSTARTDATE	Date	Expected start date of absence. Applicable for Childbirth/placement absence pattern
IV_PLANNEDENDDATE	Date	Expected end date of absence. Applicable for Childbirth/placement absence pattern
IV_ABSENCE_REASON	Text	Absence reason in the language of the current session
IV_ATTRIBUTE_CATEGORY	Text	Context of the DFF row for Absence Recordings DFF
IV_ATTRIBUTE_1*	Text	First text segment of the Absence Recordings DFF
IV_ATTRIBUTE_NUMBER1*	Number	First number segment of the Absence Recordings DFF
IV_ATTRIBUTE_DATE1*	Date	First date segment of the Absence Recordings DFF
IV_ATTRIBUTE_ARR	Text_Number	Array of text segments of the Absence Recordings DFF. The index indicates the segment number. Eg: IV_ATTRIBUTE_ARR[1] holds the value of segment ATTRIBUTE1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the "exists" command. Eg: if(IV_ATTRIBUTE_ARR.exists(1))

Variable Name	Data Type	Description
IV_ATTRIBUTE_NUMBER_ARR	Number_Number	Array of number segments of the Absence Recordings DFF. The index indicates the segment number. Eg: IV_ATTRIBUTE_NUMBER_ARR[1] holds the value of segment ATTRIBUTE_NUMBER1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the "exists" command. Eg: if(IV_ATTRIBUTE_NUMBER_ARR.exists(1))
IV_ATTRIBUTE_DATE_ARR	Date_Number	Array of date segments of the Absence Recordings DFF. The index indicates the segment number. Eg: IV_ATTRIBUTE_DATE_ARR[1] holds the value of segment ATTRIBUTE_DATE1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the "exists" command. Eg: if(IV_ATTRIBUTE_DATE_ARR.exists(1))
IV_INFORMATION_CATEGORY	Text	Context of the DDF row for ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_1*	Text	First text segment of the ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_NUMBER1*	Number	First number segment of the ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_DATE1*	Date	First date segment of the ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_ARR	Text_Number	Array of text segments of the ANC_PER_ABS_ENTRIES DDF. The index indicates the segment number. Eg: IV_INFORMATION_ARR[1] holds the value of segment INFORMATION1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the "exists" command. Eg: if(IV_INFORMATION_ARR.exists(1))

Variable Name	Data Type	Description
IV_INFORMATION_NUMBER_ARR	Number_Number	Array of number segments of the ANC_PER_ABS_ENTRIES DDF. The index indicates the segment number. Eg: IV_INFORMATION_NUMBER_ARR[1] holds the value of segment INFORMATION_NUMBER1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the "exists" command. Eg: if(IV_INFORMATION_NUMBER_ARR.exists(1))
IV_INFORMATION_DATE_ARR	Date_Number	Array of date segments of the ANC_PER_ABS_ENTRIES DDF. The index indicates the segment number. Eg: IV_INFORMATION_DATE_ARR[1] holds the value of segment INFORMATION_DATE1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the "exists" command. Eg: if(IV_INFORMATION_DATE_ARR.exists(1))
IV_PAYMENT_DTL_BAND	Number	Identifier of the payment band chosen for the absence linked to an agreement
IV_NOTIFICATION_DATE	Date	Date of illness/injury notification
IV_MATCHING_DATE	Date	Date on which child is matched with the worker for adoption

\*The sequence for this input value can be extended up to 5

## Return Variables

Variable Name	Data Type	Description
REFERENCEDATE	Date	Start date for the Rolling Backward plan term.
DEBUG_MESSAGE	Text	Text passed into this output variable would get logged if application logging is enabled. Useful for debugging.

## Sample Formula

**Requirement:** The organization has a qualification plan with a rolling backward plan term which starts 365 days prior to the absence end date of the absence entry.

**Solution:** A Global Absence Plan Roll Backward End formula such as the one below can be used.

FORMULA NAME: ANC\_PLN\_ROLL\_BKWRD\_END

FORMULA TYPE: Global Absence Plan Roll Backward End

DESCRIPTION: This formula returns the end date for a Rolling Backward qualification plan term

```
INPUTS ARE IV_START_DATE (date), IV_END_DATE (date)

REFERENCE_DATE = ADD_DAYS(IV_END_DATE, - 365)

RETURN REFERENCEDATE
```

## Global Absence Plan Roll Forward Start

The Global Absence Plan Roll Forward Start fast formula returns the reference date till which the existence of a roll forward term is searched for.

If a rolling forward term needs to be searched for 365 days prior to the absence start date, required logic can be composed into this formula and the reference date returned.

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

## Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Qualification**.
4. Click **Continue**.
5. On the Edit Absence Plan page Plan Attributes tab, Plan Term section, choose **Rolling forward** in the **Type** field.
6. Select **Formula** in the **Start Rule** field.
7. Choose the defined formula in the **Start Formula** field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_CATEGORY_ID	Number
ABSENCE_ENTRY_ID	Number

Context Name	Data Type
ABSENCE_MATERNITY_ID	Number
ABSENCE_REASON_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_START_DATE	Date	Start date of absence entry
IV_END_DATE	Date	End date of absence entry
IV_TOTALDURATION	Number	Duration of absence entry
IV_START_DURATION	Number	Start date duration of the absence entry. Applicable when absence entry is being

Variable Name	Data Type	Description
		recorded against an Elapsed based work schedule
IV_END_DURATION	Number	End date duration of the absence entry. Applicable when absence entry is being recorded against an Elapsed based work schedule
IV_START_TIME	Text	Start time on the start date of the absence entry. Applicable when absence entry is being recorded against a Time based work schedule. For elapsed work schedules it will be 00:00.
IV_END_TIME	Text	End time on the end date for the absence being entered. Applicable when absence entry is being recorded against a Time based work schedule. For elapsed work schedules it will be 23:59.
IV_ACTUALCHILDBIRTHDATE	Date	Actual date of childbirth entered for the absence. Applicable for Childbirth/placement absence pattern
IV_ACTUALSTARTDATE	Date	Actual start date of absence. Applicable for Childbirth/placement absence pattern
IV_ACTUALENDDATE	Date	Actual end date of absence. Applicable for Childbirth/placement absence pattern
IV_EXPECTEDCHILDBIRTHDATE	Date	Expected date of childbirth entered for the absence. Applicable for Childbirth/placement absence pattern
IV_PLANNEDSTARTDATE	Date	Expected start date of absence. Applicable for Childbirth/placement absence pattern
IV_PLANNEDENDDATE	Date	Expected end date of absence. Applicable for Childbirth/placement absence pattern
IV_ABSENCE_REASON	Text	Absence reason in the language of the current session
IV_ATTRIBUTE_CATEGORY	Text	Context of the DFF row for Absence Recordings DFF
IV_ATTRIBUTE_1*	Text	First text segment of the Absence Recordings DFF
IV_ATTRIBUTE_NUMBER1*	Number	First number segment of the Absence Recordings DFF

Variable Name	Data Type	Description
IV_ATTRIBUTE_DATE1*	Date	First date segment of the Absence Recordings DFF
IV_ATTRIBUTE_ARR	Text_Number	Array of text segments of the Absence Recordings DFF. The index indicates the segment number. Eg: IV_ATTRIBUTE_ARR[1] holds the value of segment ATTRIBUTE1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_ATTRIBUTE_ARR.exists(1))
IV_ATTRIBUTE_NUMBER_ARR	Number_Number	Array of number segments of the Absence Recordings DFF. The index indicates the segment number. Eg: IV_ATTRIBUTE_NUMBER_ARR[1] holds the value of segment ATTRIBUTE_NUMBER1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_ATTRIBUTE_NUMBER_ARR.exists(1))
IV_ATTRIBUTE_DATE_ARR	Date_Number	Array of date segments of the Absence Recordings DFF. The index indicates the segment number. Eg: IV_ATTRIBUTE_DATE_ARR[1] holds the value of segment ATTRIBUTE_DATE1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_ATTRIBUTE_DATE_ARR.exists(1))
IV_INFORMATION_CATEGORY	Text	Context of the DDF row for ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_1*	Text	First text segment of the ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_NUMBER1*	Number	First number segment of the ANC_PER_ABS_ENTRIES DDF

Variable Name	Data Type	Description
IV_INFORMATION_DATE1*	Date	First date segment of the ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_ARR	Text_Number	Array of text segments of the ANC_PER_ABS_ENTRIES DDF. The index indicates the segment number. Eg: IV_INFORMATION_ARR[1] holds the value of segment INFORMATION1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_INFORMATION_ARR.exists(1))
IV_INFORMATION_NUMBER_ARR	Number_Number	Array of number segments of the ANC_PER_ABS_ENTRIES DDF. The index indicates the segment number. Eg: IV_INFORMATION_NUMBER_ARR[1] holds the value of segment INFORMATION_NUMBER1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_INFORMATION_NUMBER_ARR.exists(1))
IV_INFORMATION_DATE_ARR	Date_Number	Array of date segments of the ANC_PER_ABS_ENTRIES DDF. The index indicates the segment number. Eg: IV_INFORMATION_DATE_ARR[1] holds the value of segment INFORMATION_DATE1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_INFORMATION_DATE_ARR.exists(1))
IV_PAYMENT_DTL_BAND	Number	Identifier of the payment band chosen for the absence linked to an agreement
IV_NOTIFICATION_DATE	Date	Date of Illness/Injury notification
IV_MATCHING_DATE	Date	Date on which child is matched with the worker for adoption

\*The sequence for this input value can be extended up to 5

## Return Variables

Variable Name	Data Type	Description
REFERENCEDATE	Date	Date to which a rolling forward term is searched for.

## Sample Formula

**Requirement:** The organization has a qualification plan with a rolling forward plan term which needs to be searched for 365 days prior to the start date of the absence entry.

**Solution:** A Global Absence Plan Roll Forward Start formula such as the one below can be used.

FORMULA NAME: ANC\_PLN\_ROLL\_FWD\_END

FORMULA TYPE: Global Absence Plan Roll Forward Start

DESCRIPTION: This formula returns the reference date to which a Rolling Forward term is searched for

```
INPUTS ARE IV_START_DATE (date), IV_END_DATE (date)  
REFERENCE_DATE = ADD_DAYS(IV_START_DATE, - 365)  
RETURN REFERENCEDATE
```

## Global Absence Plan Use Rate

The Global Absence Plan Use Rate fast formula can be used to dynamically specify the rate definition associated with the qualification plan depending on custom conditions.

If the rate definition associated with the same qualification plan varies depending on the location of the worker being evaluated, a Global Absence Plan Use Rate can be composed to associate the corresponding rate definition to the Worker.

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence plan definition.

## Navigation

1. In the Absence Administration work area, click **Absence Plans** to open the Absence Plans page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Plan dialog box.
3. In the Plan Type field, select **Qualification**.
4. Click **Continue**.
5. On the Edit Absence Plan page Entries and Balances tab, Rates section, choose **Formula** in the **Absence Payment Rule** field.
6. Select the defined formula in the **Formula** field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_CATEGORY_ID	Number
ABSENCE_ENTRY_ID	Number
ABSENCE_MATERNITY_ID	Number
ABSENCE_REASON_ID	Number
ABSENCE_TYPE_ID	Number
ACCRUAL_PLAN_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_ATTRIBUTE_CATEGORY	Text	Context of the DFF row for Absence Recordings DFF
IV_ATTRIBUTE_1*	Text	First text segment of the Absence Recordings DFF
IV_ATTRIBUTE_NUMBER1*	Number	First number segment of the Absence Recordings DFF
IV_ATTRIBUTE_DATE1*	Date	First date segment of the Absence Recordings DFF
IV_ATTRIBUTE_ARR	Text_Number	Array of text segments of the Absence Recordings DFF. The index indicates the segment number. Eg: IV_ATTRIBUTE_ARR[1] holds the value of segment ATTRIBUTE1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the "exists" command. Eg: if(IV_ATTRIBUTE_ARR.exists(1))
IV_ATTRIBUTE_NUMBER_ARR	Number_Number	Array of number segments of the Absence Recordings DFF. The index indicates the segment number. Eg: IV_ATTRIBUTE_NUMBER_ARR[1] holds the value of segment ATTRIBUTE_NUMBER1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the "exists" command. Eg: if(IV_ATTRIBUTE_NUMBER_ARR.exists(1))
IV_ATTRIBUTE_DATE_ARR	Date_Number	Array of date segments of the Absence Recordings DFF. The index indicates the segment number. Eg: IV_ATTRIBUTE_DATE_ARR[1] holds the value of segment ATTRIBUTE_DATE1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before

Variable Name	Data Type	Description
		using the input value. This can be done using the “exists” command. Eg: if(IV_ATTRIBUTE_DATE_ARR.exists(1))
IV_INFORMATION_CATEGORY	Text	Context of the DDF row for ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_1*	Text	First text segment of the ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_NUMBER1*	Number	First number segment of the ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_DATE1*	Date	First date segment of the ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_ARR	Text_Number	Array of text segments of the ANC_PER_ABS_ENTRIES DDF. The index indicates the segment number. Eg: IV_INFORMATION_ARR[1] holds the value of segment INFORMATION1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_INFORMATION_ARR.exists(1))
IV_INFORMATION_NUMBER_ARR	Number_Number	Array of number segments of the ANC_PER_ABS_ENTRIES DDF. The index indicates the segment number. Eg: IV_INFORMATION_NUMBER_ARR[1] holds the value of segment INFORMATION_NUMBER1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_INFORMATION_NUMBER_ARR.exists(1))
IV_INFORMATION_DATE_ARR	Date_Number	Array of date segments of the ANC_PER_ABS_ENTRIES DDF. The index indicates the segment number. Eg: IV_INFORMATION_DATE_ARR[1] holds the value of segment INFORMATION_DATE1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using

Variable Name	Data Type	Description
		the “exists” command. Eg: if(IV_INFORMATION_DATE_ARR.exists(1))
IV_PAYMENT_DTL_BAND	Number	Identifier of the payment band chosen for the absence linked to an agreement
IV_NOTIFICATION_DATE	Date	Date of Illness/Injury notification
IV_MATCHING_DATE	Date	Date on which child is matched with the worker for adoption

\*The sequence for this input value can be extended up to 5

## Return Variables

Variable Name	Data Type	Description
RATECODE	Text	Rate code for rate associated with qualification plan
DEBUG_MESSAGE	Text	Text passed into this output variable would get logged if application logging is enabled. Useful for debugging

## Sample Formula

**Requirement:** The organization has an absence payment rate rule for the qualification plan that depends on the hourly/salaried nature of the Worker. Separate rate definition rules exist for each category of workers.

**Solution:** A Global Absence Plan Use Rate formula such as the one below can be used.

FORMULA NAME: ANC\_PLN\_RATE

FORMULA TYPE: Global Absence Plan Use Rate

DESCRIPTION: This formula returns the rate code for the rate definition that needs to be associated with the qualification plan

```

DEFAULT FOR PER_ASG_HOURLY_SALARIED_CODE IS 'H'

IF (PER_ASG_HOURLY_SALARIED_CODE = 'H')
THEN
(
RATECODE = 'HOURLY_RATE'
)
ELSE
(
RATECODE = 'ANNUAL_RATE'
)

RETURN RATECODE

```

# Formulas for Absence Type

Use the Absence Types pages to define absence type rules. However, if you want to define other special rules to suit your requirement, you can write your own formulas.

## Formulas for Absence Types

The following table lists the aspects of an absence type for which you can write a formula and identifies the formula type for each.

Rule	Description	Formula Type to Use
Conversion	Method to convert the absence duration to other units of measure. For example, your workers' work schedules are in work hours, but you want to display the duration in work days.	Global Absence Type Duration  You can use the formula to convert absence duration values that are in work days or work hours only.
Validation	Rules in addition to the ones that you can define on the Absence Types pages to check the validity of the absence.	Global Absence Entry Validation

## Global Absence Entry Validation

The Global Absence Entry Validation fast formula can be used in cases where a custom data validation or business rule validation needs to be performed when an absence entry is being made.

An organization might have a rule that allows a worker to record an absence entry of type Maternity Leave – Unpaid, only in sequence with a regular Maternity Leave absence entry. In such a case, a validation formula can be composed which looks at the Workers absence history to identify if there is a Maternity Leave that is ending on the day just before the absence entry of Maternity Leave - Unpaid.

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence type definition.

## Navigation

1. In the Absence Administration work area, click **Absence Types** to open the Absence Types page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Type dialog box.
3. Click **Continue**.

4. On the Create Absence Type page Type Attributes tab, General Attributes section, select the defined formula in the **Validation Formula** field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE AGREEMENT_ID	Number
ABSENCE CERTIFICATION_ID	Number
ABSENCE ENTRY_ID	Number
ABSENCE REASON_ID	Number
ABSENCE TYPE_ID	Number
DATE EARNED	Date
EFFECTIVE DATE	Date
END DATE	Date
ENTERPRISE ID	Number
HR ASSIGNMENT ID	Number
HR RELATIONSHIP ID	Number
HR TERM ID	Number
JOB ID	Number
LEGAL EMPLOYER ID	Number
LEGISLATIVE DATA GROUP ID	Number
ORGANIZATION ID	Number
PAYROLL ASSIGNMENT ID	Number
PAYROLL ID	Number
PAYROLL RELATIONSHIP ID	Number
PAYROLL TERM ID	Number
PERSON AGREEMENT ID	Number
PERSON ID	Number
START DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_START_DATE	Date	Start date of absence entry
IV_END_DATE	Date	End date of absence entry
IV_TOTALDURATION	Number	Duration of absence entry
IV_START_DURATION	Number	Start date duration of the absence entry. Applicable when absence entry is being recorded against an Elapsed based work schedule
IV_END_DURATION	Number	End date duration of the absence entry. Applicable when absence entry is being recorded against an Elapsed based work schedule
IV_START_TIME	Text	Start time on the start date of the absence entry. Applicable when absence entry is being recorded against a Time based work schedule. For elapsed work schedules it will be 00:00.
IV_END_TIME	Text	End time on the end date for the absence being entered. Applicable when absence entry is being recorded against a Time based work schedule. For elapsed work schedules it will be 23:59.
IV_ACTUALCHILDBIRTHDATE	Date	Actual date of childbirth entered for the absence. Applicable for Childbirth/placement absence pattern
IV_ACTUALSTARTDATE	Date	Actual start date of absence. Applicable for Childbirth/placement absence pattern
IV_ACTUALENDDATE	Date	Actual end date of absence. Applicable for Childbirth/placement absence pattern
IV_EXPECTEDCHILDBIRTHDATE	Date	Expected date of childbirth entered for the absence. Applicable for Childbirth/placement absence pattern
IV_PLANNEDSTARTDATE	Date	Expected start date of absence. Applicable for Childbirth/placement absence pattern
IV_PLANNEDENDDATE	Date	Expected end date of absence. Applicable for Childbirth/placement absence pattern

Variable Name	Data Type	Description
IV_ABSENCE_REASON	Text	Absence reason in the language of the current session
IV_ATTRIBUTE_CATEGORY	Text	Context of the DFF row for Absence Recordings DFF
IV_ATTRIBUTE_1*	Text	First text segment of the Absence Recordings DFF
IV_ATTRIBUTE_NUMBER1*	Number	First number segment of the Absence Recordings DFF
IV_ATTRIBUTE_DATE1*	Date	First date segment of the Absence Recordings DFF
IV_ATTRIBUTE_ARR	Text_Number	Array of text segments of the Absence Recordings DFF. The index indicates the segment number. Eg: IV_ATTRIBUTE_ARR[1] holds the value of segment ATTRIBUTE1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the "exists" command. Eg: if(IV_ATTRIBUTE_ARR.exists(1))
IV_ATTRIBUTE_NUMBER_ARR	Number_Number	Array of number segments of the Absence Recordings DFF. The index indicates the segment number. Eg: IV_ATTRIBUTE_NUMBER_ARR[1] holds the value of segment ATTRIBUTE_NUMBER1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the "exists" command. Eg: if(IV_ATTRIBUTE_NUMBER_ARR.exists(1))
IV_ATTRIBUTE_DATE_ARR	Date_Number	Array of date segments of the Absence Recordings DFF. The index indicates the segment number. Eg: IV_ATTRIBUTE_DATE_ARR[1] holds the value of segment ATTRIBUTE_DATE1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using

Variable Name	Data Type	Description
		the “exists” command. Eg: if(IV_ATTRIBUTE_DATE_ARR.exists(1))
IV_INFORMATION_CATEGORY	Text	Context of the DDF row for ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_1*	Text	First text segment of the ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_NUMBER1*	Number	First number segment of the ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_DATE1*	Date	First date segment of the ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_ARR	Text_Number	Array of text segments of the ANC_PER_ABS_ENTRIES DDF. The index indicates the segment number. Eg: IV_INFORMATION_ARR[1] holds the value of segment INFORMATION1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_INFORMATION_ARR.exists(1))
IV_INFORMATION_NUMBER_ARR	Number_Number	Array of number segments of the ANC_PER_ABS_ENTRIES DDF. The index indicates the segment number. Eg: IV_INFORMATION_NUMBER_ARR[1] holds the value of segment INFORMATION_NUMBER1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_INFORMATION_NUMBER_ARR.exists(1))
IV_INFORMATION_DATE_ARR	Date_Number	Array of date segments of the ANC_PER_ABS_ENTRIES DDF. The index indicates the segment number. Eg: IV_INFORMATION_DATE_ARR[1] holds the value of segment INFORMATION_DATE1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using

Variable Name	Data Type	Description
		the “exists” command. Eg: if(IV_INFORMATION_DATE_ARR.exists(1))
IV_PAYMENT_DTL_BAND	Number	Identifier of the payment band chosen for the absence linked to an agreement
IV_NOTIFICATION_DATE	Date	Date of Illness/Injury notification
IV_MATCHING_DATE	Date	Date on which child is matched with the worker for adoption

\*The sequence for this input value can be extended up to 5

## Return Variables

Variable Name	Data Type	Description
VALID	Text	Determines validity of absence entry. ‘N’ throws an error message and stops the transaction from being submitted
ERROR_MESSAGE	Text	Custom error message for invalid absence entry. Error message can either be written into the formula or defined via Manage Messages and invoked here.
ERROR_CODE	Text	Identifies if the message is Error or Warning.. Valid values are ‘E’ for Error and ‘W’ for Warning.
TOKEN_NAME	Text_Number	Array of token names, if any, in the message configured via Manage Messages
TOKEN_VALUE	Text_Number	Array of values passed to each token name in the message
DEBUG_MESSAGE	Text	Text passed into this output variable would get logged if application logging is enabled. Useful for debugging.

## Sample Formula

**Requirement:** The organization has a Sickness absence entitlement policy according to which a worker can record a regular sickness absence entry only after the completion of 365 days from their last Unpaid Sickness entry.

**Solution:** A Global Absence Entry Validation formula such as the one below can be used.

FORMULA NAME: ANC\_VALID

## FORMULA TYPE: Global Absence Entry Validation

DESCRIPTION: This formula validates the Sickness absence entry

```

DEFAULT FOR IV_START_DATE IS '4712/12/31 00:00:00' (date)
DEFAULT FOR IV_END_DATE IS '4712/12/31 00:00:00' (date)
DEFAULT_DATA_VALUE for ANC_PER_ABS_ENTRs_ABSENCE_ENTRY_ID_ARR is 0
DEFAULT FOR ANC_ABS_ENTRs_ABSENCE_TYPE_ID is 0
DEFAULT FOR ANC_ABS_ENTRs_ABSENCE_STATUS_CD IS ''
DEFAULT FOR ANC_ABS_TYP_NAME IS ''
DEFAULT FOR ANC_ABS_ENTRs_END_DATE IS '4712/12/31 00:00:00' (date)

INPUTS ARE IV_END_DATE (date), IV_START_DATE (date)

lc_sickness_unpaid_name = 'Sickness - Unpaid'
ld_start_date = TO_DATE('1951-01-01','yyyy-MM-dd')
ld_sickness_end_date = TO_DATE('4712-12-31','yyyy-MM-dd')

i=1

CHANGE_CONTEXTS (START_DATE=ld_start_date, END_DATE=IV_END_DATE)
(
  WHILE ANC_PER_ABS_ENTRs_ABSENCE_ENTRY_ID_ARR.exists(i)
  LOOP
  (
  CHANGE_CONTEXTS (ABSENCE_ENTRY_ID = ANC_PER_ABS_ENTRs_ABSENCE_ENTRY_ID_ARR[i])
  (
  CHANGE_CONTEXTS (ABSENCE_TYPE_ID = ANC_ABS_ENTRs_ABSENCE_TYPE_ID)
  (
  IF (lc_sickness_unpaid_name = ANC_ABS_TYP_NAME AND ANC_ABS_ENTRs_ABSENCE_STATUS_CD = 'SUBMITTED')
  THEN
  (
  IF (365 > DAYS_BETWEEN(TRUNC(IV_START_DATE), ANC_ABS_ENTRs_END_DATE) + 1)
  THEN
  (
  VALID = 'N'
  ERROR_MESSAGE = 'ORA_CUSTOM_ERROR_MESSAGE'
  RETURN VALID, ERROR_MESSAGE
  )
  )
  )
  )
  i=i+1
  )
)
VALID = 'Y'
RETURN VALID

```

## Global Absence Linkage Exclusion

The Global Absence Linkage Exclusion fast formula can be used to define rules for skipping absence entry linkages in cases where linkage rules have been defined at the absence type setup level.

An organization might have a linkage rule that links together Sickness absences. However, if the Sickness absence entry is only of a single day duration, the business might wish to exclude the specific absence entry from being linked. In such a case, a linkage exclusion formula can be composed which checks the duration of the entered absence and informs the application whether to exclude the absence entry from a potential linkage.

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence type definition.

## Navigation

1. In the Absence Administration work area, click **Absence Types** to open the Absence Types page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Type dialog box.
3. Choose Illness or injury in the **Pattern** field and click **Continue**.
4. On the Create Absence Type page Type Attributes tab, **Linkage** section, choose Duration or Chained in the **Linkage Rule** field.
5. Select the defined formula in the **Exclusion Formula** field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_REASON_ID	Number
ABSENCE_TYPE_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON AGREEMENT_ID	Number

Context Name	Data Type
PERSON_ID	Number
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_START_DATE	Date	Start date of absence entry
IV_END_DATE	Date	End date of absence entry
IV_TOTALDURATION	Number	Duration of absence entry
IV_START_DURATION	Number	Start date duration of the absence entry. Applicable when absence entry is being recorded against an Elapsed based work schedule
IV_END_DURATION	Number	End date duration of the absence entry. Applicable when absence entry is being recorded against an Elapsed based work schedule
IV_START_TIME	Text	Start time on the start date of the absence entry. Applicable when absence entry is being recorded against a Time based work schedule. For elapsed work schedules it will be 00:00.
IV_END_TIME	Text	End time on the end date for the absence being entered. Applicable when absence entry is being recorded against a Time based work schedule. For elapsed work schedules it will be 23:59.
IV_ACTUALCHILDBIRTHDATE	Date	Actual date of childbirth entered for the absence. Applicable for Childbirth/placement absence pattern
IV_ACTUALSTARTDATE	Date	Actual start date of absence. Applicable for Childbirth/placement absence pattern
IV_ACTUALENDDATE	Date	Actual end date of absence. Applicable for Childbirth/placement absence pattern
IV_EXPECTEDCHILDBIRTHDATE	Date	Expected date of childbirth entered for the absence. Applicable for Childbirth/placement absence pattern

Variable Name	Data Type	Description
IV_PLANNEDSTARTDATE	Date	Expected start date of absence. Applicable for Childbirth/placement absence pattern
IV_PLANNEDENDDATE	Date	Expected end date of absence. Applicable for Childbirth/placement absence pattern
IV_ABSENCE_REASON	Text	Absence reason in the language of the current session
IV_ATTRIBUTE_CATEGORY	Text	Context of the DFF row for Absence Recordings DFF
IV_ATTRIBUTE_1*	Text	First text segment of the Absence Recordings DFF
IV_ATTRIBUTE_NUMBER1*	Number	First number segment of the Absence Recordings DFF
IV_ATTRIBUTE_DATE1*	Date	First date segment of the Absence Recordings DFF
IV_ATTRIBUTE_ARR	Text_Number	Array of text segments of the Absence Recordings DFF. The index indicates the segment number. Eg: IV_ATTRIBUTE_ARR[1] holds the value of segment ATTRIBUTE1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the "exists" command. Eg: if(IV_ATTRIBUTE_ARR.exists(1))
IV_ATTRIBUTE_NUMBER_ARR	Number_Number	Array of number segments of the Absence Recordings DFF. The index indicates the segment number. Eg: IV_ATTRIBUTE_NUMBER_ARR[1] holds the value of segment ATTRIBUTE_NUMBER1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the "exists" command. Eg: if(IV_ATTRIBUTE_NUMBER_ARR.exists(1))
IV_ATTRIBUTE_DATE_ARR	Date_Number	Array of date segments of the Absence Recordings DFF. The index indicates the segment number. Eg: IV_ATTRIBUTE_DATE_ARR[1] holds the value of segment ATTRIBUTE_DATE in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the "exists" command. Eg: if(IV_ATTRIBUTE_DATE_ARR.exists(1))

Variable Name	Data Type	Description
		DATE1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_ATTRIBUTE_DATE_ARR.exists(1))
IV_INFORMATION_CATEGORY	Text	Context of the DDF row for ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_1*	Text	First text segment of the ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_NUMBER1*	Number	First number segment of the ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_DATE1*	Date	First date segment of the ANC_PER_ABS_ENTRIES DDF
IV_INFORMATION_ARR	Text_Number	Array of text segments of the ANC_PER_ABS_ENTRIES DDF. The index indicates the segment number. Eg: IV_INFORMATION_ARR[1] holds the value of segment INFORMATION1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_INFORMATION_ARR.exists(1))
IV_INFORMATION_NUMBER_ARR	Number_Number	Array of number segments of the ANC_PER_ABS_ENTRIES DDF. The index indicates the segment number. Eg: IV_INFORMATION_NUMBER_ARR[1] holds the value of segment INFORMATION_NUMBER1 in ANC_PER_ABS_ENTRIES table. The index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the “exists” command. Eg: if(IV_INFORMATION_NUMBER_ARR.exists(1))
IV_INFORMATION_DATE_ARR	Date_Number	Array of date segments of the ANC_PER_ABS_ENTRIES DDF. The index indicates the segment number. Eg: IV_INFORMATION_DATE_ARR[1] holds the value of segment INFORMATION_DATE1 in ANC_PER_ABS_ENTRIES table. The

Variable Name	Data Type	Description
		index begins at 1 and ends at 30. If a particular index is used and data does not exist, the application would throw an error. Hence it is a good practice to first check if data exists before using the input value. This can be done using the "exists" command. Eg: if(IV_INFORMATION_DATE_ARR.exists(1))
IV_PAYMENT_DTL_BAND	Number	Identifier of the payment band chosen for the absence linked to an agreement
IV_NOTIFICATION_DATE	Date	Date of Illness/Injury notification
IV_MATCHING_DATE	Date	Date on which child is matched with the worker for adoption

\*The sequence for this input value can be extended up to 5

## Return Variables

Variable Name	Data Type	Description
EXCLUDE	Text	Determines whether the absence entry needs to be excluded from a linkage. Valid values are 'Y' or 'N'
DEBUG_MESSAGE	Text	Text passed into this output variable would get logged if application logging is enabled. Useful for debugging

## Sample Formula

**Requirement:** The organization has a linkage rule setup for Sickness absence entries. However, the Sickness absence entry needs to be excluded from linkages if the absence reason entered is a specific value.

**Solution:** A Global Absence Linkage Exclusion formula such as the one below can be used.

FORMULA NAME: ANC\_LNKG\_EXCL

FORMULA TYPE: Global Absence Linkage Exclusion

DESCRIPTION: This formula determines if the absence entry needs to be excluded from a linkage

```

DEFAULT FOR IV_ABSENCE_REASON IS ''

INPUTS ARE IV_ABSENCE_REASON (text)
EXCLUDE = 'N'
IF (IV_ABSENCE_REASON = 'OCCUPATIONAL')
THEN
(
EXCLUDE = 'Y'

```

```
)  
RETURN EXCLUDE
```

## Global Absence Type Duration

The Global Absence Type Duration fast formula can be used to override the default calculation logic that the application uses to determine the absence entry duration and accrual plan usage against absence entries.

In an organization when a worker makes an absence entry for an absence type that is measured in Days, there might be a need to have the total duration pro-rated based on the scheduled work hours of the Worker for those days of absence. In such a case, a type duration formula can be composed with the required logic of calculating the pro-rated duration so that it replaces the default duration calculation rules in the application.

The same formula needs to return the required duration when the entry is processed as a single unit (between start date and end date), as well as when processed for each day between the start and end dates.

## Configuration Point in HCM Cloud

If you have created this formula, you can attach this formula to the absence type definition. This is currently available only for absence types with UoM as Hours or Days.

## Navigation

1. In the Absence Administration work area, click **Absence Types** to open the Absence Types page.
2. On the Search Results section toolbar, click **Create** to open the Create Absence Type dialog box.
3. Click **Continue**.
4. On the Create Absence Type page Type Attributes tab, General Attributes section, select the defined formula in the **Conversion Formula** field.

## Contexts

The following contexts are available in this formula type:

Context Name	Data Type
ABSENCE_ENTRY_ID	Number
ABSENCE_TYPE_ID	Number
DATE_EARNED	Date
EFFECTIVE_DATE	Date
END_DATE	Date
ENTERPRISE_ID	Number
HR_ASSIGNMENT_ID	Number
HR_RELATIONSHIP_ID	Number

Context Name	Data Type
HR_TERM_ID	Number
JOB_ID	Number
LEGAL_EMPLOYER_ID	Number
LEGISLATIVE_DATA_GROUP_ID	Number
ORGANIZATION_ID	Number
PAYROLL_ASSIGNMENT_ID	Number
PAYROLL_ID	Number
PAYROLL_RELATIONSHIP_ID	Number
PAYROLL_TERM_ID	Number
PERSON_ID	Number
START_DATE	Date

## Input Values

Variable Name	Data Type	Description
IV_START_DATE	Date	Start date of absence entry
IV_END_DATE	Date	End date of absence entry
IV_START_DURATION	Number	Start date duration of the absence entry. Applicable when absence entry is being recorded against an Elapsed based work schedule
IV_END_DURATION	Number	End date duration of the absence entry. Applicable when absence entry is being recorded against an Elapsed based work schedule
IV_START_TIME	Text	Start time on the start date of the absence entry. Applicable when absence entry is being recorded against a Time based work schedule. For elapsed work schedules it will be 00:00.
IV_END_TIME	Text	End time on the end date for the absence being entered. Applicable when absence entry is being recorded against a Time based work schedule. For elapsed work schedules it will be 23:59.
IV_UOM	Text	Unit of Measure

Variable Name	Data Type	Description
IV_ABSENCE_TYPE_ID	Number	Absence type ID for the absence being recorded.
IV_USER_MODE	Text	Identifies the role of the person recording the absence.

## Return Variables

Variable Name	Data Type	Description
DURATION	Number	Duration of absence entry

## Sample Formula

**Requirement:** The organization requires that for a particular absence type for which partial day absence recording is not allowed, and the duration is measured in Days, the total duration needs to be pro-rated by multiplying with the FTE value of the Worker.

**Solution:** A Global Absence Type duration formula such as the one below can be used.

FORMULA NAME: ANC\_DURCAL

FORMULA TYPE: Global Absence Type Duration Formula

DESCRIPTION: This formula returns the pro-rated duration for absence entries based on FTE

```

DEFAULT FOR IV_START_DATE IS '4712/12/31 00:00:00' (date)
DEFAULT FOR IV_END_DATE IS '4712/12/31 00:00:00' (date)
DEFAULT FOR IV_START_TIME IS '00:00'
DEFAULT FOR IV_END_TIME IS '23:59'
DEFAULT FOR PER_ASG_FTE_VALUE IS 1

INPUTS ARE IV_START_DATE (date), IV_END_DATE (date), IV_START_TIME(text), IV_END_TIME(text)

ln_entry_duration_d = 0
ln_unrounded_duration_d = 0

ld_period_start_date = to_date(to_char(IV_START_DATE, 'DD/MM/YYYY') || ' ' || IV_START_TIME, 'DD/MM/YYYY
HH24:MI:SS')
ld_period_end_date = to_date(to_char(IV_END_DATE, 'DD/MM/YYYY') || ' ' || IV_END_TIME, 'DD/MM/YYYY HH24:MI:SS')

ln_entry_duration_d = GET_PAY_AVAILABILITY ('ASSIGN',
ld_period_start_date,
ld_period_end_date,
'Y',
'Y',
'Y',
'Y',
'D')

ln_unrounded_duration_d = ln_entry_duration_d * PER_ASG_FTE_VALUE

DURATION = ROUND(ln_unrounded_duration_d,2)

```

**RETURN\_DURATION**

## Frequently Used Formula Functions

Formula functions are used for complex data fetch routines for which database items are not sufficient. These formula functions usually have input parameters that you can customize within a fast formula to ensure the formula function returns customized data.

### GET\_PLAN\_BALANCE

This formula function can be used to return the accrual balance of an absence plan calculated up until the latest 'Last Balance Calculation Date' which is lesser than or earlier than effective date.

#### Mandatory Contexts

- HR\_ASSIGNMENT\_ID
- PERSON\_ID
- EFFECTIVE\_DATE
- LEGISLATIVE\_DATA\_GROUP

#### Parameters

Parameter name	Data Type	Description
p_pl_name	Character	Name of accrual plan

#### Data Returned

The accrual balance in Number format.

### GET\_ACCRUAL\_BALANCE

This formula function can be used to return the accrual balance of an absence plan calculated up until the latest 'Last Balance Calculation Date' for an enrollment which is active as of the effective date.

#### Mandatory Contexts

- HR\_ASSIGNMENT\_ID
- PERSON\_ID
- EFFECTIVE\_DATE

- ACCRUAL\_PLAN\_ID

## Data Returned

The accrual balance in Number format.

# GET\_ABSENCE\_COUNTS

This formula function can be used to calculate the number of absence entries against a person and return the total durations between a start date and end date passed as parameters.

## Mandatory Contexts

This formula function does not require any contexts to return values.

## Parameters

Parameter name	Data Type	Description
p_person_id	Number	Person id for whom absence entries durations and occurrences need to be fetched
p_include_type_id	Number	Absence type id to be included
p_exclude_type_id	Number	Absence type id to be excluded
p_include_categor_id	Number	Absence category id to be included
p_exclude_category_id	Number	Absence category id to be excluded
p_include_reason_id	Number	Absence reason to be included
p_exclude_reason_id	Number	Absence reason to be excluded
p_start_date_from	Date	Absences that end on or after this date are included in the calculations
p_start_date_to	Date	Absences that end on or before this date are included in the calculations
p_duration_days	Number	Out parameter – sum of durations of absence entries that were recorded in Days
p_duration_hours	Number	Out parameter – sum of durations of absence entries that were recorded in Hours
p_duration_cal	Number	Out parameter – sum of durations of absence entries that were recorded in Calendar Days
p_duration_weeks	Number	Out parameter – sum of durations of absence entries that were recorded in Weeks

Parameter name	Data Type	Description
p_duration_months	Number	Out parameter – sum of durations of absence entries that were recorded in Months
p_duration_years	Number	Out parameter – sum of durations of absence entries that were recorded in Years
p_occurrences	Number	Out parameter – count of absence entries

## Data Returned

Apart from the out parameters returned above, the return data for the formula function is a Number value of 0 in all cases.

## GET\_ABS\_MIN\_MAX\_DATES

This formula function can be used to look at the absence entries against a person and return the earliest start date and the latest end date in a period.

### Mandatory Contexts

This formula function does not require any contexts to return values.

### Parameters

Parameter name	Data Type	Description
p_person_id	Number	Person id for whom absence entries durations and occurrences need to be fetched
p_include_type_id	Number	Absence type id to be included
p_exclude_type_id	Number	Absence type id to be excluded
p_include_categor_id	Number	Absence category id to be included
p_exclude_category_id	Number	Absence category id to be excluded
p_include_reason_id	Number	Absence reason to be included
p_exclude_reason_id	Number	Absence reason to be excluded
p_start_date_from	Date	Absences that end on or after this date are included in the calculations
p_start_date_to	Date	Absences that end on or before this date are included in the calculations

Parameter name	Data Type	Description
p_min_date	Date	Out parameter - Earliest absence start date in the period
p_max_date	Date	Out parameter – Latest absence end date in the period

## Data Returned

Apart from the out parameters returned above, the return data for the formula function is a Number value of 0 in all cases.

## GET\_BAL\_COMP\_VAL

This formula function would return the sum of accrual entries of a particular type within a period.

### Mandatory Contexts

- PERSON\_ID
- HR\_ASSIGNMENT\_ID
- EFFECTIVE\_DATE
- LEGISLATIVE\_DATA\_GROUP\_ID

## Parameters

Parameter name	Data Type	Description
p_pl_name	Character	Name of accrual plan
p_start_date	Date	Start date of period for when accrual entry amounts are required
p_end_date	Date	End date of period for which accrual entry amounts are required
p_type	Character	The accrual entry type for which information is required
p_adj_reason	Character	If information is required for 'adjustment' type then a specific adjustment reason can also be specified

## Data Returned

The sum accrual entries of the particular type between within the period.

## GET\_BAL\_HDR\_VAL

This formula function fetches summary level accrual information for the first accrual period after effective date for which accrual has been run.

### Mandatory Contexts

- PERSON\_ID
- HR\_ASSIGNMENT\_ID
- EFFECTIVE\_DATE
- LEGISLATIVE\_DATA\_GROUP\_ID

### Parameters

Parameter name	Data Type	Description
p_pl_name	Character	The name of the accrual plan for which accrual and balance information is required
p_begin_bal	Number	Out parameter - The accrual balance at the start of the repeating period
p accrued	Number	Out parameter - The amount of time accrued in the repeating period
p_used	Number	Out parameter - The amount of time used during the repeating period
p_end_bal	Number	Out parameter - The final balance as of the end of the repeating period
p_accrual_period	Date	Out parameter - The effective date of the repeating period

## Data Returned

The function returns a value '1' if data is found and '0' if no data is found.

## GET\_ENRT\_DTLS

This formula function can be used to fetch information regarding key dates relevant for an accrual plan enrollment whose start date is earlier than the effective date.

### Mandatory Contexts

- PERSON\_ID
- HR\_ASSIGNMENT\_ID
- EFFECTIVE\_DATE
- LEGISLATIVE\_DATA\_GROUP\_ID

### Parameters

Parameter name	Data Type	Description
p_plan_name	Character	The name of the accrual plan for which enrollment information is required
p_enrt_st_dt	Date	Out parameter – Start date of enrollment
p_enrt_end_dt	Date	Out parameter – End date of enrollment
p_last_accrual_run	Date	Out parameter – Last date as of which accrual has been calculated.

### Data Returned

The function returns a value ‘1’ if data is found and ‘0’ if no data is found.

## GET\_ENTITLEMENTS

This formula function returns the number of plan entitlements of a particular band or payment factor consumed by absence entries in a period.

### Mandatory Contexts

- PERSON\_ID
- LEGISLATIVE\_DATA\_GROUP\_ID

## Parameters

Parameter name	Data Type	Description
p_pl_name	Character	The name of the qualification plan for which the entitlement usage needs to be fetched
p_band_name	Character	The entitlement payment band name for within the qualification plan
p_band_percentage	Number	Instead of the entitlement band name, the payment percentage can also be passed. Payment factor takes precedence over band name.
p_start_date	Date	Start of the period for which entitlement usage data needs to be fetched
p_end_date	Date	End of the period for which entitlement usage data needs to be fetched

## Data Returned

The entitlement units against the band or pay factor consumed during the specified period.

## GET\_ABSENCE\_DAYS\_PER\_TYPE

This formula function returns the sum of absence days of a particular type in a period.

### Mandatory Contexts

PERSON\_ID

## Parameters

Parameter name	Data Type	Description
p_absence_type_name	Character	The name of the absence type for which the number of days need to be summed
p_start_date	Date	Start of the period for which absence days need to be summed
p_end_date	Date	End of the period for which absence days need to be summed

## Data Returned

The sum of absence days in Number format.

## Database Items

In accordance with the fast formula architecture, only database items whose contexts are available within the formula type maybe used in the various absence formulas. Information about the contexts associated with DBIs and the SQL WHERE clause behind the DBI is stored in the HCM Cloud database and can be retrieved using the following SQL query. Replace <ENTER DBI NAME HERE> with the actual DBI name with quotes. This information is to be used to help determine which DBI should be used in a formula.

```
SELECT d.base_user_name DBI_NAME
, d.data_type DBI_DATA_TYPE
, d.definition_text SELECT_CLAUSE
, r.text WHERE_CLAUSE
, (select listagg('<' || rcu.sequence_no || ',' || c.base_context_name || '>', ', ')
within group (order by rcu.sequence_no)
from ff_route_context_usages rcu
, ff_contexts_b c
where rcu.route_id = r.route_id
and rcu.context_id = c.context_id) ROUTE_CONTEXT_USAGES
from ff_database_items_b d
, ff_user_entities_b u
, ff_routes_b r
where d.base_user_name = <ENTER DBI NAME HERE>
and d.user_entity_id = u.user_entity_id
and r.route_id = u.route_id;
```

To find all DBIs based on ANC tables use the following query.

```
SELECT d.base_user_name DBI_NAME
, d.data_type DBI_DATA_TYPE
, d.definition_text SELECT_CLAUSE
, r.text WHERE_CLAUSE
, (select listagg('<' || rcu.sequence_no || ',' || c.base_context_name || '>', ', ')
within group (order by rcu.sequence_no)
from ff_route_context_usages rcu
, ff_contexts_b c
where rcu.route_id = r.route_id
and rcu.context_id = c.context_id) ROUTE_CONTEXT_USAGES
from ff_database_items_b d
, ff_user_entities_b u
, ff_routes_b r
where UPPER(d.base_user_name) LIKE 'ANC%'
and d.user_entity_id = u.user_entity_id
and r.route_id = u.route_id;
```

# 7 Formulas for Compensation Plans

## Overview of Formulas for Compensation Plans

Use compensation formulas in your application when you perform these actions:

- Specify compensation worksheet default values
- Refresh or process specific employees only
- Define how local currency is determined for a plan component
- Create unique hierarchies for compensation

This table introduces the types of formulas that you use within your compensation application.

Formula Type	Description	Example
Compensation Currency Selection	Confirm the currency associated with a workforce compensation component.	Verify the currency of a plan based on a component ID.
Compensation Default and Override	Alter the default values populated in a column for a workforce compensation plan.	Truncate a salary amount so there are four decimals.
Compensation Hierarchy Determination	Locate the hierarchy for an associated workforce compensation plan.	Find the name of an employee's manager using an assignment ID.
Compensation Person Selection	Verify the person selected for an associated workforce compensation plan.	Check if a person is eligible to be paid by a specific legislative data group.
Total Compensation Item Formula	Access compensation information that isn't stored in the other predefined item source types.	Return multiple variables including a person's ID, their start date, and their end date.

## Formula Type Is Compensation Currency Selection

To determine the local currency for a workforce compensation component, you can create formulas using the Compensation Currency Selection formula type.

Here's a comprehensive list of the contexts available to this type of formula:

- DATE\_EARNED
- EFFECTIVE\_DATE
- END\_DATE

- START\_DATE
- HR\_ASSIGNMENT\_ID
- HR\_TERM\_ID
- JOB\_ID
- LEGISLATIVE\_DATA\_GROUP\_ID
- COMPENSATION\_RECORD\_TYPE
- ORGANIZATION\_ID
- PAYROLL\_ASSIGNMENT\_ID
- PAYROLL\_RELATIONSHIP\_ID
- PAYROLL\_TERM\_ID
- PERSON\_ID

The database items available for this type of formula are related to Person, Assignment, Salary, Element Entries, Compensation Record, and From and End Dates.

These are the input variables available to this type of formula:

Input Value	Data Type	Required	Description
CMP_IV_PLAN_ID	Number	Y	Plan ID
CMP_IV_ASSIGNMENT_ID	Number	Y	Assignment ID
CMP_IV_PERIOD_ID	Number	Y	Period ID
CMP_IV_COMPONENT_ID	Number	Y	Component ID
CMP_IV_PLAN_START_DATE	Date	Y	Plan Start Date
CMP_IV_PLAN_END_DATE	Date	Y	Plan End Date
CMP_IV_PLAN_EXTRACTION_DATE	Date	Y	Plan Extraction Date
CMP_IV_PLAN_ELIG_DATE	Date	Y	Plan Eligibility Date
CMP_IV_PERFORMANCE_EFF_DATE	Date	Y	Performance Effective Date
CMP_IV_PROMOTION_EFF_DATE	Date	Y	Promotion Effective Date
CMP_IV_XCHG_RATE_DATE	Date	Y	Currency Conversion Date
CMP_IV_ASSIGNMENT_ID	Number	Y	Assignment ID
CMP_IV_PERSON_ID	Number	Y	Worker ID

These are the return variables available to this type of formula:

Return Value	Data Type	Required	Description
L_CURR_CODE	Char	N	Currency code from the formula

This sample formula determines the currency for a plan based on the component ID.

```
*****  
FORMULA NAME : Compensation Currency Selection Formula  
FORMULA TYPE : Compensation Currency Selection  
DESCRIPTION: It returns the currency code based on component_id.  
*****  
***** INPUT VALUES DEFAULTS BEGIN *****  
INPUTS ARE CMP_IV_ASSIGNMENT_ID (number), CMP_IV_PLAN_ID (number), CMP_IV_PERIOD_ID (number),  
CMP_IV_COMPONENT_ID (number)  
***** INPUT VALUES DEFAULTS ENDS*****  
***** FORMULA SECTION BEGIN *****  
DEFAULT FOR CMP_IV_COMPONENT_ID IS 0  
l_curr_code = 'XXX'  
IF (CMP_IV_COMPONENT_ID = 489) THEN  
(  
l_curr_code = 'USD'  
)  
ELSE IF (CMP_IV_COMPONENT_ID = 490) THEN  
(  
l_curr_code = 'GBP'  
)  
RETURN l_curr_code  
***** FORMULA SECTION END *****
```

#### Related Topics

- [Formula Compilation Errors](#)
- [Formula Execution Errors](#)
- [When do I run the Compile Formula process?](#)
- [Example of Writing a Fast Formula Using Formula Text](#)

## Formula Type Is Compensation Default and Override

To determine the default values for a worksheet column in a workforce compensation plan, you can create formulas using the Compensation Default and Override formula type.

Here's a comprehensive list of the contexts available to this type of formula:

- DATE\_EARNED
- EFFECTIVE\_DATE
- END\_DATE
- START\_DATE
- HR\_ASSIGNMENT\_ID
- HR\_TERM\_ID
- JOB\_ID
- LEGISLATIVE\_DATA\_GROUP\_ID
- COMPENSATION\_RECORD\_TYPE
- ORGANIZATION\_ID

- PAYROLL\_ASSIGNMENT\_ID
- PAYROLL\_RELATIONSHIP\_ID
- PAYROLL\_TERM\_ID
- PERSON\_ID

The database items available for this type of formula are related to Person, Assignment, Salary, Element Entries, Compensation Record, and From and End Dates.

These are the input variables available to this type of formula:

Input Value	Data Type	Required	Description
CMP_IV_PLAN_ID	Number	Y	Unique numeric identifier for the workforce compensation plan
CMP_IV_PERIOD_ID	Number	Y	Unique numeric identifier for the fiscal calendar period
CMP_IV_COMPONENT_ID	Number	Y	Unique numeric identifier for the workforce compensation plan component
CMP_IV_ITEM_NAME	Char	Y	Name for the workforce compensation plan item
CMP_IV_PERSON_ID	Number	Y	Unique numeric identifier for the individual associated with the workforce compensation plan
CMP_IV_PLAN_START_DATE	Date	Y	Date when the workforce compensation plan becomes active
CMP_IV_PLAN_END_DATE	Date	Y	Date when the workforce compensation plan becomes inactive
CMP_IV_PLAN_ELIG_DATE	Date	Y	Date when the workforce compensation plan becomes eligible
CMP_IV_PERFORMANCE_EFF_DATE	Date	Y	Date to use for compensation performance ratings
CMP_IV_PROMOTION_EFF_DATE	Date	Y	Date on which job, grade, and position changes take effect
CMP_IV_XCHG_RATE_DATE	Date	Y	Date when the <b>Start Workforce Compensation Cycle and Refresh Workforce Compensation Data</b> processes obtain conversion rates from the general ledger daily rates table.
CMP_IV_ASSIGNMENT_ID	Number	Y	Date to use for assignments

These are the return variables available to this type of formula:

Return Value	Data Type	Required	Description
L_DEFAULT_VALUE	Number/Char/Date	Y	Default value from the formula. The date should be in yyyy/mm/dd format
L_DATA_TYPE	Char	Y	Data type of the column

This sample formula determines the value of a column based on its item name.

```
*****
FORMULA NAME : Compensation Default and Override Formula
FORMULA TYPE : Compensation Default and Override
DESCRIPTION : Defaults the value of a column based on its item_name
*****
```

```
/*===== INPUT VALUES DEFAULTS BEGIN =====*/
INPUTS ARE CMP_IV_PLAN_ID (number), CMP_IV_PERIOD_ID (number), CMP_IV_COMPONENT_ID (number),
      CMP_IV_ITEM_NAME (text)
/*===== INPUT VALUES DEFAULTS ENDS=====*/
/*===== FORMULA SECTION BEGIN =====*/
DEFAULT FOR CMP_IV_ITEM_NAME IS 'YYYYYYYY'
L_DEFAULT_VALUE = to_char(0)
IF (CMP_IV_ITEM_NAME = 'AmountComp1') THEN
(
L_DEFAULT_VALUE = to_char(3333)
)
ELSE IF (CMP_IV_ITEM_NAME = 'AmountComp2') THEN
(
L_DEFAULT_VALUE = to_char(7777)
)
ELSE
(
L_DEFAULT_VALUE = to_char(-999)
)
RETURN L_DEFAULT_VALUE
/*===== FORMULA SECTION END =====*/
```

#### Related Topics

- [Formula Compilation Errors](#)
- [Formula Execution Errors](#)
- [When do I run the Compile Formula process?](#)
- [Default Value Properties of Worksheet Columns](#)
- [Example of Writing a Fast Formula Using Formula Text](#)

# Formula Type Is Compensation Hierarchy Determination

To build your own management or reviewer hierarchy for use in a workforce compensation plan, you can create formulas using the Compensation Hierarchy Determination formula type. You select the formula when you configure hierarchies for the workforce compensation plan.

Here's a comprehensive list of the contexts available to this type of formula:

- DATE\_EARNED
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- END\_DATE
- START\_DATE
- HR\_TERM\_ID
- JOB\_ID
- LEGISLATIVE\_DATA\_GROUP\_ID
- COMPENSATION\_RECORD\_TYPE
- ORGANIZATION\_ID
- PAYROLL\_ASSIGNMENT\_ID
- PAYROLL\_RELATIONSHIP\_ID
- PAYROLL\_TERM\_ID
- PERSON\_ID

The database items available for this type of formula are related to Person, Assignment, Salary, Element Entries, Compensation Record, and From and End Dates.

These are the input variables available to this type of formula:

Input Value	Data Type	Required	Description
CMP_IV_ASSIGNMENT_ID	Number	Y	Assignment ID
CMP_IV_PLAN_ID	Number	Y	Plan ID
CMP_IV_PERIOD_ID	Number	Y	Period ID
CMP_IV_COMPONENT_ID	Number	Y	Component ID
CMP_IV_PERSON_ID	Number	Y	Worker ID
CMP_IV_PLAN_START_DATE	Date	Y	Plan Start Date
CMP_IV_PLAN_END_DATE	Date	Y	Plan End Date
CMP_IV_PLAN_EXTRACTION_DATE	Date	Y	Plan Extraction Date

Input Value	Data Type	Required	Description
CMP_IV_PLAN_ELIG_DATE	Date	Y	Plan Eligibility Date
CMP_IV_PERFORMANCE_EFF_DATE	Date	Y	Performance Effective Date
CMP_IV_PROMOTION_EFF_DATE	Date	Y	Promotion Effective Date
CMP_IV_XCHG_RATE_DATE	Date	Y	Currency Conversion Date

These are the return variables available to this type of formula:

Return Value	Data Type	Required	Description
L_PERSON_ID	Number	Y	Person ID of manager
L_ASSIGNMENT_ID	Number	Y	Assignment ID of manager

Or

Return Value	Data Type	Required	Description
L_PERSON_NUMBER	Number	Y	Person number of manager

The **Start Workforce Compensation Cycle** and **Refresh Workforce Compensation Data** processes show this error when they can't find the assignment ID:

- Formula passed in an invalid person number <15465857>. Assignment ID couldn't be obtained.

This sample formula determines the manager of a person when the assignment\_id is passed.

```

*****
FORMULA NAME : Compensation Hierarchy Determination Formula
FORMULA TYPE : Compensation Hierarchy Determination
DESCRIPTION: Hierarchy determination fast formula which is based on assignment_id
*****



/*===== INPUT VALUES DEFAULTS BEGIN =====*/
INPUTS ARE CMP_IV_ASSIGNMENT_ID (number), CMP_IV_PLAN_ID (number), CMP_IV_PERIOD_ID (number)/*=====
INPUT VALUES DEFAULTS ENDS=====*/
/*===== FORMULA SECTION BEGIN =====*/
DEFAULT FOR CMP_IV_ASSIGNMENT_ID IS 0
L_PERSON_ID = '0' L_ASSIGNMENT_ID = '0'
if (CMP_IV_ASSIGNMENT_ID = 100000008154060 ) THEN
(
L_PERSON_ID = to_char(-999) // -999 indicates top level
//Manager.
L_ASSIGNMENT_ID = to_char(-999)
)
ELSE
(
L_PERSON_ID = to_char(100000008153756)
L_ASSIGNMENT_ID = to_char(100000008154060)
)

RETURN L_PERSON_ID , L_ASSIGNMENT_ID

```

```
/*===== FORMULA SECTION END =====*/
```

#### Related Topics

- [Formula Compilation Errors](#)
- [Formula Execution Errors](#)
- [When do I run the Compile Formula process?](#)
- [Options to Configure Workforce Compensation Plan Hierarchies](#)
- [Example of Writing a Fast Formula Using Formula Text](#)

## Formula Type Is Compensation Person Selection

To determine the person selected for an associated workforce compensation plan, you can create formulas using the Compensation Person Selection formula type. You select the formula when you run these batch processes:

- Start Workforce Compensation Cycle
- Refresh Workforce Compensation Data
- Transfer Workforce Compensation Data to HR

Here's a comprehensive list of the contexts available to this type of formula:

- DATE\_EARNED
- EFFECTIVE\_DATE
- END\_DATE
- START\_DATE
- HR\_ASSIGNMENT\_ID
- HR\_TERM\_ID
- JOB\_ID
- LEGISLATIVE\_DATA\_GROUP\_ID
- COMPENSATION\_RECORD\_TYPE
- ORGANIZATION\_ID
- PAYROLL\_ASSIGNMENT\_ID
- PAYROLL\_RELATIONSHIP\_ID
- PAYROLL\_TERM\_ID
- PERSON\_ID

The database items available for this type of formula are related to Person, Assignment, Salary, Element Entries, Compensation Record, and From and End Dates.

These are the input variables available to this type of formula:

Input Value	Data Type	Required	Description
CMP_IV_PLAN_ID	Number	Y	Plan ID

Input Value	Data Type	Required	Description
CMP_IV_PERIOD_ID	Number	Y	Period ID
CMP_IV_PLAN_START_DATE	Date	Y	Plan Start Date
CMP_IV_PLAN_END_DATE	Date	Y	Plan End Date
CMP_IV_PLAN_ELIG_DATE	Date	Y	Plan Eligibility Date
CMP_IV_PERFORMANCE_EFF_DATE	Date	Y	Performance Effective Date
CMP_IV_PROMOTION_EFF_DATE	Date	Y	Promotion Effective Date
CMP_IV_XCHG_RATE_DATE	Date	Y	Currency Conversion Date
CMP_IV_ASSIGNMENT_ID	Number	Y	Assignment ID
CMP_IV_PERSON_ID	Number	Y	Worker ID

These are the return variables available to this type of formula:

Return Value	Data Type	Required	Description
L_SELECTED	Char	N	Y or N

This sample formula determines if a person is selected for a workforce compensation plan based on their assignment\_id.

```
*****
FORMULA NAME : Compensation Selection Formula
FORMULA TYPE : Compensation Person Selection
DESCRIPTION: Assignment_id based selection fast formula
*****  

/*===== INPUT VALUES DEFAULTS BEGIN =====*/
INPUTS ARE CMP_IV_ASSIGNMENT_ID (number), CMP_IV_PLAN_ID (number)
/*===== INPUT VALUES DEFAULTS ENDS=====*/
  

/*===== FORMULA SECTION BEGIN =====*/
DEFAULT FOR CMP_IV_ASSIGNMENT_ID IS 0
l_selected = 'Y'
/* 100000008154095 - Ariel.Aimar@oracle.com - GBI data*/
if (CMP_IV_ASSIGNMENT_ID = 100000008154095) THEN
(
l_selected = 'N'
)
else
(
l_selected = 'Y'
)
RETURN l_selected
/*===== FORMULA SECTION END =====*/
```

*Related Topics*

- [Formula Compilation Errors](#)
- [Formula Execution Errors](#)
- [When do I run the Compile Formula process?](#)
- [Example of Writing a Fast Formula Using Formula Text](#)

## Formula Type for Total Compensation Item

The Total Compensation Item formula determines compensation information that isn't stored in the other predefined item source types.

You select the formula when you manage compensation items on the Create or Edit Compensation Items page. Click the Validate button to validate your expectations for compensation items before you generate the statements.

The following contexts are available to formulas of this type:

- DATE\_EARNED
- EFFECTIVE\_DATE
- END\_DATE
- START\_DATE
- HR\_ASSIGNMENT\_ID
- HR\_TERM\_ID
- JOB\_ID
- LEGISLATIVE\_DATA\_GROUP\_ID
- COMPENSATION\_RECORD\_TYPE
- ORGANIZATION\_ID
- PAYROLL\_ASSIGNMENT\_ID
- PAYROLL\_RELATIONSHIP\_ID
- PAYROLL\_TERM\_ID
- PERSON\_ID

Database items related to Person, Assignment, Salary, Element Entries, Compensation Record, and From and End Dates are available to formulas of this type.

The following input variables are available to formula of this type.

Input Value	Data Type	Required	Description
CMP_IV_PERIOD_ID	Char	Y	Period ID
CMP_IV_PERIOD_START_DATE	Date	Y	Statement Period Start Date
CMP_IV_PERIOD_END_DATE	Date	Y	Statement Period End Date

Input Value	Data Type	Required	Description

The following return variables are available to formula of this type.

Return Value	Data Type	Required	Description
COMPENSATION_DATES	Date	Y	One to 15 transaction dates delimited by semicolon, maximum 250 characters.
VALUES	Char	Y	One to 15 transaction values delimited by semicolon, maximum 250 characters. Must be the same number of values as dates.
ASSIGNMENTS	Char	N	One to 15 transaction assignments delimited by semicolon, maximum 250 characters. Must be the same number of assignments as dates. Can return an empty space with a delimiter ( ; ;).
LEGALEMPLOYERS	Char	N	One to 15 legal employer IDs delimited by semicolon, maximum 250 characters. Must be the same number of assignments as dates. Can return an empty space with a delimiter ( ; ;).
UNIT	Char	N	4 optional variables for every value and every variable can have more than one value delimited by “;”. Can return currency code from monetary units.
COMPENSATION_DATES1	Date	Y	Second variable for transaction dates from 16 to 30 if limit of 250 characters is exceeded.
VALUES1	Char	Y	Second variable for transaction values from 16 to 30 if limit of 250 characters is exceeded.
ASSIGNMENTS1	Char	N	Second variable for transaction assignments from 16 to 30 if limit of 250 characters is exceeded.
LEGALEMPLOYERS1	Char	N	Second variable for legal employer IDs from 16 to 30 if limit of 250 characters is exceeded.

Return Value	Data Type	Required	Description
UNIT1	Char	N	4 optional variables for every value and every variable can have more than one value delimited by ;. Can return currency code from monetary units.
COMPENSATION_DATES2	Date	Y	Transaction dates from 31 to 45.
VALUES2	Char	Y	Transaction values from 31 to 45.
ASSIGNMENTS2	Char	N	Transaction assignments from 31 to 45.
LEGALEMPLOYERS2	Char	N	Legal employers from 31 to 45.
UNIT2	Char	N	4 optional variables for every value and every variable can have more than one value delimited by ;. Can return currency code from monetary units.
COMPENSATION_DATES3	Dates	Y	Transaction dates from 46 to 60.
VALUES3	Char	Y	Transaction values from 46 to 60.
ASSIGNMENTS3	Char	N	Transaction assignments from 46 to 60.
LEGALEMPLOYERS3	Char	N	Legal employers from 46 to 60.
UNIT3	Char	N	4 optional variables for every value and every variable can have more than one value delimited by ;. Can return currency code from monetary units.

This sample formula returns one date and one value based on the worker ID.

```
*****
FORMULA NAME : Total Compensation Simple Item Formula
FORMULA TYPE : Total Compensation Item
DESCRIPTION : Returns one date and one value.
*****
```

```
===== INPUT VALUES DEFAULTS BEGIN =====
INPUTS ARE CMP_IV_PERSON_ID (text), CMP_IV_PERIOD_START_DATE (date), CMP_IV_PERIOD_END_DATE (date)
DEFAULT FOR CMP_IV_PERSON_ID IS '-1'
DEFAULT FOR CMP_IV_PERIOD_START_DATE IS '4712/12/31' (date)
DEFAULT FOR CMP_IV_PERIOD_END_DATE IS '4712/12/31' (date)
===== INPUT VALUES DEFAULTS ENDS =====
```

```
===== FORMULA SECTION BEGIN =====
COMPENSATION_DATES = '2005/01/01'
VALUES = '500.00'
```

```

ASSIGNMENTS = to_char(get_context(HR_ASSIGNMENT_ID,-1))

RETURN COMPENSATION_DATES, VALUES, ASSIGNMENTS

/*===== FORMULA SECTION END =====*/

```

This sample formula returns multiple variables.

```

***** FORMULA NAME : Total Compensation Multi Item Formula
***** FORMULA TYPE : Total Compensation Item
***** DESCRIPTION : Returns multiple variables.
***** INPUT VALUES DEFAULTS BEGIN *****
INPUTS ARE CMP_IV_PERSON_ID (text), CMP_IV_PERIOD_START_DATE (date) , CMP_IV_PERIOD_END_DATE (date)
***** INPUT VALUES DEFAULTS ENDS *****
***** FORMULA SECTION BEGIN *****
COMPENSATION_DATES = '2009/01/01;2009/02/01;2009/03/01'
COMPENSATION_DATES1 = '2009/07/01;2009/08/01;2009/09/01'
COMPENSATION_DATES2 = '2009/10/01;2009/11/01;2009/12/01'
COMPENSATION_DATES3 = '2009/10/01;2009/11/01;2009/12/01'

VALUES = '200.00;200.00;300.00'
VALUES1 = '300.00;500.00;500.00'
VALUES2 = '500.00;500.00;600.00'
VALUES3 = '600.00;600.00;700.00'

/* Returns only first two assignment */
ASSIGNMENTS = '1234567890;1234567890'
ASSIGNMENTS1 = '1234567890;1234567890;1234567890'

/* Returns last two assignments */
ASSIGNMENTS2 = '1234567890;1234567890'
/* Returns first and last assignments */
ASSIGNMENTS3 = '1234567890;;1234567890'

LEGALEMPLOYERS = '0123456789;;0123456789'
LEGALEMPLOYERS1 = '0123456789;0123456789;0123456789'
LEGALEMPLOYERS2 = '0123456789;0123456789;0123456789'
LEGALEMPLOYERS3 = '0123456789;0123456789'

UNIT = 'USD'
UNIT1 = 'EUR'
UNIT2 = 'AUD'
UNIT3 = 'CAD'
RETURN
  COMPENSATION_DATES,VALUES,COMPENSATION_DATES1,VALUES1,COMPENSATION_DATES2,VALUES2,COMPENSATION_DATES3,VALUES3,ASSIG
GALEMPLOYERS,LEGALEMPLOYERS1,LEGALEMPLOYERS2,LEGALEMPLOYERS3,UNIT,UNIT1,UNIT2,UNIT3

/*===== FORMULA SECTION END =====*/

```

### Related Topics

- [Formula Compilation Errors](#)
- [Formula Execution Errors](#)
- [When do I run the Compile Formula process?](#)
- [Example of Writing a Fast Formula Using Formula Text](#)

# Using Profiles DBI in Fast Formula

Profiles DBI can be reported using HRT\_PERSON\_GENERIC\_CONTENT\_TYPE\_UE.

You can access the DBI associated with this UE only in the formula type: Generic Formula for Profile Content (HRT\_PROFILE\_GENERIC\_FORMULA\_TYPE)

Set these Contexts to create a fast formula using the HRT\_PERSON\_GENERIC\_CONTENT\_TYPE\_\*\* DBIs.

- EFFECTIVE\_DATE
- PERSON\_ID
- TALENT\_PROFILE\_CONTENT\_TYPE

Any another formula type which also sets these three Contexts will be able to check the DBI items from the above UE. However among the seeded Formula Types only the Generic Formula for Profile Content requires these 3 contexts. Hence currently we can only use the this Formula Type for formulas based on the Profiles DBIs (HRT\_PERSON\_GENERIC\_CONTENT\_TYPE\_\*\* , e.g. HRT\_PERSON\_GENERIC\_CONTENT\_TYPE\_ITEM\_TEXT240\_1).

In case you require these DBIs in other formula types, call the Profile formula from within your own Formula (it can be another type e.g. Compensation Default and Override). However you need to set all the required CONTEXTS.

**Note:** If you are using Enhanced Talent Profiles, when you use the context TALENT\_PROFILE\_CONTENT\_TYPE, you will see data for all the sections that are associated with your specific content type or template. You must loop through the data retrieved and use the attribute SECTION\_CONTEXT to find the specific profile section data you are looking for.

Use Profiles DBI in Fast Formula by following these steps:

1. Create a new Fast Formula of the formula type "Generic Formula type for Profile Content".
2. Set the context values: Person Id, Effective Date and TALENT\_PROFILE\_CONTENT\_TYPE. Include the DBI values customer to be used in Compensation Fast Formula in the return statement like HRT\_PERSON\_GENERIC\_CONTENT\_TYPE\_ITEM\_TEXT240\_1 etc...
3. Use the above created Fast Formula in Compensation Formula "Default and Override". Set the context values (Person Id, Effective Date and TALENT\_PROFILE\_CONTENT\_TYPE) for the above Profiles Fast Formula created in Step1 before calling the Profiles Fast Formula.
4. Reference the Profiles Fast Formula from Compensation Fast Formula to get the values present in the Return statement of Profiles Fast Formula.
5. Use these Return values in the Compensation Fast Formula.

## Example 1: Simple implementation to return the number of Competency items

1. Formula compiled for "Compensation default and override" type:

Sample Formula name: "Sr Test Two"

Sample Formula text:

```
/*-----*/  
L_DATA_TYPE = 'TEXT'
```

```

L_DEFAULT_VALUE = '111'
L_PERSON_ID = get_context(person_id, -1)
IF (IS_EXECUTABLE('SR_FORMULA_GENERIC_TWO')) THEN
(
SET_INPUT('PERSON_ID', L_PERSON_ID)
SET_INPUT('TALENT_PROFILE_CONTENT_TYPE', 'COMPETENCY')
EXECUTE('SR_FORMULA_GENERIC_TWO')
)
ELSE(
L_DATA = ESS_LOG_WRITE('SR_FORMULA_GENERIC_TWO is not executable')
)
L_DEFAULT_VALUE = GET_OUTPUT('L_RETURN_VALUE_FROM_PROFILE', 'NULL')
RETURN L_DATA_TYPE, L_DEFAULT_VALUE
/*-----*/

```

2. Formula compiled for "Generic Formula type for Profile Content" type:

Sample Formula name: "SR\_FORMULA\_GENERIC\_TWO"

Sample Formula text:

```

/*-----*/
DEFAULT_DATA_VALUE FOR HRT_PERSON_GENERIC_CONTENT_TYPE_ITEM_TEXT240_1 IS 'NULL'
DEFAULT_DATA_VALUE FOR HRT_PERSON_GENERIC_CONTENT_TYPE_ITEM_TEXT240_2 IS 'NULL'
I=1
L_RETURN_VALUE_FROM_PROFILE = '111'
WHILE HRT_PERSON_GENERIC_CONTENT_TYPE_ITEM_TEXT240_1.EXISTS(I) LOOP
(
I= I+1
)
L_RETURN_VALUE_FROM_PROFILE = TO_CHAR(I)
RETURN L_RETURN_VALUE_FROM_PROFILE
/*-----*/

```

## Example 2: Implementation to return the first Item\_Text\_240\_1

Sample Formula name: "TPC\_Override"

Sample Formula text:

```

/*-----*/
/* TPC_Override formula */
L_VALUE = 'NULL'
SET_INPUT('TALENT_PROFILE_CONTENT_TYPE', 'COMPETENCY')
L_PERSON_ID = get_context(person_id, -1)
SET_INPUT('PERSON_ID', L_PERSON_ID)
EXECUTE('TPC_Profile')
L_VALUE = GET_OUTPUT('ITEM2401', 'NA')
L_DATA = ESS_LOG_WRITE('TPC executed '||L_VALUE)
RETURN L_VALUE
/*-----*/

```

2. Formula compiled for "Generic Formula type for Profile Content" type:

Sample Formula name: TPC\_Profile

Sample Formula text:

```

/*-----*/
/*TPC_Profile formula */ -- Generic Profile Content
/* DATABASE ITEM DEFAULTS BEGIN */
DEFAULT_DATA_VALUE FOR HRT_PERSON_GENERIC_CONTENT_TYPE_ITEM_TEXT240_1 IS 'NA'
/* DATABASE ITEM DEFAULTS END */
I = HRT_PERSON_GENERIC_CONTENT_TYPE_ITEM_TEXT240_1.FIRST(-1)
ITEM2401 = HRT_PERSON_GENERIC_CONTENT_TYPE_ITEM_TEXT240_1[I]

```

```
return ITEM2401
/*-----*/
```

# 8 Formulas for Benefits

## Overview of Benefits Formulas

Use formulas to configure your plan design to the requirements of your enterprise.

They provide a flexible alternative to delivered business rules for such purposes as:

- Date calculations, such as:
  - Enrollment start and end dates
  - Rate or coverage start and end dates
  - Waiting periods and enrollment periods
  - Action item due dates
- Calculations of rate and coverage amount, minimum and maximum, or upper and lower limits
- Certification requirements
- Partial month and proration calculations
- Eligibility and participation evaluation

For example, you can write a formula to calculate benefits eligibility for those cases where the provided eligibility criteria don't accommodate your particular requirements.

A formula can return more than one value to a calling package. In Benefits, different packages which call the formula expect outputs to be returned differently. Based on the formula type, the number and data types of values to be returned are predefined. Some packages expect the predefined names for output variables; the order of a variable in the return statement does not matter. Some packages expect the return variables to be returned in a certain order; in this case the name of output variable does not matter. If predefined names are expected to be used and the formula does not use the expected names, the process would error out at run time.

### Related Topics

- [Guidelines to Test Benefits Formulas](#)

## Person Change Causes Life Event

This formula type can be used to determine whether to trigger a person life event or not.

### Contexts

The following contexts are available to formulas of this type:

- HR\_RELATIONSHIP\_ID
- HR\_TERM\_ID
- PAYROLL\_RELATIONSHIP\_ID

- PAYROLL\_TERM\_ID
- LEGAL\_EMPLOYER\_ID
- DATE\_EARNED
- HR\_ASSIGNMENT\_ID
- BUSINESS\_GROUP\_ID
- PERSON\_ID
- JOB\_ID
- EFFECTIVE\_DATE
- PAYROLL\_ASSIGNMENT\_ID
- LEGISLATIVE\_DATA\_GROUP\_ID
- ORGANIZATION\_ID
- BENEFIT\_RELATION\_ID

## Database Items

Use only the following database items that are available to formulas of this type:

- All columns except attribute columns from the following tables:
  - BEN\_CVG\_AMT\_CALC\_MTHD\_F
  - BEN\_LER\_F
  - BEN\_OIPL\_F
  - BEN\_OPT\_F
  - BEN\_PGM\_F
  - BEN\_PLIP\_F
  - BEN\_PL\_F
  - BEN\_PL\_TYP\_F
  - BEN\_PTIP\_F
  - BEN\_YR\_PERD
- Database items based on person's assignment – HR\_ASSIGNMENT\_ID AND PERSON\_ID.

## Input Variables

Input variables are available to formulas of this type. Refer to Appendix 2 for list of input value.

## Return Variables

Use predefined names for return variables. The following return variables are available to formulas of this type.

### Return variables table

Return Value	Data Type	Required	Description
L_RETURN	Char	N	Returns Y or N

## Errors

If any other output value is returned then the participation process errors with BEN\_91329\_FORMULA\_RETURN.

### Sample Formula 1:

FORMULA NAME: BEN\_TRIGGER\_ASG\_EVENT FORMULA

TYPE: Person Changes Causes Life Event

DESCRIPTION: When a person's assignment type changes from one specific value to another, the event should get triggered.

```
/*===== INPUT VALUES DEFAULTS BEGIN =====*/
Inputs are BEN_ASG_IN_ASSIGNMENT_TYPE (text), BEN_ASG_IO_ASSIGNMENT_TYPE (text)
DEFAULT for BEN_ASG_IN_ASSIGNMENT_TYPE is 'xyz'
DEFAULT for BEN_ASG_IO_ASSIGNMENT_TYPE is 'abc'
/*===== INPUT VALUES DEFAULTS ENDS=====*/
/*===== FORMULA SECTION BEGIN =====*/ l_return = 'N'
IF ( BEN_ASG_IN_ASSIGNMENT_TYPE = 'A'
and BEN_ASG_IO_ASSIGNMENT_TYPE= 'B' )
then
(l_return = 'Y')
else
(l_return = 'N')
return l_return
/*===== FORMULA SECTION END =====*/
```

### Sample Formula 2:

FORMULA NAME: BEN\_ADDRESS\_CHG\_EVENT

FORMULA TYPE: Person Changes Causes Life Event

DESCRIPTION: Life event should trigger only for persons based in US or CA.

**Note:** Needs PERSON\_ID context to work.

```
Default for PER_PER_ADD_COUNTRY is 'AA'
l_country = PER_PER_ADD_COUNTRY
l_return = 'N'
IF (l_country = 'US' or l_country = 'CA')
THEN
(l_return = 'Y')
return l_return
```

### Sample Formula 3:

FORMULA NAME: TERM\_YES\_NO

DESCRIPTION: Formula to determine Termination life event

```
/*===== DATABASE ITEM DEFAULTS BEGIN =====*/
DEFAULT for PER_ASG_ACTION_CODE is ''
/*===== DATABASE ITEM DEFAULTS ENDS=====*/
/*===== FORMULA SECTION BEGIN =====*/
l_action_code = PER_ASG_ACTION_CODE
l_create_ptnl = 'N'
if NOT PER_ASG_ACTION_CODE was defaulted then (
if (l_action_code='RESIGNATION' or l_action_code='TERMINATION') then (l_create_ptnl = 'Y')
else
(l_create_ptnl = 'N')
)
return l_create_ptnl
```

## Sample Formula 4:

FORMULA NAME : BEN\_SAL\_CHG\_TRIG\_FF

FORMULA TYPE : Person Changes Causes Life event

DESCRIPTION : Formula to determine when to trigger SAL CHANGE Life event.

```
inputs are BEN_SAL_IN_SALARY_AMOUNT(text), BEN_SAL_IO_SALARY_AMOUNT(text)
default for BEN_SAL_IN_SALARY_AMOUNT is '99999999' default for BEN_SAL_IO_SALARY_AMOUNT is '99999999'
l_create_ptnl = 'N'
If (BEN_SAL_IO_SALARY_AMOUNT = '99999999' or
BEN_SAL_IN_SALARY_AMOUNT = BEN_SAL_IO_SALARY_AMOUNT )
THEN
( l_create_ptnl = 'N' )
ELSE
( l_create_ptnl = 'Y' )
return l_create_ptnl
```

## Sample Formula 5:

```
*****
FORMULA NAME : Legal Employer Transfer or New Hire
FORMULA TYPE : Person Changes Causes Life Event
*****
INPUTS are BEN_ASG_IN_ACTION_CODE(text), BEN_ASG_IO_ACTION_CODE(text)
DEFAULT for PER_ASG_ACTION_CODE is 'AA'
l_action_code='XX'
l_old_action_code='YY'
l_action_code = BEN_ASG_IN_ACTION_CODE
l_old_action_code = BEN_ASG_IO_ACTION_CODE
l_return = 'N'
if (l_action_code<>'XX') then
(
if ((l_action_code = 'HIRE' or l_action_code= 'GLB_TRANSFER') and
l_action_code <> l_old_action_code ) then
(
l_return = 'Y'
)
else
(
l_return = 'N'
)
)
return l_return
```

## Sample Formula 6:

```
*****
FORMULA NAME: BEN_POS_CHG_TRIGGER_FF
```

```

FORMULA TYPE: Person Change Causes Life Event
DESCRIPTION: FF to trigger 'Position Change' life event.
*****
INPUTS ARE NEW_VAL (text), OLD_VAL (text), COLUMN (text), PK_ID (text), BEN_ASG_IO_HOURLY_SALARIED_CODE
(text), BEN_ASG_IN_HOURLY_SALARIED_CODE (text)
DEFAULT FOR OLD_VAL IS 'NONE'
DEFAULT FOR NEW_VAL IS 'NONE'
DEFAULT FOR BEN_ASG_IN_HOURLY_SALARIED_CODE IS 'NONE'
DEFAULT FOR BEN_ASG_IO_HOURLY_SALARIED_CODE IS 'NONE'
DEFAULT FOR PER_ASG_ACTION_CODE IS 'NONE'
DEFAULT for PER_ASG_HOURLY_SALARIED_CODE is 'NONE'
l_hourly_code = PER_ASG_HOURLY_SALARIED_CODE
l_action_code = PER_ASG_ACTION_CODE
l_return = 'N'
/* Check if Action is 'Position Change' OR 'Promotion' */
IF (((l_hourly_code = 'H') AND ((l_action_code = 'POSITION_CHANGE') OR (l_action_code = 'PROMOTION'))) AND
(OLD_VAL <> NEW_VAL))
OR ((l_action_code = 'PROMOTION') AND ((BEN_ASG_IO_HOURLY_SALARIED_CODE = 'H') AND
(BEN_ASG_IN_HOURLY_SALARIED_CODE = 'S'))))
THEN
(
l_return = 'Y'
)
RETURN l_return

```

## Sample Formula 7:

```

*****
FORMULA TYPE: Person Change Causes Life Event
*****
INPUTS ARE NEW_VAL (text), OLD_VAL (text), COLUMN (text), PK_ID (text)
DEFAULT FOR OLD_VAL IS 'NONE'
DEFAULT FOR NEW_VAL IS 'NONE'
DEFAULT FOR PER_ASG_ACTION_CODE IS 'NONE'
DEFAULT for PER_ASG_HOURLY_SALARIED_CODE is 'NONE'
l_debug = 'Y'
l_hourly_code = PER_ASG_HOURLY_SALARIED_CODE
l_action_code = PER_ASG_ACTION_CODE
l_return = 'N'
IF ((l_hourly_code = 'H') AND (l_action_code = 'WORK_HOURS_CHANGE') AND (OLD_VAL <> NEW_VAL)) THEN
(
l_return = 'Y'
)
RETURN l_return

```

## Sample Formula 8:

FORMULA NAME : New Hire

FORMULA TYPE : Person Changes Causes Life event

DESCRIPTION : Formula to check if there is any assignment change. If the action code is hire, rehire, or global transfer, it returns Yes and then triggers the life event.

```

INPUTS are NEW_VAL(text), OLD_VAL(text), COLUMN(text), PK_ID(text),
BEN_ASG_IN_ACTION_CODE(text), BEN_ASG_IO_ACTION_CODE(text), BEN_ASG_IN_ASSIGNMENT_TYPE(text)

default for NEW_VAL is 'XXX'
default for OLD_VAL is 'ZZZ'
default for BEN_ASG_IN_ACTION_CODE is 'XX'
default for BEN_ASG_IO_ACTION_CODE is 'XX'

l_create_ptnl = 'N'
lterm = 'XX'

```

```
l_old_term ='XX'

lterm = BEN_ASG_IN_ACTION_CODE
l_old_term =BEN_ASG_IO_ACTION_CODE
if (ISNULL(lterm)='N' ) then
(
lterm='XX'
)

if (ISNULL(l_old_term)='N' ) then
(
l_old_term='XX'
)

if (lterm<>'XX') then
(
if ((lterm='HIRE' or lterm='REHIRE' or lterm='GLB_TRANSFER' ) AND
(lterm <> l_old_term) and ,BEN_ASG_IN_ASSIGNMENT_TYPE='E') THEN
(l_create_ptnl = 'Y')
)
return l_create_ptnl
```

## Participation and Rate Eligibility

This formula type can be used to determine whether a person is eligible for an associated compensation object. You select the formula as criteria while defining Eligibility Profiles.

### Contexts

The following contexts are available to formulas of this type:

- BUSINESS\_GROUP\_ID ( ENTERPRISE\_ID)
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- LER\_ID
- OPT\_ID
- ORGANIZATION\_ID
- PGM\_ID
- PL\_ID
- PL\_TYP\_ID
- PERSON\_ID

### Database Items

Use only the following database items that are available to formulas of this type:

- All columns except attribute columns from tables:
  - EN\_CVG\_AMT\_CALC\_MTHD\_F

- BEN\_LER\_F
- BEN\_OIPL\_F
- BEN\_OPT\_F
- BEN\_PGM\_F
- BEN\_PLIP\_F
- BEN\_PL\_F
- BEN\_PL\_TYP\_F
- BEN\_PTIP\_F
- BEN\_YR\_PERD

- Database items based on person's assignment – HR\_ASSIGNMENT\_ID AND PERSON\_ID.

## Input Variables

Input variables are not available to formulas of this type.

## Return Variables

Use predefined names for return variables. The following return variables are available to formulas of this type.

### ***Return variables table***

Return Value	Data Type	Required	Description
ELIGIBLE	Char	N	Returns Y or N

## Errors

If any other output value is returned then the participation process errors with BEN\_91329\_FORMULA\_RETURN as shown below.

The formula BEN\_PRTT\_FTE\_ELIG returned an invalid value.

## Sample Formula 1:

FORMULA NAME: BEN\_PRTT\_FTE\_ELIG

FORMULA TYPE: Participation and Rate Eligibility

DESCRIPTION: This sample formula determines if a person is eligible for a compensation object based on the FTE on the assignment.

```
/*===== DATABASE ITEM DEFAULTS BEGIN =====*/
Default for PER_ASG_FTE is 0
/*===== DATABASE ITEM DEFAULTS ENDS=====*/
/*===== FORMULA SECTION BEGIN =====*/
1_fte = PER_ASG_FTE
ELIGIBLE = 'N'
```

```
IF l_fte = 1 THEN
  (ELIGIBLE = 'Y')
  return ELIGIBLE
/*===== FORMULA SECTION END =====*/
```

## Sample Formula 2:

FORMULA NAME: BEN\_PRTT\_ASG\_ELIG

FORMULA TYPE: Participation and Rate Eligibility

DESCRIPTION: This sample formula determines if a person is eligible for a compensation object based on his country of residence. Note: Needs PERSON\_ID context to work.

```
Default for PER_PER_ADD_COUNTRY is 'X'
l_cntry = PER_PER_ADD_COUNTRY
ELIGIBLE = 'N'
IF (l_cntry = 'US') THEN
  (ELIGIBLE = 'Y')
  return ELIGIBLE
```

## Sample Formula 3:

Formula Type: Participation and Rate Eligibility

Purpose: Return eligible if the participant has a dependent (child) whose age is less than 26 years by using DBI items (when there are custom user roles configured in the system).

```
DEFAULT_DATA_VALUE FOR PER_PER_CONT_REL_CONTACT_PERSON_ID is 0
DEFAULT_DATA_VALUE FOR PER_PER_CONT_REL_CONTACT_TYPE is 'NA'
DEFAULT FOR PER_PER_DATE_OF_BIRTH is '1951/01/01 00:00:00' (date)
l_eff_date = GET_CONTEXT(EFFECTIVE_DATE, to_date('1951/01/01 00:00:00'))
ELIGIBLE='N'
i=1
WHILE PER_PER_CONT_REL_CONTACT_PERSON_ID.EXISTS(i)
loop
(
if ( PER_PER_CONT_REL_CONTACT_TYPE[i]='C' ) then
(
CHANGE_CONTEXTS(PERSON_ID=PER_PER_CONT_REL_CONTACT_PERSON_ID[i])
(
l_dob = PER_PER_DATE_OF_BIRTH
if ( months_between(l_eff_date,l_dob) < 312 ) then
(
/* there is a child whose age is less than 26 years */
ELIGIBLE='Y'
EXIT
)
) /* end change_contexts */
) /* end contact_type = C */
i=i+1
) /* end loop on contacts */
return ELIGIBLE
```

## Sample Formula 4:

FORMULA NAME: BEN\_TOB\_USE\_DEP\_ELIG\_FF

FORMULA TYPE: Participation and Rate Eligibility

DESCRIPTION: Validating Tobacco Usage details for dependent.

DEFAULT for BEN\_PHB\_TOBACCO\_TYPE\_USAGE is 'xyz'

DEFAULT\_DATA\_VALUE for PER\_EXT\_CONT\_PER\_PERSON\_ID is 0

```
*****FORMULA SECTION BEGIN*****
ELIGIBLE= 'N'
i = 1
/*Checking for the TOBACCO_USER lookup code as 'Y' corresponding to the meaning as 'ANY'*/
while PER_EXT_CONT_PER_PERSON_ID.exists(i) loop
(
CHANGE_CONTEXTS(PERSON_ID = PER_EXT_CONT_PER_PERSON_ID[i])
(
If BEN_PHB_TOBACCO_TYPE_USAGE = 'Y' THEN
(ELIGIBLE = 'Y' )
i = i + 1
)
)
Return ELIGIBLE
```

## Rounding

This formula type can be used to round coverage or rate values.

### Contexts

The following contexts are available to formulas of this type:

- BUSINESS\_GROUP\_ID ( ENTERPRISE\_ID)
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID

### Database items

Use only the following database items that are available to formulas of this type:

- Database items based on person's assignment – HR\_ASSIGNMENT\_ID

### Input Variables

Input variables are not available to formulas of this type.

#### ***Input Variables table***

Input	Data Type	Required	Description
VALUE	NUMBER	Y	Value to be rounded

### Return Variables

Use predefined names for return variables. The following return variables are available to formulas of this type.

Return Value	Data type	Required	Description
L_OUTPUT	Char	N	Returns rounded value

## Errors

If any other output value is returned then the participation process errors with BEN\_91329\_FORMULA\_RETURN.

The formula BEN\_ROUNDING returned an invalid value.

## Sample Formula 1:

FORMULA NAME: BEN\_ROUNDING

FORMULA TYPE: Rounding

DESCRIPTION: Round up value to 2 decimal places. Ex - (123.17792,2) WOULD result in 123.18

```
INPUTS are VALUE
DEFAULT FOR VALUE IS 0
l_input_val = VALUE
l_output = ROUNDUP(l_input_val,2)
return l_output
```

## Sample Formula 2:

FORMULA NAME: BEN\_ROUNDING\_UP

FORMULA TYPE: Rounding

DESCRIPTION: Round value

```
INPUTS are VALUE
DEFAULT FOR VALUE IS 0
l_input_val = VALUE
l_output = ROUND(l_input_val,2)
return l_output
```

## Age Calculation

This formula type can be used to determine age to be used for eligibility, coverage or benefit, and rate calculations.

## Contexts

The following contexts are available to formulas of this type:

- BUSINESS\_GROUP\_ID ( ENTERPRISE\_ID)
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- ORGANIZATION\_ID

- JURISDICTION\_CODE
- PGM\_ID
- PL\_ID
- PL\_TYP\_ID
- OPT\_ID
- LER\_ID
- PERSON\_ID

## Database Items

Use only the following database items that are available to formulas of this type:

- All columns except attribute columns from tables: ben\_ler\_f, ben\_pgm\_f, ben\_pl\_f, ben\_pl\_typ\_f, ben\_opt\_f
- Database items based on person's assignment – HR\_ASSIGNMENT\_ID

## Input Variables

Input	Data Type	Required	Description
PERSON_ID	NUMBER		Person id is passed as input value

## Return Variables

Use predefined names for return variables. The following return variables are available to formulas of this type.

Return Value	Data Type	Required	Description
L_OUTPUT	NUMBER	Y	Returns Age

## Errors

If any other output value is returned then the participation process errors with BEN\_91329\_FORMULA\_RETURN.

The formula BEN\_AGE\_CALC returned an invalid value.

## Sample Formula 1:

FORMULA NAME: BEN\_AGE\_CALC

FORMULA TYPE: Age Calculation

DESCRIPTION: The formula returns the age of the person as of END OF YEAR.

```
DEFAULT for PER_PER_DATE_OF_BIRTH is '2013/01/01 12:00:00' (date)
l_output = 0
l_date = '2011/01/01 12:00:00'
```

```
l_start_date = PER_PER_DATE_OF_BIRTH
l_effective_date = to_date('2013/01/01 12:00:00')
l_age = months_between(l_effective_date, l_start_date)
l_output = l_age/12
l_output = floor(l_output)
if l_output < 30 then
  l_date = '2012/01/01 12:00:00'
else
  l_date = '2013/01/01 12:00:00'
return l_date
```

## Person Selection

This formula type can be used in most benefits batch processes to determine who should be processed by the batch process.

### Contexts

The following contexts are available to formulas of this type:

- BUSINESS\_GROUP\_ID ( ENTERPRISE\_ID)
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- PERSON\_ID
- LER\_ID

### Database Items

Use only the following database items that are available to formulas of this type:

- All columns except attribute columns from tables: ben\_ler\_f
- Database items based on person's assignment – HR\_ASSIGNMENT\_ID AND PERSON\_ID

### Input Variables

Input variables are not available to formulas of this type.

#### ***Input variables table***

Input	Data Type	Required	Description
BEN_IV_PERSON_ID	Char	N	Person id is passed as input value

### Return Variables

Use predefined names for return variables. The following return variables are available to formulas of this type.

### Return variables table

Return Value	Data Type	Required	Description
L_OUTPUT	Char	Y	Returns Y or N

## Errors

If any other output value is returned then the participation process errors with BEN\_91329\_FORMULA\_RETURN.

### Sample Formula 1:

FORMULA NAME: BEN\_PER\_SEL\_RULE

FORMULA TYPE: Person Selection

DESCRIPTION: The formula selects employees based on marital status, job, and work from home status.

```
/*===== DATABASE ITEM DEFAULTS BEGIN =====*/ DEFAULT for PER_PER_MARITAL_STATUS IS 'S'
DEFAULT for PER_ASG_JOB_NAME IS 'Temp'
DEFAULT for PER_ASG_WORK_AT_HOME IS 'N'
/*===== DATABASE ITEM DEFAULTS ENDS=====*/
/*===== FORMULA SECTION BEGIN =====*/ l_output = 'N'
l_mar_status = PER_PER_MARITAL_STATUS
l_job_name = PER_ASG_JOB_NAME
l_wrk_home = PER_ASG_WORK_AT_HOME
if ( l_mar_status = 'M'
and ( l_job_name = 'Manager' or l_job_name = 'Consultant' )
and l_wrk_home = 'Y' )
then
( l_output = 'Y' )
return l_output
/*===== FORMULA SECTION END =====*/
```

### Sample Formula 2:

FORMULA NAME: BEN\_PER\_SEL\_RULE2

FORMULA TYPE: Person Selection

DESCRIPTION: The formula selects employees based on person numbers.

```
/*===== DATABASE ITEM DEFAULTS BEGIN =====*/ DEFAULT for PER_ASG_PERSON_NUMBER IS '-1'
/*===== DATABASE ITEM DEFAULTS ENDS=====*/
/*===== FORMULA SECTION BEGIN =====*/ l_output = 'N'
l_per_number = PER_ASG_PERSON_NUMBER
if ( l_per_number = '1138' OR l_per_number = '1139' OR l_per_number = '1140' OR l_per_number = '1141' OR
l_per_number = '100023' ) then
( l_output = 'Y' )
return l_output
/*===== FORMULA SECTION END =====*/
```

### Sample Formula 3:

FORMULA NAME: CLOSE\_PER\_SELECTION\_1

FORMULA TYPE: Person Selection

**DESCRIPTION:** Formula coded assuming 30 day enrollment window and DFLT as benefit relation name. These need to be substituted with appropriate values based on customer setup.

```

l_return='N'
l_eff_date = '1951/01/01 00:00:00'(date)
l_eff_date = GET_CONTEXT(EFFECTIVE_DATE,to_date('1951/01/01 00:00:00')) l_end_date_char='NA'
l_end_date = '1951/01/01 00:00:00'(date)
l_name ='NA'
l_end_date_char = ben_fn_get_char_value(
'BEN_PER_IN_LER'
,'LF_EVT_OCRD_DT'
,
'
,
'BENEFIT_RELATION_NAME'
,'DFLT'
)
l_name = ben_fn_get_char_value(
'BEN_PER_IN_LER'
,'NAME'
,
'
,
'BENEFIT_RELATION_NAME'
,'DFLT'
)
/* conversion */
if (l_end_date_char <> 'NA' and l_end_date_char <> 'NO_DATA_FOUND') then
(
l_end_date=TO_DATE(l_end_date_char)
)
if ( l_name = 'Conversion' or l_name = 'Admin Maintenance')
then
(
if (l_eff_date > add_days(l_end_date,30)) then
(l_return='Y')
else
(l_return='N')
)
if ( l_name = 'Dependent Age Out' or l_name = 'Change in Day Care')
then
(
if (l_eff_date >= l_end_date) then
(l_return='Y')
else
(l_return='N')
)
return l_return

```

## Benefits Extract Custom Data Rule

You can use this formula type to insert 3 extra fields to the delivered output file.

### Contexts

The following contexts are available to formulas of this type:

- HR\_RELATIONSHIP\_ID
- HR\_TERM\_ID
- LC\_DATE\_FROM

- LC\_DATE\_TO
- PAYROLL\_RELATIONSHIP\_ID
- PAYROLL\_TERM\_ID
- LEGAL\_EMPLOYER\_ID
- DATE\_EARNED
- HR\_ASSIGNMENT\_ID BUSINESS\_GROUP\_ID
- PERSON\_ID JOB\_ID
- EFFECTIVE\_DATE PAYROLL\_ASSIGNMENT\_ID
- PAYROLL\_ID
- LEGISLATIVE\_DATA\_GROUP\_ID
- LER\_ID
- OPT\_ID
- ORGANIZATION\_ID ACTY\_BASE\_RT\_ID
- PGM\_ID
- PL\_ID
- PL\_TYP\_ID
- BENEFIT\_RELATION\_ID
- PER\_IN\_LER\_ID

## Database Items

There are no specific DBIs created for this formula type. All contexts that are available for the Participation and Rate Eligibility formula type are available for this extract formula type too.

## Input Variables

Input variables aren't available to formulas of this type.

## Return Variables

This formula can return a maximum of three values.

## Errors

If the formula returns any other output value, then the participation process errors with BEN\_91329\_FORMULA\_RETURN.

## Life Event Reason Timeliness

This formula type can be used to determine whether a potential life event should be processed or not.

## Contexts

The following contexts are available to formulas of this type:

- BUSINESS\_GROUP\_ID ( ENTERPRISE\_ID)
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- PERSON\_ID
- LER\_ID

## Database Items

Use only the following database items that are available to formulas of this type:

- All columns except attribute columns from tables: ben\_ler\_f
- Database items based on person's assignment – HR\_ASSIGNMENT\_ID AND PERSON\_ID.

## Input Variables

Input variables are not available to formulas of this type.

## Return Variables

Use predefined names for return variables. The following return variables are available to formulas of this type.

### ***Return variables table***

Return Value	Data Type	Required	Description
L_LIFEVENT_VOIDED	Char	Y	Returns Y or N

## Errors

If any other output value is returned then the participation process errors with BEN\_91329\_FORMULA\_RETURN.

## Sample Formula 1:

FORMULA NAME: BEN\_LF\_EVT\_TMLNS

FORMULA TYPE: Life Event Timeliness

DESCRIPTION: The formula voids specific events where notified date is more than 30 days after event occurred date.

```

DEFAULT for BEN_LER_NAME is 'xyz'
DEFAULT for BEN_PIL_LF_EVT_OCRD_DT is '01-JAN-1951' (date)
DEFAULT for BEN_PIL_NTFN_DT is '01-JAN-1951' (date)
l_event_name = BEN_LER_NAME
L_LIFEVENT_VOIDED = 'N'
if ( l_event_name = 'Age Changed' or l_event_name = 'Address Changed' ) and
( add_days(BEN_PIL_LF_EVT_OCRD_DT,30) < BEN_PIL_NTFN_DT ) then
(L_LIFEVENT_VOIDED = 'Y' )

```

```
return L_LIFEVENT_VOIDED
```

## Sample Formula 2:

FORMULA NAME: BEN\_LF\_EVT\_TMLNS2

FORMULA TYPE: Life Event Timeliness

DESCRIPTION: The formula voids all events where notified date is more than 90 days after event occurred date. This is to avoid back out of future dated processed events.

```
DEFAULT for BEN_PPL_LF_EVT_OCRD_DT is '1951/01/01 00:00:00' (date)
DEFAULT for BEN_PPL_NTFN_DT is '1951/01/01 00:00:00' (date)
L_LIFEVENT_VOIDED = 'N'
if ( add_days(BEN_PPL_LF_EVT_OCRD_DT,90) < BEN_PPL_NTFN_DT )
then
(L_LIFEVENT_VOIDED = 'Y' )
return L_LIFEVENT_VOIDED
```

## Coverage Amount Calculation

This formula type can be used to calculate coverage for a person.

### Contexts

The following contexts are available to formulas of this type:

- BUSINESS\_GROUP\_ID ( ENTERPRISE\_ID)
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- PERSON\_ID
- LER\_ID
- ORGANIZATION\_ID
- JURISDICTION\_CODE
- PGM\_ID
- PL\_ID
- PL\_TYP\_ID
- OPT\_ID
- LER\_ID

### Database Items

Use only the following database items that are available to formulas of this type:

- All columns except attribute columns from tables: ben\_ler\_f, ben\_pgm\_f, ben\_pl\_f, ben\_opt\_f, ben\_pl\_typ\_f
- Database items based on person's assignment – HR\_ASSIGNMENT\_ID AND PERSON\_ID

## Input Variables

Input variables are not available to formulas of this type.

## Return Variables

Use predefined names for return variables. The following return variables are available to formulas of this type.

### ***Return variables table***

Return Value	Data Type	Required	Description
L_CVG	Number	Y	Coverage value

## Errors

If any other output value is returned then the participation process errors with BEN\_91329\_FORMULA\_RETURN.

### Sample Formula 1:

FORMULA NAME: BEN\_CVG\_CALC

FORMULA TYPE: Coverage Amount Calculation

DESCRIPTION: Coverage amount is determined based on persons DOB.

```

/ DEFAULT for PER_PER_DATE_OF_BIRTH is '01-Jan-1951' (date)
1_start_date = PER_PER_DATE_OF_BIRTH
1_cvg = 0
if ( 1_start_date >= TO_DATE('01-Jan-1961') and 1_start_date <= TO_DATE('31-Dec-1974') ) then (1_cvg = 800 )
if ( 1_start_date >= TO_DATE('01-Jan-1975') and 1_start_date <= TO_DATE('31-Dec-1989') ) then (1_cvg =
1000 )
if ( 1_start_date >= TO_DATE('01-Jan-1990') and 1_start_date <= TO_DATE('31-Dec-2010') ) then ( 1_cvg =
1500 )
return 1_cvg

```

### Sample Formula 2:

FORMULA NAME: BEN\_CVG\_CALC2

FORMULA TYPE: Coverage Amount Calculation

DESCRIPTION: Formula derives coverage as factor of person's age.

```

Default for PER_PER_DATE_OF_BIRTH is '01-JAN-1901' (date)
1_dob = PER_PER_DATE_OF_BIRTH
1_effective_date = GET_CONTEXT(EFFECTIVE_DATE,to_date('01-JAN-2012'))
if not PER_PER_DATE_OF_BIRTH was defaulted then (
1_age = trunc(ROUNDUP(months_between(1_effective_date , 1_dob)/12))
1_cvg = 1_age * 2500 )
else
( 1_cvg = 10000 )
return 1_cvg

```

## Sample Formula 3:

FORMULA TYPE: Coverage Amount Calculation

DESCRIPTION: Calculate coverage for retirees based on their retirement period.

```

DEFAULT for CMP_ASSIGNMENT_SALARY_ANNUAL_AMOUNT is 0
DEFAULT for PER_PER_LATEST_TERMINATION_DATE is '1951/01/01 00:00:00' (date)
l_sal = 0
l_yrs = 0
l_fctr = 0
l_val = 0
l_eff_date = GET_CONTEXT(EFFECTIVE_DATE, to_date('1951/01/01 0:00:00'))
IF PER_PER_LATEST_TERMINATION_DATE WAS NOT DEFAULTED
THEN
(
l_term_dt = PER_PER_LATEST_TERMINATION_DATE
l_sal_dt = add_days(l_term_dt,-1)
/* get the salary as of the term date -1 */
CHANGE_CONTEXTS (EFFECTIVE_DATE=l_sal_dt)
(
l_sal = CMP_ASSIGNMENT_SALARY_ANNUAL_AMOUNT
) /* end change contexts */
/* calculate the time passed since the employee retired */
l_yrs = Round(MONTHS_BETWEEN(l_eff_date,l_term_dt)/12)
if (l_yrs>=0 and l_yrs < 5) then
( l_fctr = 1 )
else if (l_yrs >= 5) then
( l_fctr = 0.5 )
l_val = ROUND(l_sal * l_fctr)
) /* end defaulted */
return l_val

```

## Rate Value Calculation

This formula type can be used to calculate the rate for a benefits object.

### Contexts

The following contexts are available to formulas of this type:

- BUSINESS\_GROUP\_ID ( ENTERPRISE\_ID )
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- PERSON\_ID
- LER\_ID
- ORGANIZATION\_ID
- JURISDICTION\_CODE
- PGM\_ID
- PL\_ID
- PL\_TYP\_ID
- OPT\_ID

- LER\_ID

## Database Items

Use only the following database items that are available to formulas of this type:

- All columns except attribute columns from tables: ben\_ler\_f, ben\_pgm\_f, ben\_pl\_f, ben\_opt\_f, ben\_pl\_typ\_f
- Database items based on person's assignment – HR\_ASSIGNMENT\_ID AND PERSON\_ID

## Input Variables

Input variables are not available to formulas of this type.

## Return Variables

Use predefined names for return variables. The following return variables are available to formulas of this type.

### ***Return variables table***

Return Value	Data Type	Required	Description
L_RATE	Number	Y	Return rate value

## Errors

If any other output value is returned then the participation process errors with BEN\_91329\_FORMULA\_RETURN.

## Sample Formula 1:

FORMULA NAME: BEN\_RT\_CALC

FORMULA TYPE: Rate Value Calculation

DESCRIPTION: Rate amount is determined based on number of sons & daughters, and spouse.

```
default for l_sons is 0
default for l_spouse is 0
default for l_daughters is 0
l_rate = 0
l_sons = ben_fn_get_char_value(
  'PER_CONTACT_RELSHIPS_F'
, 'SON_COUNT'
, NULL
, NULL
, 'LEGISLATION_CODE' , 'US'
)
l_daughters = ben_fn_get_char_value(
  'PER_CONTACT_RELSHIPS_F'
, 'DAUGHTER_COUNT'
, NULL
, NULL
, 'LEGISLATION_CODE' , 'US'
)
l_spouse = ben_fn_get_char_value(
  'PER_CONTACT_RELSHIPS_F'
, 'SPOUSE_COUNT'
```

```

,NULL
,NULL
, 'LEGISLATION_CODE' , 'US'
)
if ( l_spouse = 0 and l_sons= 0 and l_daughters = 0 ) then l_rate = 80 )
if (l_spouse = 1 and l_sons= 0 and l_daughters = 0) then (l_rate = 100 )
if (l_spouse = 1 and (l_sons> 0 or l_daughters > 0)) then ( l_rate = 150 )
return l_rate

```

## Sample Formula 2:

FORMULA NAME: BEN\_RT\_CALC2

FORMULA TYPE: Rate Value Calculation

DESCRIPTION: Formula derives rate based on plan design.

```

Default for BEN_PLN_NAME is 'ABC'
Default for BEN_OPT_NAME is 'ABC'
l_pl_name = BEN_PLN_NAME
l_opt_name = BEN_OPT_NAME
l_rate = 0
if ( l_pl_name = 'Aetna PPO' and l_opt_name = 'Employee Only' ) then ( l_rate = 50 )
if ( l_pl_name = 'Aetna PPO' and l_opt_name = 'Employee + Family' ) then ( l_rate = 100 )
if ( l_pl_name = 'Be Healthy HMO' and l_opt_name = 'Employee + Family' ) then ( l_rate = 110 )
if ( l_pl_name = 'Be Healthy HMO' and l_opt_name = 'Employee Only' ) then
( l_rate = 60 )
Else
( l_rate = 90 )
return l_rate

```

## Sample Formula 3:

FORMULA Type: Rate Value Calculation

Description: Calculate the rate based on the benefits balance.

```

DEFAULT_DATA_VALUE FOR BEN_PBB_BNFTS_BAL_NAME_TN IS 'XXX'
DEFAULT_DATA_VALUE FOR BEN_PBB_VAL_NN IS 0
Default for BEN_PBB_VAL is 0
l_BEN_PBB_VAL = 0
j=1
l_rate= 1062.32
WHILE (BEN_PBB_BNFTS_BAL_NAME_TN.EXISTS(j)) LOOP
(
IF (BEN_PBB_BNFTS_BAL_NAME_TN[j] = 'HEALTH_BAL') then
( l_BEN_PBB_VAL = BEN_PBB_VAL_NN[j] )
j=j+1
)
if (l_BEN_PBB_VAL = 2)
THEN
(
l_rate = l_rate + 25
)
return l_rate

```

# Dependent Eligibility

This formula type can be used to determine whether a contact is eligible to be covered as a dependent in a compensation object.

## Contexts

The following contexts are available to formulas of this type:

- BUSINESS\_GROUP\_ID ( ENTERPRISE\_ID)
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- PERSON\_ID
- LER\_ID
- ORGANIZATION\_ID
- JURISDICTION\_CODE
- PGM\_ID
- PL\_ID
- PL\_TYP\_ID
- OPT\_ID
- LER\_ID

## Database Items

Use only the following database items that are available to formulas of this type:

- All columns except attribute columns from tables: ben\_ler\_f, ben\_pgm\_f, ben\_pl\_f, ben\_opt\_f, ben\_pl\_typ\_f
- Database items based on person's assignment – HR\_ASSIGNMENT\_ID AND PERSON\_ID

## Input Variables

Input variables are available to formulas of this type.

### *Input variables table*

Input	Data Type	Required	Description
CON_PERSON_ID	Number		Contact person id

## Return Variables

Use predefined names for return variables. The following return variables are available to formulas of this type.

### Return variables table

Return Value	Data Type	Required	Description
L_output	Char	Y	Y or N

## Errors

If any other output value is returned then the participation process errors with BEN\_91329\_FORMULA\_RETURN.

### Sample Formula 1:

FORMULA NAME: BEN\_DPNT\_ELIG

FORMULA TYPE: Dependent Eligibility

DESCRIPTION: One son or daughter are eligible. This formula relies upon benefits seeded formula function - ben\_fn\_get\_char\_value.

```
default for l_sons is 0
default for l_daughters is 0
l_output = 'N'
l_sons_t = ben_fn_get_char_value(
  'PER_CONTACT_RELSHIPS_F', 'SON_COUNT'
  ,
  ,
  ,
  'LEGISLATION_CODE', 'US'
)
l_sons = to_num(l_sons_t)
l_daughters_t = ben_fn_get_char_value(
  'PER_CONTACT_RELSHIPS_F'
  ,
  'DAUGHTER_COUNT'
  ,
  ,
  ,
  'LEGISLATION_CODE'
  ,
  'US'
)
l_daughters = to_num(l_daughters_t)
if ( l_sons <= 1 and l_daughters = 1 ) then
  (l_output = 'Y')
if (l_sons= 1 and l_daughters <= 1) then
  (l_output = 'Y')
return l_output
```

### Sample Formula 2:

FORMULA NAME: BEN\_DPNT\_ELIG2

FORMULA TYPE: Dependent Eligibility

DESCRIPTION: Dependents of persons above 60 yrs are ineligible.

```
DEFAULT for PER_PER_DATE_OF_BIRTH is '01-JAN-1951' (date)
l_output = 'N'
l_dob = PER_PER_DATE_OF_BIRTH
l_eff_date = GET_CONTEXT(EFFECTIVE_DATE,to_date('01-JAN-2012'))
if ( add_months (l_dob,720) > l_eff_date ) then
  (l_output = 'N')
else
  (l_output = 'Y' )
```

```
return l_output
```

## Sample Formula 3:

FORMULA NAME: BEN\_DPNT\_ELIG2

FORMULA TYPE: Dependent Eligibility

DESCRIPTION: Dependents of persons based on age and address, relationship.

```
DEFAULT_DATA_VALUE for PER_EXT_CONT_CONTACT_TYPE is 'NA'
DEFAULT_DATA_VALUE FOR PER_EXT_CONT_DATE_OF_BIRTH IS '1951-01-01' (DATE)
DEFAULT_DATA_VALUE FOR PER_EXT_CONT_PER_PERSON_ID IS 0 DEFAULT FOR PER_AS_G_ADD_REGION2 IS 'NA'
INPUTS ARE CON_PERSON_ID
ELIGIBLE = 'N'
l_lf_evt_dt_char = ben_fn_get_char_value('BEN_PER_IN_LER', 'LF_EVT_OCRD_DT'
,
,
,
'BENEFIT_RELATION_NAME'
,'DFLT')
l_lf_evt_ocrd_dt = TO_DATE(l_lf_evt_dt_char)
i=1
WHILE PER_EXT_CONT_CONTACT_TYPE.exists(i)
LOOP
(
IF (PER_EXT_CONT_PER_PERSON_ID[i] = CON_PERSON_ID) AND
(PER_EXT_CONT_CONTACT_TYPE[i]='SAME_SEX_DOMESTIC_PARTNER_CHIL' OR
PER_EXT_CONT_CONTACT_TYPE[i]='C' OR
PER_EXT_CONT_CONTACT_TYPE[i]='O' OR
PER_EXT_CONT_CONTACT_TYPE[i]='A' OR
PER_EXT_CONT_CONTACT_TYPE[i]='T') AND
(PER_AS_G_ADD_REGION2 = 'OH' or upper(PER_AS_G_ADD_REGION2) = 'OHIO') AND
(MONTHS_BETWEEN(l_lf_evt_ocrd_dt, PER_EXT_CONT_DATE_OF_BIRTH[i])) < 336 THEN
(
ELIGIBLE = 'Y'
)
i=i+1
)
RETURN ELIGIBLE
```

## Age Determination Date

This formula type can be used to determine the date from which age will be calculated for a person.

### Contexts

The following contexts are available to formulas of this type:

- BUSINESS\_GROUP\_ID ( ENTERPRISE\_ID)
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- PERSON\_ID
- LER\_ID
- ORGANIZATION\_ID

- JURISDICTION\_CODE
- PGM\_ID
- PL\_ID
- PL\_TYP\_ID
- OPT\_ID
- LER\_ID

## Database Items

Use only the following database items that are available to formulas of this type:

- All columns except attribute columns from tables: ben\_ler\_f, ben\_pgm\_f, ben\_pl\_f, ben\_opt\_f, ben\_pl\_typ\_f
- Database items based on person's assignment – HR\_ASSIGNMENT\_ID AND PERSON\_ID

## Input Variables

Input variables are not available to formulas of this type.

## Return Variables

Use predefined names for return variables. The following return variables are available to formulas of this type.

### ***Return variables table***

Return Value	Data Type	Required	Description
l_output	Char	Y	Return variable

## Errors

If any other output value is returned then the participation process errors with BEN\_91329\_FORMULA\_RETURN.

### Sample Formula 1:

FORMULA NAME: BEN\_AGE\_DET\_DT

FORMULA TYPE: Age Determination Date

DESCRIPTION: The Age determination date is the start date of next month.

```
l_effective_date = GET_CONTEXT(EFFECTIVE_DATE, TO_DATE('01-JAN-2012'))
l_last_day = last_day(l_effective_date)
l_output = add_days(l_last_day,1)
return l_output
```

### Sample Formula 2:

FORMULA NAME: BEN\_AGE\_DET\_DT

FORMULA TYPE: Age Determination Date

**DESCRIPTION:** The Age determination date is based on person's job.

```
DEFAULT for PER_ASG_JOB_NAME is 'XYZ'  
l_job = PER_ASG_JOB_NAME  
l_effective_date = GET_CONTEXT(EFFECTIVE_DATE, TO_DATE('01-JAN-2012'))  
l_output = last_day(add_months(l_effective_date,-1))  
if l_job = 'Remote Worker' then  
( l_output = add_days(l_output,1) )  
Else  
( l_output = last_day(l_effective_date))  
return l_output
```

## Enrollment Opportunity

This formula type can be used to determine whether a benefits object should be an electable choice for the person.

### Contexts

The following contexts are available to formulas of this type:

- BUSINESS\_GROUP\_ID ( ENTERPRISE\_ID)
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- PERSON\_ID
- LER\_ID
- ORGANIZATION\_ID
- JURISDICTION\_CODE
- PGM\_ID
- PL\_ID
- PL\_TYP\_ID
- OPT\_ID
- LER\_ID

### Database Items

Use only the following database items that are available to formulas of this type:

- All columns except attribute columns from tables: ben\_ler\_f, ben\_pgm\_f, ben\_pl\_f, ben\_opt\_f, ben\_pl\_typ\_f
- Database items based on person's assignment – HR\_ASSIGNMENT\_ID AND PERSON\_ID

### Input Variables

Input variables are not available to formulas of this type.

### Return Variables

Use predefined names for return variables. The following return variables are available to formulas of this type.

### Return variables table

Return Value	Data Type	Required	Description
L_ENRT_OPP	Char	Y	<p>Y - Yes</p> <p>A - Leave as choice and use elctbl_flag</p> <p>L - Lose only condition, if enrolled, will be de-enrolled</p> <p>N - No</p>

## Errors

If any other output value is returned then the participation process errors with BEN\_91329\_FORMULA\_RETURN.

### Sample Formula 1:

FORMULA NAME: BEN\_ENRT\_OPTY

FORMULA TYPE: Enrollment Opportunity

DESCRIPTION: Formula to determine whether a compensation object should be an electable choice for the person.

```

DEFAULT for BEN_LER_NAME is 'ABC'
DEFAULT for BEN_PLN_NAME is 'XYZ'
DEFAULT for BEN_OPT_NAME is 'XYZ'
l_enrt_opp = 'A'
l_ler_name = BEN_LER_NAME
l_pl_name = BEN_PLN_NAME
l_opt_name = BEN_OPT_NAME
If ( l_pl_name = 'Aetna Medical'
and ( l_opt_name = 'Domestic Partner' or l_opt_name = 'Domestic Partner + one')
and ( l_ler_name = 'Open' or l_ler_name = 'New Hire' ) ) then
( l_enrt_opp = 'N' )
If ( l_pl_name = 'Be Healthy HMO'
and ( l_opt_name = 'Employee + family' or l_opt_name = 'Employee + spouse')
and ( l_ler_name <> 'Open' ) ) then
(l_enrt_opp = 'A' )
Return l_enrt_opp

```

### Sample Formula 2:

FORMULA NAME: BEN\_ENROLLMENT\_OPPORTUNITY

FORMULA TYPE: Enrollment Opportunity

```

Default for BEN_PLN_NAME is 'Default'
Default for BEN_LER_NAME is 'Default'
l_return='N'
l_med_wv='N'
l_pt2_p1='N'
l_pt2_p1_opt1='N'
l_pt2_p1_opt2='N'
l_pt2_p1_opt3='N'
l_pt2_p2='N'
l_pt2_p2_opt1='N'

```

```

l_pt2_p2_opt2='N'
l_pt2_p2_opt3='N'
l_pt2_p2='N'
/* Below is a function which returns Y if there is an enrollment */
l_med_wv=ben_fn_get_char_value(
'BEN_PRTT_ENRT_RSLT'
,'ENROLLED'
,'SO_CHANGE_PT2_PLAN_3_WAIVE'
,'NA'
)
if (BEN_PLN_NAME='SO_CHANGE_PT2_PLAN_1') then (
l_pt2_p1_opt1=ben_fn_get_char_value(
'BEN_PRTT_ENRT_RSLT'
,'ENROLLED'
,'SO_CHANGE_PT2_PLAN_1'
,'SO_CHANGE_PT2_PLAN_1_OPT_1'
)
l_pt2_p1_opt2=ben_fn_get_char_value(
'BEN_PRTT_ENRT_RSLT'
,'ENROLLED'
,'SO_CHANGE_PT2_PLAN_1'
,'SO_CHANGE_PT2_PLAN_1_OPT_2'
)
l_pt2_p1_opt3=ben_fn_get_char_value(
'BEN_PRTT_ENRT_RSLT'
,'ENROLLED'
,'SO_CHANGE_PT2_PLAN_1'
,'SO_CHANGE_PT2_PLAN_1_OPT_3'
)
) /* end if for SO_CHANGE_PT2_PLAN_1 */
if (l_pt2_p1_opt1='Y' or l_pt2_p1_opt2='Y' or l_pt2_p1_opt3='Y') then (l_pt2_p1='Y')
58 Oracle Benefits Fast Formula Reference Guide / Version 1.6
Copyright © 2021, Oracle and/or its affiliates / Public
if (BEN_PLN_NAME='SO_CHANGE_PT2_PLAN_2') then (
l_pt2_p2_opt1=ben_fn_get_char_value(
'BEN_PRTT_ENRT_RSLT'
,'ENROLLED'
,'SO_CHANGE_PT2_PLAN_2'
,'SO_CHANGE_PT2_PLAN_2_OPT_1'
)
l_pt2_p2_opt2=ben_fn_get_char_value(
'BEN_PRTT_ENRT_RSLT'
,'ENROLLED'
,'SO_CHANGE_PT2_PLAN_2'
,'SO_CHANGE_PT2_PLAN_2_OPT_2'
)
l_pt2_p2_opt3=ben_fn_get_char_value(
'BEN_PRTT_ENRT_RSLT'
,'ENROLLED'
,'SO_CHANGE_PT2_PLAN_2'
,'SO_CHANGE_PT2_PLAN_2_OPT_3'
)
) /*end if SO_CHANGE_PT2_PLAN_2 */
if (l_pt2_p2_opt1='Y' or l_pt2_p2_opt2='Y' or l_pt2_p2_opt3='Y') then (l_pt2_p2='Y')
if (l_med_wv='Y') then
(
l_return='Y'
)
else if ( (l_med_wv='N') and (BEN_LER_NAME='SO_MARRIAGE_1') ) then
(
if ( (BEN_PLN_NAME='SO_CHANGE_PT2_PLAN_1') and (l_pt2_p1='Y') ) then (l_return='Y')
if ( (BEN_PLN_NAME='SO_CHANGE_PT2_PLAN_2') and (l_pt2_p2='Y') ) then (l_return='Y')
) /*end else if */
return l_return

```

## Sample Formula 3:

FORMULA TYPE : Enrollment Opportunity

DESCRIPTION: Provide enrollment opportunity if person is rehired within 30 days of termination.

```

DEFAULT for PER_PER_LATEST_REHIRE_DATE is '1951/01/01 00:00:00' (date)
DEFAULT for PER_PER_LATEST_TERMINATION_DATE is '1951/01/01 00:00:00' (date)
DEFAULT_DATA_VALUE FOR PER_HIST_ASG_ASSIGNMENT_ID is 99
DEFAULT_DATA_VALUE FOR PER_HIST_ASG_EFFECTIVE_START_DATE is '1951/01/01 00:00:00' (date)
DEFAULT_DATA_VALUE FOR PER_HIST_ASG_EFFECTIVE_END_DATE is '1951/01/01 00:00:00' (date)
DEFAULT_DATA_VALUE FOR PER_HIST_ASG_ASSIGNMENT_TYPE is 'NA'
DEFAULT_DATA_VALUE FOR PER_HIST_ASG_STATUS_TYPE is 'NA'
DEFAULT_DATA_VALUE FOR PER_HIST_ASG_EFFECTIVE_LATEST_CHANGE is 'NA'
/* if the person is a first time hire, the PER_PER_LATEST_REHIRE_DATE will return the hire date */
l_hire_dt = PER_PER_LATEST_REHIRE_DATE
l_term_dt = PER_PER_LATEST_TERMINATION_DATE
l_prev_asg_id=9999
l_enrt_opportunity = 'N'
if PER_PER_LATEST_TERMINATION_DATE was not defaulted then
(
if l_hire_dt < add_days(l_term_dt,30)
then
(
/* person is rehired within 30 days */
/* get the assignment_id */
l_prev_asg_dt = add_days(l_term_dt,-1)
i=1
while ( PER_HIST_ASG_ASSIGNMENT_ID.exists(i) ) loop
(
if ( (PER_HIST_ASG_EFFECTIVE_START_DATE[i] < l_prev_asg_dt)
AND (PER_HIST_ASG_EFFECTIVE_END_DATE[i] >= l_prev_asg_dt)
AND (PER_HIST_ASG_ASSIGNMENT_TYPE[i] = 'ET')
AND (PER_HIST_ASG_STATUS_TYPE[i] = 'ACTIVE')
AND (PER_HIST_ASG_EFFECTIVE_LATEST_CHANGE[i] = 'Y' )
)
then
(
l_prev_asg_id = PER_HIST_ASG_ASSIGNMENT_ID[i]
) /* end if */
i=i+1
) /* end while */
/* check if the person is enrolled in any of the medical plan */
CHANGE_CONTEXTS(HR_ASSIGNMENT_ID=l_prev_asg_id, EFFECTIVE_DATE=l_prev_asg_dt)
(
l_med_enroll = ben_fn_get_char_value(
'BEN_PRTT_ENRT_RSLT'
,'ENROLLED'
,'Healthy HMO'
,'NA'
)
if (l_med_enroll = 'Y') then
(
l_enrt_opportunity = 'Y'
)
) /* end change contexts */
) /* < 30 days */
) /* end not defaulted */
return l_enrt_opportunity

```

## Sample Formula 4:

FORMULA\_NAME: ADD\_CHANGE\_BENEFICIARY

**DESCRIPTION:** Provide opportunity only if already enrolled.

```
DEFAULT for BEN_LER_NAME is 'NA'  
DEFAULT for BEN_PLN_NAME is 'NA'  
DEFAULT for BEN_OPT_NAME is 'NA'  
l_opp = 'N'  
l_flag = ben_fn_get_char_value('BEN_PRTT_ENRT_RSLT', 'ENROLLED', BEN_PLN_NAME, BEN_OPT_NAME)  
IF ( l_flag = 'Y') then  
{  
l_opp = 'Y'  
}  
return l_opp
```

## Beneficiary Certification Required

This formula type can be used to determine the conditions under which a person must provide certification for his or her designated beneficiaries.

### Contexts

The following contexts are available to formulas of this type:

- BUSINESS\_GROUP\_ID ( ENTERPRISE\_ID)
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- PERSON\_ID
- LER\_ID
- ORGANIZATION\_ID
- JURISDICTION\_CODE
- PGM\_ID
- PL\_ID
- PL\_TYP\_ID
- OPT\_ID
- LER\_ID

### Database Items

Use only the following database items that are available to formulas of this type:

- All columns except attribute columns from tables: ben\_ler\_f, ben\_pgm\_f, ben\_pl\_f, ben\_opt\_f, ben\_pl\_typ\_f
- Database items based on person's assignment – HR\_ASSIGNMENT\_ID AND PERSON\_ID

### Input Variables

Input variables are not available to formulas of this type.

## Return Variables

Use predefined names for return variables. The following return variables are available to formulas of this type.

### ***Return variables table***

Return Value	Data Type	Required	Description
L_OUTPUT	Char	Y	Y or N

## Errors

If any other output value is returned then the participation process errors with BEN\_91329\_FORMULA\_RETURN.

### Sample Formula 1:

FORMULA NAME: BEN\_BNFCR\_CERT\_REQD

FORMULA TYPE: Beneficiary Certification Required

DESCRIPTION: A Single person need to provide certificates for beneficiary.

```
DEFAULT for PER_PER_MARITAL_STATUS is 'X'
l_output = 'N'
l_mar_status = PER_PER_MARITAL_STATUS
if (l_mar_status = 'S')
then
(l_output = 'Y')
else
(l_output = 'N')
Return l_output
```

### Sample Formula 2:

FORMULA NAME: BEN\_BNFCR\_CERT\_REQD

FORMULA TYPE: Beneficiary Certification Required

DESCRIPTION: Person needs to provide certificates for plans which allow minors/trustees and organizations as beneficiaries.

```
DEFAULT for BEN_PLN_BNF_DSGE_MNR_TTEE_RQD_FLAG is 'X'
DEFAULT for BEN_PLN_BNF_MAY_DSGT_ORG_FLAG is 'X'
l_output = 'N'
l_minor_tte = BEN_PLN_BNF_DSGE_MNR_TTEE_RQD_FLAG
l_org_bnf = BEN_PLN_BNF_MAY_DSGT_ORG_FLAG
if (l_minor_tte = 'Y' or l_org_bnf = 'Y' )
then
(l_output = 'Y')
else
(l_output = 'N')
Return l_output
```

# Compensation Calculation

This formula type can be used to determine a total compensation amount to be used for eligibility, coverage or benefit, and rate calculations.

## Contexts

The following contexts are available to formulas of this type:

- BUSINESS\_GROUP\_ID ( ENTERPRISE\_ID)
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- PERSON\_ID
- LER\_ID
- ORGANIZATION\_ID
- JURISDICTION\_CODE
- PGM\_ID
- PL\_ID
- PL\_TYP\_ID
- OPT\_ID
- LER\_ID

## Database Items

Use only the following database items that are available to formulas of this type:

- All columns except attribute columns from tables: ben\_ler\_f, ben\_pgm\_f, ben\_pl\_f, ben\_opt\_f, ben\_pl\_typ\_f
- Database items based on person's assignment – HR\_ASSIGNMENT\_ID AND PERSON\_ID.

## Input Variables

Input variables are available to formulas of this type.

### *Input variables table*

Input	Data Type	Required	Description
PERSON_ID	Number		Person ID as input value

## Return Variables

Use predefined names for return variables. The following return variables are available to formulas of this type.

### Return variables table

Return Value	Data Type	Required	Description
L_OUTPUT	Number	Y	Return variable

## Errors

If any other output value is returned then the participation process errors with BEN\_91329\_FORMULA\_RETURN.

### Sample Formula 1:

FORMULA NAME: BEN\_COMP\_CALC

FORMULA TYPE: Compensation calculation

DESCRIPTION: Calculate gross annual salary.

```
DEFAULT for CMP_ASSIGNMENT_SALARY_AMOUNT is 0
DEFAULT for CMP_ASSIGNMENT_SALARY_ANNUALIZATION_FACTOR is 0
l_output = 0
l_sal = CMP_ASSIGNMENT_SALARY_AMOUNT
l_ann_fctr = CMP_ASSIGNMENT_SALARY_ANNUALIZATION_FACTOR
l_output = l_sal * l_ann_fctr
Return l_output
```

### Sample Formula 2:

FORMULA NAME: BEN\_COMP\_CALC\_BAR\_UNIT

FORMULA TYPE: Compensation calculation

DESCRIPTION: Calculate gross annual compensation based on bargaining unit.

```
DEFAULT for CMP_ASSIGNMENT_SALARY_AMOUNT is 0
DEFAULT for CMP_ASSIGNMENT_SALARY_ANNUALIZATION_FACTOR is 0 DEFAULT for PER_ASG_BARGAINING_UNIT_CODE is
'xyz'
l_output = 0
l_sal = CMP_ASSIGNMENT_SALARY_AMOUNT
l_ann_fctr = CMP_ASSIGNMENT_SALARY_ANNUALIZATION_FACTOR l_unit = PER_ASG_BARGAINING_UNIT_CODE
if (l_unit = 'Technical Unit' or l_unit = 'Service Unit') then l_output = (l_sal * l_ann_fctr) * 1.1 else
l_output = (l_sal * l_ann_fctr) * 1.05
Return l_output
```

### Sample Formula 3:

FORMULA TYPE: Compensation Calculation

```
DEFAULT FOR PER_ASG_FTE_VALUE IS 0
DEFAULT FOR CMP_ASSIGNMENT_SALARY_ANNUALIZATION_FACTOR IS 0
DEFAULT FOR CMP_ASSIGNMENT_SALARY_AMOUNT IS 0
l_cmp_amt = CMP_ASSIGNMENT_SALARY_AMOUNT
l_ann_sal= 0
l_ann_fctr = CMP_ASSIGNMENT_SALARY_ANNUALIZATION_FACTOR
l_fte = PER_ASG_FTE_VALUE
IF (l_ann_fctr > 1) THEN
(
l_ann_sal=(l_cmp_amt * l_fte * l_ann_fctr)
)
```

```

ELSE
(
1_ann_sal=(1_cmp_amt * 1_ann_fctr)
)
RETURN 1_ann_sal

```

## Sample Formula 4:

FORMULA TYPE: Compensation Calculation

Description: For executives pick the compensation from the benefits balance if already defined.

```

DEFAULT FOR CMP_ASSIGNMENT_SALARY_ANNUAL_AMOUNT IS 0
DEFAULT_DATA_VALUE FOR BEN_PBB_BNFTS_BAL_NAME_TN IS 'NA'
DEFAULT_DATA_VALUE FOR BEN_PBB_VAL_NN IS 0
DEFAULT FOR BEN_PIL_LF_EVT_OCRD_DT IS '1951/01/01 00:00:00' (date)
1_ann_sal = CMP_ASSIGNMENT_SALARY_ANNUAL_AMOUNT
1_lf_evt_ocrd_dt = BEN_PIL_LF_EVT_OCRD_DT
i=1
1_bal = 0
1_val = 0
while BEN_PBB_BNFTS_BAL_NAME_TN.exists(i)
loop
(
IF BEN_PBB_BNFTS_BAL_NAME_TN[i] = 'Executive Salary' THEN
(
1_bal = BEN_PBB_VAL_NN[i]
exit
)
i=i+1
) /* end while */
1_val = 1_bal/12
IF 1_val = 0 THEN
( 1_val = 1_ann_sal/12)
RETURN 1_val

```

## Post Election Edit

This formula type can be used to enforce rules while making enrollments. Example: Enroll in basic life plan when you enroll in spousal or child life plans.

## Contexts

The following contexts are available to formulas of this type:

- BUSINESS\_GROUP\_ID ( ENTERPRISE\_ID)
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- PERSON\_ID
- LER\_ID
- ORGANIZATION\_ID
- JURISDICTION\_CODE
- PGM\_ID

- PL\_ID
- PL\_TYP\_ID
- OPT\_ID
- LER\_ID

**Note:** OPT\_ID is not available if the edit rule is attached at plan or plan type level, and no options exists for corresponding compensation object.

## Database Items

Use only the following database items that are available to formulas of this type:

- All columns except attribute columns from tables: ben\_ler\_f, ben\_pgm\_f, ben\_pl\_f, ben\_opt\_f, ben\_pl\_typ\_f
- Database items based on person's assignment – HR\_ASSIGNMENT\_ID AND PERSON\_ID

## Input Variables

Input variables are not available to formulas of this type.

## Return Variables

Use predefined names for return variables. The following return variables are available to formulas of this type.

### ***Return variables table***

Return Value	Data Type	Required	Description
SUCCESSFUL	Char	Y	Y or N
ERROR_MESSAGE	Char	Y	Error message to be displayed

## Errors

If any other output value is returned then the participation process errors with BEN\_91329\_FORMULA\_RETURN.

## Sample Formula 1:

FORMULA NAME: BEN\_POST\_ELEC\_EDIT

FORMULA TYPE: Post Election

DESCRIPTION: When a person makes an election to Spousal supplemental, validate the person is also enrolled in plan Basic Life. This formula relies upon benefits seeded formula function - ben\_fn\_get\_char\_value.

```
default for l_enrlid is 'Y'  
default for l_dpnt_enrlid1 is 'Y'  
default for l_dpnt_enrlid2 is 'Y'  
SUCCESSFUL = 'Y'  
ERROR_MESSAGE=''  
l_enrlid = ben_fn_get_char_value(  
'BEN_PRTT_ENRT_RSLT'  
, 'ENROLLED'
```

```

, 'Spousal Supplemental'
, 'NA'
)
l_dpnt_enrld1 = ben_fn_get_char_value(
'BEN_PRTT_ENRT_RSLT'
,'ENROLLED'
,'Basic_Life'
,'Lifex1'
)
l_dpnt_enrld2 = ben_fn_get_char_value(
'BEN_PRTT_ENRT_RSLT'
,'ENROLLED'
,'Basic_Life'
,'Lifex2'
)
If ( l_enrld = 'Y' and l_dpnt_enrld1='N' and l_dpnt_enrld2='N' ) then (SUCCESSFUL='N'
ERROR_MESSAGE = 'Election to any option within plan Spousal Supplemental, you should also enroll in plan
Basic_Life.')
return SUCCESSFUL,ERROR_MESSAGE

```

## Sample Formula 2:

FORMULA NAME: POSTELECTIONEDIT\_GREEN\_COMPANY

FORMULA TYPE: Postelection Edit using db array items

DESCRIPTION : When a person makes an election to plan name PPO (Out of Area), validate the person is also enrolled in plan Aetna PPO . If yes, return 'Y' to the post enrollment save edit. If not, return 'N' (error condition).

```

DEFAULT_DATA_VALUE for BEN_PEN_PL_NAME_TN is 'My-Default'
default for l_enrolled is 'N'
default for l_dpnt_enrolled is 'N'
default for i is 1
default for SUCCESSFUL is 'Y'
default for ERROR_MESSAGE is ' '
*-----
SUCCESSFUL='Y'
ERROR_MESSAGE=' '
i=1
l_enrolled='N'
l_dpnt_enrolled='N'
while BEN_PEN_PL_NAME_TN.exists(i) loop
(
if BEN_PEN_PL_NAME_TN[i]='PPO (Out of Area)' then (l_enrolled='Y')
if BEN_PEN_PL_NAME_TN[i]='Aetna PPO' then
(l_dpnt_enrolled='Y')
i=i+1
)
if l_enrolled='Y' and l_dpnt_enrolled='N' then
(SUCCESSFUL='N'
ERROR_MESSAGE='Election to PPO (Out of Area) requires enr in Aetna PPO.')
)
return SUCCESSFUL,ERROR_MESSAGE

```

## Sample Formula 3:

FORMULA TYPE: Postelection Edit

Description: Ensure that a person gets enrolled either as participant or as a dependent when spouse/domestic partners work in the same organization.

```

DEFAULT_DATA_VALUE FOR PER_PER_CONT_REL_CONTACT_TYPE IS 'NA'
DEFAULT_DATA_VALUE FOR PER_PER_CONT_REL_CONTACT_PERSON_ID IS -99

```

```

DEFAULT_DATA_VALUE FOR BEN_EXT_RSLT_PL_NAME IS 'NA'
l_couple_flag = 'N'
l_sp_dp_id = 0
SUCCESSFUL = 'Y'
ERROR_MESSAGE = ''
i =1
WHILE (PER_PER_CONT_REL_CONTACT_TYPE.EXISTS(i))
LOOP
(
IF ((PER_PER_CONT_REL_CONTACT_TYPE[i] = 'S') OR (PER_PER_CONT_REL_CONTACT_TYPE[i] = 'DP')) THEN
(
l_sp_dp_id = PER_PER_CONT_REL_CONTACT_PERSON_ID[i]
)
i = i + 1
)
i = 1
CHANGE_CONTEXTS(PERSON_ID = l_sp_dp_id)
(
WHILE(BEN_EXT_RSLT_PL_NAME.EXISTS(i))
LOOP
(
IF ( BEN_EXT_RSLT_PL_NAME[i] = 'Dependent Life Insurance - Spouse/DP' )
THEN
( l_couple_flag = 'Y'
)
/* the dependent has enrolled the participant */
i = i + 1
)
)
IF (l_couple_flag = 'Y')
THEN
(
SUCCESSFUL = 'N'
ERROR_MESSAGE = 'You cannot enroll in Life Insurance as you are already covered in Dependent Life Insurance
- Spouse/DP under your spouse'
)
)
RETURN SUCCESSFUL, ERROR_MESSAGE

```

## Sample Formula 4:

FORMULA TYPE: Postelection Edit

```

DEFAULT_DATA_VALUE for BEN_PEN_PL_NAME_TN is 'My-Default'
DEFAULT_DATA_VALUE for BEN_PEN_OPT_NAME_TN is 'My-Default'
DEFAULT FOR BEN_PIL_LF_EVT_OCRD_DT is '1950/01/01 00:00:00'(date)
DEFAULT FOR BEN_LER_NAME is 'abc'
SUCCESSFUL='N'
ERROR_MESSAGE=' '
i=1
l_plan1_enrld='N'
l_plan2_enrld='N'
l_new_effective_date = GET_CONTEXT(EFFECTIVE_DATE,to_date('1951/01/0100:00:00') )
l_person_id = GET_CONTEXT(PERSON_ID,0 )
l_last_day = '1950/01/01 00:00:00'(date)
l_ler_name = BEN_LER_NAME
l_lf_evt_ocrd_dt = BEN_PIL_LF_EVT_OCRD_DT
l_new_effective_date = l_lf_evt_ocrd_dt
l_log_data = ESS_LOG_WRITE( 'Effective Date ='||to_char(l_new_effective_date))
ERROR_MESSAGE = to_text(l_new_effective_date)||' : '
CHANGE_CONTEXTS(EFFECTIVE_DATE = l_new_effective_date)
(
while BEN_PEN_PL_NAME_TN.exists(i) loop
(
if (BEN_PEN_PL_NAME_TN[i] = 'Voluntary Child Life' ) then
(

```

```

l_plan1_enrld= 'Y'
ERROR_MESSAGE = 'Vol Life : '
l_log_data = ESS_LOG_WRITE('Employee has elected Voluntary Spouse Life')
)
if (BEN_PEN_PL_NAME_TN[i] = 'Supplemental Employee Life') then
(
l_plan2_enrld= 'Y'
ERROR_MESSAGE = ERROR_MESSAGE||'Supp Life : '
l_log_data = ESS_LOG_WRITE('Employee has elected Supplemental Employee Life')
)
i=i+1
)
)
IF (l_plan1_enrld = 'Y') THEN
(
l_log_data = ESS_LOG_WRITE('Employee has elected Voluntary Child Life')
IF (l_plan2_enrld= 'Y') THEN
(
l_log_data = ESS_LOG_WRITE('Employee has elected Supplemental Employee Life')
SUCCESSFUL='Y'
return SUCCESSFUL,ERROR_MESSAGE
)
ELSE
(
l_log_data = ESS_LOG_WRITE('Employee has NOT elected Supplemental Employee Life')
SUCCESSFUL='N'
ERROR_MESSAGE = ERROR_MESSAGE||'You cannot elected Voluntary Child Life without electing Supplemental
Employee Life as well '
return SUCCESSFUL,ERROR_MESSAGE
)
)
ELSE
(
SUCCESSFUL='Y'
return SUCCESSFUL,ERROR_MESSAGE
)

```

## Sample Formula 5:

Problem statement: Customer is loading participant enrollments using HDL. They have a post-election edit formula which forces workers to enroll dependents in order to submit elections. When loading participant enrollments using HDL, this formula is getting triggered and the enrollments are not getting loaded.

While conversion load is being performed, you can remove the formula from plan design, perform the load for participant and dependents and then add it back.

Other option is to use below formula functions -

- BEN\_GET\_PROCESS\_NAME\_SS() is provided for Spreadsheet loader
- BEN\_GET\_PROCESS\_NAME() is provided for HDL Enrollment

When using participant HDL, you can use code like below to skip these validations while loading enrollments -

```
optext= BEN_GET_PROCESS_NAME() if ( optext != 'PARTICIPANT_ENROLLMENT' ) then ( 'your checks go here' )
For other 2 HDL loaders, the values you can use are -
```

- Dependent Enrollment using HDL - 'DEPENDENT\_DESIGNATION'
- Beneficiary Enrollment using HDL - 'BENEFICIARY\_DESIGNATION'

FORMULA TYPE: Postelection Edit

Description: Private Health plan requires designation of at least one dependent.

```

DEFAULT_DATA_VALUE for BEN_PEN_PL_NAME_TN is 'My-Default'
DEFAULT_DATA_VALUE for BEN_PEN_OPT_NAME_TN is 'My-Default'
DEFAULT_DATA_VALUE for BEN_EXT_DPNT_PL_NAME IS 'XYZ'
DEFAULT_DATA_VALUE for BEN_EXT_BNF_OPT_NAME IS 'ABC'
DEFAULT FOR BEN_PIL_LF_EVT_OCRD_DT IS '1950/01/01 12:00:00' (date)
SUCCESSFUL='Y'
ERROR_MESSAGE=' '
i=1
j=1
l_family = 'N'
l_count = 1
l_lf_evt_ocrd_dt = BEN_PIL_LF_EVT_OCRD_DT
l_phase = BEN_GET_PROCESS_NAME()
/* skip the rule when loading participant enrollments through HDL */
if (l_phase != 'PARTICIPANT_ENROLLMENT')
then
(
CHANGE_CONTEXTS(EFFECTIVE_DATE = l_lf_evt_ocrd_dt)
(
while BEN_PEN_PL_NAME_TN.exists(i) loop
(
if BEN_PEN_PL_NAME_TN[i] like 'Private Health%' then
(
if BEN_PEN_OPT_NAME_TN[i] like 'Couple%' then
(
l_family = 'Y'
EXIT
)
)
)
i=i+1
) /* end while */
IF l_family = 'Y' THEN
(
j=1
while BEN_EXT_DPNT_PL_NAME.exists(j)
loop
(
if ( BEN_EXT_DPNT_PL_NAME[i] like 'Private Health%' and BEN_EXT_BNF_OPT_NAME[i] like 'Couple%' )
then
(
l_plan =BEN_EXT_DPNT_PL_NAME[i]
l_option=BEN_EXT_BNF_OPT_NAME[i]
l_count = l_count + 1
exit
)
j=j+1
) /* end loop for checking for dependents */
) /* if l_family */
) /* end change contexts */
IF l_family = 'Y' and l_count=1 THEN
(
SUCCESSFUL = 'N'
ERROR_MESSAGE='Designate at least one dependent in your Healthcare plans' ||to_char(i)
)
) /* end l_phase */
RETURN SUCCESSFUL, ERROR_MESSAGE

```

# Enrollment Period Start Date

This formula type can be used to determine the date from which the enrollment period starts.

## Contexts

The following contexts are available to formulas of this type:

- BUSINESS\_GROUP\_ID ( ENTERPRISE\_ID)
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- PERSON\_ID
- LER\_ID
- ORGANIZATION\_ID
- JURISDICTION\_CODE
- PGM\_ID
- PL\_ID
- PL\_TYP\_ID
- OPT\_ID
- LER\_ID

## Database Items

Use only the following database items that are available to formulas of this type:

- All columns except attribute columns from tables: ben\_ler\_f, ben\_pgm\_f, ben\_pl\_f, ben\_opt\_f, ben\_pl\_typ\_f
- Database items based on person's assignment – HR\_ASSIGNMENT\_ID AND PERSON\_ID

## Input Variables

Input variables are not available to formulas of this type.

## Return Variables

Use predefined names for return variables. The following return variables are available to formulas of this type.

### ***Return variables table***

Return Value	Data Type	Required	Description
L_start_dt	Date		Return variable

## Errors

If any other output value is returned then the participation process errors with BEN\_91329\_FORMULA\_RETURN.

### Sample Formula 1:

FORMULA NAME: BEN\_ENRT\_PERD\_START

FORMULA TYPE: Enrollment Period start date

DESCRIPTION: Enrollment period starts MID of next month of marital status change date.

```
default for PER_ASG_MARITAL_STATUS_DATE is '01-JAN-1951'(date)
L_OCRD_DT = PER_ASG_MARITAL_STATUS_DATE
L_EFF_DATE = GET_CONTEXT(EFFECTIVE_DATE, to_date('01-JAN-2012'))
l_start_dt = L_EFF_DATE
if ( L_OCRD_DT < L_EFF_DATE ) then
( l_last_dt = last_day(L_OCRD_DT)
l_start_dt = add_days(l_last_dt,15)
)
else
( l_start_dt = L_EFF_DATE
)
return l_start_dt
```

Same formula using canonical date formats:

```
default for PER_PER_MARITAL_STATUS_DATE is '1951/01/01 00:00:00'(date)
L_OCRD_DT = PER_PER_MARITAL_STATUS_DATE
L_EFF_DATE = GET_CONTEXT(EFFECTIVE_DATE, TO_DATE('1951/01/01
00:00:00','YYYY/MM/DD HH:MI:SS'))
l_start_dt = L_EFF_DATE
if ( L_OCRD_DT <= L_EFF_DATE ) then
( l_last_dt = last_day(L_OCRD_DT)
l_start_dt = add_days(l_last_dt,15)
)
else
( l_start_dt = L_EFF_DATE
)
return l_start_dt
```

## Default Enrollment

This formula type can be used to determine whether an option in plan should be assigned to a person as part of the default enrollment process.

### Contexts

The following contexts are available to formulas of this type:

- BUSINESS\_GROUP\_ID ( ENTERPRISE\_ID)
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- PERSON\_ID

- LER\_ID
- ORGANIZATION\_ID
- JURISDICTION\_CODE
- PGM\_ID
- PL\_ID
- PL\_TYP\_ID
- OPT\_ID

## Database Items

Use only the following database items that are available to formulas of this type:

- All columns except attribute columns from tables: ben\_ler\_f, ben\_pgm\_f, ben\_pl\_f, ben\_opt\_f, ben\_pl\_typ\_f
- Database items based on person's assignment – HR\_ASSIGNMENT\_ID AND PERSON\_ID

## Input Variables

Input variables are available to formulas of this type.

### *Input variables table*

Input	Data Type	Required	Description
NUM_ELIG_DPNT	Number		Number of eligible dependents

## Return Variables

Use predefined names for return variables. The following return variables are available to formulas of this type.

### *Return variables table*

Return Value	Data Type	Required	Description
AUTO_DFLT_VAL	Char	Y	Y or N
CARRY_FORWARD_ELIG_DPNT	Char	Y	CFRRWP - Carry Forward or Reinstate Within Program  CFWP - Carry Forward Within Program

## Errors

If any other output value is returned then the participation process errors with BEN\_91329\_FORMULA\_RETURN.

## Sample Formula 1:

FORMULA NAME: BEN\_DFLT\_ENRT

FORMULA TYPE: Default Enrollment

DESCRIPTION: Formula to determine whether a compensation object should be an defaulted for the person.

```

INPUTS are NUM_ELIG_DPNT
Default for BEN_OPT_NAME is 'aa'
Default for BEN_PLN_NAME is 'aa'
Default for NUM_ELIG_DPNT is 0
l_pln_name = BEN_PLN_NAME
l_opt_name = BEN_OPT_NAME
l_num_elig_dpnt = NUM_ELIG_DPNT
IF l_pln_name = 'Stay Healthy HMO' then
(
if l_opt_name = 'Employee Only' and l_num_elig_dpnt = 0 then
(
AUTO_DFLT_VAL = 'Y'
CARRY_FORWARD_ELIG_DPNT = 'CFWP'
)
else if l_opt_name = 'Employee Plus One' and l_num_elig_dpnt = 1 then
(
AUTO_DFLT_VAL = 'Y'
CARRY_FORWARD_ELIG_DPNT = 'CFWP'
)
else if l_opt_name = 'Employee Plus Family' and
l_num_elig_dpnt > 1 then
(
AUTO_DFLT_VAL = 'Y'
CARRY_FORWARD_ELIG_DPNT = 'CFWP'
)
else
(AUTO_DFLT_VAL = 'N')
)
else
(AUTO_DFLT_VAL = 'N')
return AUTO_DFLT_VAL, CARRY_FORWARD_ELIG_DPNT

```

## Sample Formula 2:

FORMULA TYPE: Default Enrollment

Description: Derive the default plan/option based on the labor union membership.

```

Default for BEN_OPT_NAME is 'NA'
Default for BEN_PLN_NAME is 'NA'
Default for PER_ASG_LABOUR_UNION_MEMBER_FLAG is 'XXX'
DEFAULT_DATA_VALUE for BEN_EPE_PI_NAME_TN is 'NA'
DEFAULT_DATA_VALUE for BEN_EPE_OPT_NAME_TN is 'NA'
DEFAULT FOR PER_ASG_ORG_LEGAL_EMPLOYER_NAME is 'XXX'
l_pln_name = BEN_PLN_NAME
l_opt_name = BEN_OPT_NAME
l_flag = PER_ASG_LABOUR_UNION_MEMBER_FLAG
AUTO_DFLT_VAL = 'N'
l_le_name = 'Vision Corporation'
l_asg_leg_name = PER_ASG_ORG_LEGAL_EMPLOYER_NAME
l_member_plan = 'Healthy HMO'
l_non_plan = 'Healthy HMO (NM)'
l_default_plan = 'NA'
l_default_option = 'Employee Only'
l_epe_created = 'N'
i=1

```

```

WHILE BEN_EPE_PL_NAME_TN.exists(i) LOOP
(
IF ( l_asg_leg_name = l_le_name
AND l_flag = 'Y'
AND BEN_EPE_PL_NAME_TN[i] = l_member_plan
AND BEN_EPE_OPT_NAME_TN[i] = l_default_option) THEN
(
l_epe_created = 'Y'
l_default_plan = l_member_plan
EXIT
)
ELSE IF ( l_asg_leg_name <> l_le_name
AND l_flag = 'N'
AND BEN_EPE_PL_NAME_TN[i] = l_non_plan
AND BEN_EPE_OPT_NAME_TN[i] = l_default_option) THEN
(
l_epe_created = 'Y'
l_default_plan = l_non_plan
EXIT
)
i = i+1
)
IF ( l_epe_created = 'Y' AND l_pln_name = l_default_plan AND l_opt_name = l_default_option) THEN
(
AUTO_DFLT_VAL = 'Y'
CARRY_FORWARD_ELIG_DPNT = 'CFWP'
)
return AUTO_DFLT_VAL, CARRY_FORWARD_ELIG_DPNT

```

## Sample Formula 3:

Formula Type : Default Enrollment

Description: Derive the default option when a dependent is aging out.

```

INPUTS ARE NUM_ELIG_DPNT
DEFAULT FOR BEN_OPT_NAME IS 'x'
DEFAULT FOR BEN_PLN_NAME IS 'x'
/* Initialization section */
AUTO_DFLT_VAL = 'N'
CARRY_FORWARD_ELIG_DPNT = 'CWP'
PREV_PRTT_ENRT_RSLT_ID='0'
l_actual_eff_date='1951/01/01 00:00:00' (date)
l_new_effective_date='1951/01/01 00:00:00' (date)
l_ee_opt_past='N'
l_sp_opt_past='N'
l_waive_opt_past='N'
l_fam_opt_past='N'
l_plan_name=BEN_PLN_NAME
l_opt_name=BEN_OPT_NAME
l_num_elig_dpnt=NUM_ELIG_DPNT
l_dflt_val='N'
l_cwp='NA'
l_status=123
l_prev_prtt_enrt_rslt_id='123'
l_log='X'
l_log=ess_log_write('Entering R_DFLT_ENRT_DEV_RL')
l_actual_eff_date=GET_CONTEXT(EFFECTIVE_DATE,to_date('1951/01/01 00:00:00'))
l_new_effective_date=ADD_DAYS(l_actual_eff_date,-1)
CHANGE_CONTEXTS(EFFECTIVE_DATE=l_new_effective_date)
(
l_prev_prtt_enrt_rslt_id=ben_fn_get_char_value('BEN_PRTT_ENRT_RSLT','PRTT_ENRT_RSLT_ID',l_plan_name, 'NA')
l_ee_opt_past=ben_fn_get_char_value('BEN_PRTT_ENRT_RSLT','ENROLLED',l_plan_name, 'KA_MED_HDHP_EMP')
l_sp_opt_past=ben_fn_get_char_value('BEN_PRTT_ENRT_RSLT','ENROLLED',l_plan_name, 'KA_MED_HDHP_SP')
l_fam_opt_past=ben_fn_get_char_value('BEN_PRTT_ENRT_RSLT','ENROLLED',l_plan_name, 'KA_MED_HDHP_FAM')

```

```

l_waive_opt_past=ben_fn_get_char_value('BEN_PRTT_ENRT_RSLT','ENROLLED',l_plan_name, 'KA_MED_HCHP_WAIVE')
)/* end change contexts */
l_status = add_rlog (-1 , -1 , 'R_DFLT_ENRT_DEV_RL l_plan_name='||l_plan_name)
l_status = add_rlog (-1 , -1 , 'R_DFLT_ENRT_DEV_RL l_opt_name='||l_opt_name)
l_status = add_rlog (-1 , -1 , 'R_DFLT_ENRT_DEV_RL l_num_elig_dpnt='||to_char(l_num_elig_dpnt))
l_log=ess_log_write(' l_plan_name='||l_plan_name)
l_log=ess_log_write(' l_opt_name='||l_opt_name)
l_log=ess_log_write(' l_num_elig_dpnt='||to_char(l_num_elig_dpnt))
/* process further only if there is an enrollment in the plan in the previous life event */
if ( l_ue_opt_past = 'Y'
OR l_sp_opt_past = 'Y'
OR l_fam_opt_past = 'Y'
OR l_waive_opt_past = 'Y' ) then
(
if ( l_opt_name = 'KA_MED_HDHP_FAM' ) then
(
l_status = add_rlog (-1 , -1 , 'R_DFLT_ENRT_DEV_RL KA_MED_HDHP_FAM')
l_log=ess_log_write(' KA_MED_HDHP_FAM ')
/* family will be the default option if
1) the employee is enrolled in the family option in the previous life event and the num_elig_dpnt is greater
than 2
*/
if ( l_fam_opt_past='Y' and l_num_elig_dpnt>=2 ) then
(
l_dflt_val = 'Y'
l_cfwp = 'CFWP'
)
)
else if (l_opt_name = 'KA_MED_HDHP_SP') then
(
l_status = add_rlog (-1 , -1 , 'R_DFLT_ENRT_DEV_RL KA_MED_HDHP_SP')
l_log=ess_log_write(' KA_MED_HDHP_SP ')
/* spouse will become the default option if
1) the employee is enrolled in spouse in the previous life event
2) the employee is enrolled into family in the previous life event but has only one num_elig_dpnt indicating
a
dependent has aged out
*/
if ( (l_sp_opt_past = 'Y' and l_num_elig_dpnt=1)
OR ( l_fam_opt_past='Y' and l_num_elig_dpnt=1) ) then
(
l_dflt_val = 'Y'
l_cfwp = 'CFWP'
)
)
else if (l_opt_name = 'KA_MED_HDHP_EMP') then
(
l_status = add_rlog (-1 , -1 , 'R_DFLT_ENRT_DEV_RL KA_MED_HDHP_EMP')
l_log=ess_log_write(' KA_MED_HDHP_EMP ')
/* employee will become the default option if
1) the employee is enrolled in employee-option in the previous life event
2) the employee is enrolled in the spouse option in the previous life event and there is divorce event */
if (l_ue_opt_past='Y') then
(
l_dflt_val = 'Y'
l_cfwp = 'CFWP'
)
else if (l_sp_opt_past = 'Y' and l_num_elig_dpnt=0) then
(
/* the spouse loses eligibility for some reason then emp will be the default*/
l_dflt_val = 'Y'
l_cfwp = 'CFWP'
)
)
else if (l_opt_name = 'KA_MED_HCHP_WAIVE') then
(

```

```

l_status = add_rlog (-1 , -1 , 'R_DFLT_ENRT_DEV_RL KA_MED_HCHP_WAIVE')
l_log=ess_log_write(' KA_MED_HCHP_WAIVE ')
/* waive will become the default option if
1) the employee is enrolled in the waive option in the previous life event */
if ( l_waive_opt_past = 'Y') then
(
l_dflt_val = 'Y'
l_cfwp = 'CFWP'
)
)
) /* end if lee or l_sp or l_fam or l_waive*/
PREV_PRTT_ENRT_RSLT_ID=l_prev_prtt_enrt_rslt_id
l_status = add_rlog (-1 , -1 , 'R_DFLT_ENRT_DEV_RL AUTO_DFLT_VAL='||l_dflt_val)
l_status = add_rlog (-1 , -1 , 'R_DFLT_ENRT_DEV_RL CARRY_FORWARD_ELIG_DPNT='||l_cfwp)
l_log=ess_log_write(' AUTO_DFLT_VAL '||l_dflt_val)
l_log=ess_log_write(' CARRY_FORWARD_ELIG_DPNT '||l_cfwp)
l_log=ess_log_write(' PREV_PRTT_ENRT_RSLT_ID '||l_prev_prtt_enrt_rslt_id)
l_log=ess_log_write('Leaving R_DFLT_ENRT_DEV_RL ')
AUTO_DFLT_VAL=l_dflt_val
CARRY_FORWARD_ELIG_DPNT=l_cfwp
return AUTO_DFLT_VAL, CARRY_FORWARD_ELIG_DPNT, PREV_PRTT_ENRT_RSLT_ID

```

## Evaluate Life Event

This formula type can be used to determine whether a life event should be evaluated or not. You could also alter the occurred date based on your use case using this formula.

### Contexts

The following contexts are available to formulas of this type:

- BUSINESS\_GROUP\_ID ( ENTERPRISE\_ID)
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- PERSON\_ID
- ORGANIZATION\_ID
- LER\_ID

### Database Items

Use only the following database items that are available to formulas of this type:

- All columns except attribute columns from tables: ben\_ler\_f
- Database items based on person's assignment – HR\_ASSIGNMENT\_ID AND PERSON\_ID

### Input Variables

Input variables are available to formulas of this type.

### **Input variables table**

Input	Data Type	Required	Description
BEN_PPL_IV_LF_EVT_OCRD_DT	Date		Life event Occurred Date passed in format YYYY/MM/DD HH24:MI:SS
BEN_PPL_IV_PTNL_LER_FOR_PER_STAT_CD	Char		Potential Life event Status Code
BEN_PPL_IV_NTFN_DT	Date		Notified Date in format YYYY/MM/DD HH24:MI:SS
BEN_PPL_IV_DTCTD_DT	Date		Detected Date in format YYYY/MM/DD HH24:MI:SS

## Return Variables

Use predefined names for return variables. The following return variables are available to formulas of this type.

### **Return variables table**

Return Value	Data Type	Required	Description
LIFE_EVENT_OCCURRED_DATE	Date	N	Replaces the original life event occurred date. Format of return value YYYY/MM/DD HH24:MI:SS
LIFE_EVENT_HAPPENED	Char	Y	Y or N. If any other value is assigned to this variable, the process raises an error BEN_92143_LIFE_EVENT_HAPPENED
LIFE_EVENT_NOTIFICATION_DATE	Date	N	Notification date
LIFE_EVENT_VOIDED_DATE	Date	N	Voided date
LIFE_EVENT_MANUAL_DATE	Date	N	Manual date
LIFE_EVENT_STATUS_CODE	Char	N	Status of the life event
LIFE_EVENT_DETECTED_DATE	Date	N	Detected date

## Errors

If any other output value is returned then the participation process errors with BEN\_91329\_FORMULA\_RETURN.

### Sample Formula 1:

FORMULA NAME: BEN\_EVAL\_EVENT

FORMULA TYPE: Evaluate Life Event

DESCRIPTION: If a life event is notified after 30 days ignore this potential life event.

```
Inputs are BEN_PPL_IV_LF_EVT_OCRD_DT (TEXT), BEN_PPL_IV_PTNL_LER_FOR_STAT_CD (TEXT), BEN_PPL_IV_NTFN_DT (TEXT)
default for BEN_PPL_IV_PTNL_LER_FOR_STAT_CD is 'aaa' default for BEN_PPL_IV_NTFN_DT is '1951/01/01 00:00:00'
default for BEN_PPL_IV_LF_EVT_OCRD_DT is '1951/01/01 00:00:00'
life_event_happened = 'Y'
l_lf_evt_dt = to_date(BEN_PPL_IV_LF_EVT_OCRD_DT, 'YYYY/MM/DD HH24:MI:SS') l_ntfd_dt =
to_date(BEN_PPL_IV_NTFN_DT, 'YYYY/MM/DD HH24:MI:SS') l_stat_cd = BEN_PPL_IV_PTNL_LER_FOR_STAT_CD
l_lf_evt_dt_30 = add_days(l_lf_evt_dt, 30)
if ((l_ntfd_dt > l_lf_evt_dt_30) and l_stat_cd = 'DTCTD' )
then life_event_happened = 'N'
return life_event_happened
```

## Sample Formula 2:

FORMULA NAME: BEN\_EVAL\_EVENT2

FORMULA TYPE: Evaluate Life Event

DESCRIPTION: If a life event is notified after 90 days for a set of people, ignore them.

```
Inputs are BEN_PPL_IV_LF_EVT_OCRD_DT (TEXT),
BEN_PPL_IV_PTNL_LER_FOR_PER_STAT_CD (TEXT),
BEN_PPL_IV_NTFN_DT (TEXT)
default for BEN_PPL_IV_PTNL_LER_FOR_PER_STAT_CD is 'AA
` default for BEN_PPL_IV_NTFN_DT is '01-JAN-1951' default for
PER_ASG_BARGAINING_UNIT_CODE is 'XYZ'
l_lf_evt_dt = to_date(BEN_PPL_IV_LF_EVT_OCRD_DT, 'YYYY/MM/DD HH24:MI:SS') l_ntfd_dt =
to_date(BEN_PPL_IV_NTFN_DT, 'YYYY/MM/DD HH24:MI:SS') l_stat_cd = BEN_PPL_IV_PTNL_LER_FOR_PER_STAT_CD
l_bar_unit = PER_ASG_BARGAINING_UNIT_CODE
l_lf_evt_dt_90 = add_days(l_lf_evt_dt, 90)
if ((l_ntfd_dt > l_lf_evt_dt_90) and l_stat_cd = 'DTCTD'
and ( l_bar_unit = 'Sales Unit' or l_bar_unit = 'Support Unit' )) then
(life_event_happened = 'N')
else
(life_event_happened = 'Y')
return life_event_happened
```

## Sample Formula 3:

FORMULA TYPE : Evaluate Life Event

DESCRIPTION: Trigger a different life event when processing length of service.

```
LIFE_EVENT_REASON_ID=0
LIFE_EVENT_HAPPENED='Y'
l_new_ler_id = 100010026711792
LIFE_EVENT_REASON_ID=l_new_ler_id
/* Return Six Months Complete Life Event instead */
return LIFE_EVENT_REASON_ID,LIFE_EVENT_HAPPENED
```

## Waiting Period Value and UOM

Participants need to complete a waiting period before they can gain eligibility to a benefits object. Sometimes employees have to wait for six months to get eligibility to a benefits object. Use the waiting period value formula type if the waiting period varies from employee to employee based on their organization or for rehired employees.

## Contexts

The following contexts are available to formulas of this type:

- BUSINESS\_GROUP\_ID ( ENTERPRISE\_ID)
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- PERSON\_ID
- LER\_ID
- ORGANIZATION\_ID
- JURISDICTION\_CODE
- PGM\_ID
- PL\_ID
- PL\_TYP\_ID
- OPT\_ID

## Database Items

Use only the following database items that are available to formulas of this type:

- All columns except attribute columns from tables: ben\_ler\_f, ben\_pl\_f, ben\_pgm\_f, ben\_pl\_typ\_f
- Database items based on person's assignment – HR\_ASSIGNMENT\_ID AND PERSON\_ID

## Input Variables

Input variables are not available to formulas of this type.

## Return Variables

Use predefined names for return variables. The following return variables are available to formulas of this type.

### ***Return variables table***

Return Value	Data Type	Required	Description
WAIT_PERD_VAL	Number	Y	Waiting period value
WAIT_PERD_UOM	Char	Y	Return value should be one of the codes: DY, WK, MO, QTR, YR

## Errors

If any other output value is returned then the participation process errors with BEN\_91329\_FORMULA\_RETURN.

## Sample Formula 1:

FORMULA NAME: BEN\_WAIT\_VAL\_UOM

FORMULA TYPE: Waiting period value and UOM

DESCRIPTION: Determine waiting periods based on org and country of birth.

```
default for PER_ASG_COUNTRY_OF_BIRTH is 'XX' default for PER_ASG_JOB_NAME is 'ABC'  
l_cntry = PER_ASG_COUNTRY_OF_BIRTH  
l_job = PER_ASG_JOB_NAME  
if ( l_cntry <> 'US' and l_job = 'Content Writer' ) then  
  (WAIT_PERD_VAL = 6  
  WAIT_PERD_UOM = 'MO' )  
Else  
  (WAIT_PERD_VAL = 1  
  WAIT_PERD_UOM = 'MO' )  
return WAIT_PERD_VAL, WAIT_PERD_UOM
```

## Coverage Amount Limit

Use the Coverage amount limit formula type if coverage restrictions have to be applied.

### Contexts

The following contexts are available to formulas of this type:

- BUSINESS\_GROUP\_ID ( ENTERPRISE\_ID)
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- PERSON\_ID

### Database Items

Use only the following database items that are available to formulas of this type:

- Database items based on person's assignment – HR\_ASSIGNMENT\_ID AND PERSON\_ID

### Input Variables

Input variables are not available to formulas of this type.

### Return Variables

Use predefined names for return variables. The following return variables are available to formulas of this type.

### Return variables table

Return Value	Data Type	Required	Description
L_MN_CVG_RQD_AMT, L_MX_CVG_ALWD_AMT, L_MX_CVG_WCFN_AMT, L_MX_CVG_INCR_ALWD_AMT, L_MX_CVG_INCR_WCF_ALWD_AMT	Number	Y	Minimum coverage amount
			Max coverage amount
			Max with certification amount
			Max coverage increment
			Max coverage increment with certification

## Errors

If any other output value is returned then the participation process errors with BEN\_91329\_FORMULA\_RETURN.

### Sample Formula 1:

FORMULA NAME: Coverage\_Amount\_Limit

FORMULA TYPE: Coverage Amount Limit

DESCRIPTION: Determine coverage amount limit values.

```

DEFAULT FOR CMP_ASSIGNMENT_SALARY_AMOUNT IS 0
DEFAULT FOR CMP_ASSIGNMENT_SALARY_ANNUALIZATION_FACTOR IS 0
L_MN_CVG_RQD_AMT = 0
L_MX_CVG_ALWD_AMT = 0
L_MX_CVG_WCFN_AMT = 300000
L_MX_CVG_INCR_ALWD_AMT = 10000
L_MX_CVG_INCR_WCF_ALWD_AMT = 10000
IF ( 3 * CMP_ASSIGNMENT_SALARY_AMOUNT *
CMP_ASSIGNMENT_SALARY_ANNUALIZATION_FACTOR ) > 200000
THEN
( L_MX_CVG_ALWD_AMT = 200000 )
ELSE
( L_MX_CVG_ALWD_AMT = 3 * CMP_ASSIGNMENT_SALARY_AMOUNT *
CMP_ASSIGNMENT_SALARY_ANNUALIZATION_FACTOR )
Return L_MN_CVG_RQD_AMT, L_MX_CVG_ALWD_AMT, L_MX_CVG_WCFN_AMT, L_MX_CVG_INCR_ALWD_AMT,
L_MX_CVG_INCR_WCF_ALWD_AMT

```

## Rate Periodization

This formula gives complete control for users to divide the rate values into periods. User can define their one rule using the contexts and input values available to distribute the rates among defined amounts, annual amounts, and communicated amounts.

## Contexts

The following contexts are available to formulas of this type:

- BUSINESS\_GROUP\_ID (ENTERPRISE\_ID)
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- PERSON\_ID
- LER\_ID
- PGM\_ID
- PL\_ID
- PL\_TYP\_ID

## Database Items

Use only the following database items that are available to formulas of this type:

- Database items based on person's assignment: HR\_ASSIGNMENT\_ID AND PERSON\_ID

## Input Variables

Input variables are available to this formula type.

### ***Input variables table***

Name	Data Type	Remarks
BEN_IV_CONVERT_FROM	Char	Possible values are DEFINED, CMCD, and ANNUAL.  If BEN_IV_CONVERT_FROM is DEFINED, then the rule needs to compute the annual and communicated values using the input value BEN_IV_CONVERT_FROM_VAL.  If BEN_IV_CONVERT_FROM is ANNUAL, then the rule needs to compute the defined and communicated values using the input value BEN_IV_CONVERT_FROM_VAL.
BEN_IV_CONVERT_FROM_VAL	Number	Value passed to the formula

## Return Variables

Use predefined names for return variables. The following return variables are available to formulas of this type.

### Return variables table

Out Variable name	Data Type	Remarks
DFND_VAL	Number	mandatory out variable
ANN_VAL	Number	mandatory out variable
CMCD_VAL	Number	mandatory out variable

## Errors

If any other output value is returned then the participation process errors with BEN\_91329\_FORMULA\_RETURN.

### Sample Formula 1:

FORMULA Name : RATE\_PERIODIZATION\_1

FORMULA TYPE : Rate Periodization

```

Inputs are BEN_IV_CONVERT_FROM (text), BEN_IV_CONVERT_FROM_VAL (text)
DEFAULT_DATA_VALUE FOR BEN_PEN_PL_NAME_TN is 'My-Default'
DEFAULT_DATA_VALUE FOR BEN_PEN_ENRT_CVG_STRT_DT_DN is '1951/01/01 0:00:00' (date)
DEFAULT_DATA_VALUE FOR BEN_PEN_ORGNL_ENRT_DT_DN is '1951/01/01 0:00:00' (date)
DEFAULT FOR BEN_YRP_START_DATE is '1951/01/01 0:00:00' (date) DEFAULT_DATA_VALUE FOR BEN_PEN_OPT_NAME_TN is
'My-Default' DEFAULT_DATA_VALUE FOR BEN_PEN_BNFT_AMT_NN is 0
1_eff_date='1951/01/01 0:00:00' (date)
1_new_eff_date='1951/01/01 0:00:00' (date) 1_enrt_cvg strt char='NA'
1_enrt_cvg strt_dt='1951/01/01 0:00:00' (date)
1_ori_enrt_cvg strt_dt='1951/01/01 0:00:00' (date)
1_chk_date='1951/01/01 0:00:00' (date)
1_yr_perd_st_date='1951/01/01 0:00:00' (date)
1_prev_bnft_amt_chr='N'
1_bnft_amt=0
1_prev_bnft_amt=0
1_prev_cntbn=0
DFND_VAL=0
ANN_VAL=0
CMCD_VAL=0
i=1
j=1
1_total_tp_in_cal_year=1
1_pay_perd_totper_rem=99
1_pay_perd=1
1_pay_cur_perd=1
/* input values */
1_iv_period=BEN_IV_CONVERT_FROM
1_iv_val=to_number(BEN_IV_CONVERT_FROM_VAL)
/* Main () */
j=1
/* change contexts for accommodating waiting periods or future rate, coverage start dates */
1_eff_date=GET_CONTEXT(EFFECTIVE_DATE, to_date('1951/01/01 0:00:00'))
1_new_eff_date=add_days(1_eff_date,15)
CHANGE_CONTEXTS(EFFECTIVE_DATE=1_new_eff_date)
/* get the benefit amount and the enrollment coverage start date within this block. Replace plan and option
names below */
while (BEN_PEN_PL_NAME_TN.exists(j)) loop (
if ((BEN_PEN_PL_NAME_TN[j]='ABC_PT1_PLAN1') and
(BEN_PEN_OPT_NAME_TN[j]='ABC_PT1_PLAN1_OPT1')) then
(
1_bnft_amt=BEN_PEN_BNFT_AMT_NN[j]

```

```

l_enrt_cvg strt_dt=BEN_PEN_ENRT_CVG_STRT_DT_DN[j] l_ori_enrt_cvg strt_dt=BEN_PEN_ORGNL_ENRT_DT_DN[j] )
j=j+1
) /*end loop plan option */
) /*end change contexts */
j=1
l_new_eff_date=add_days(l_eff_date,-1)
CHANGE_CONTEXTS(EFFECTIVE_DATE=l_new_eff_date)
/* get the previous benefit amount within this block. Replace plan and option names below. */
l_prev_bnft_amt_chr=BEN_FN_GET_CHAR_VALUE( 'BEN_PRTT_ENRT_RSLT'
,'BNFT_AMT'
,'ABC_PT1_PLAN1'
,'ABC_PT1_PLAN1_OPT1'
)
if ( l_prev_bnft_amt_chr <> ' ' and l_prev_bnft_amt_chr <> 'NO_DATA_FOUND' and
l_prev_bnft_amt_chr <> 'N' ) then
(l_prev_bnft_amt=to_number(l_prev_bnft_amt_chr))
) /*end change contexts */
IF (l_bnft_amt = 0 and l_iv_period = 'ANNUAL' ) THEN
(
l_bnft_amt = l_iv_val
)
/* check if enrolled earlier in same year */
IF (l_ori_enrt_cvg strt_dt <= l_enrt_cvg strt_dt) THEN
(
l_chk_date = l_ori_enrt_cvg strt_dt
)
ELSE
(
l_chk_date = l_enrt_cvg strt_dt
)
/* substitute the program id in below clause to pick year period start date */
CHANGE_CONTEXTS(PGM_ID = 300100083634356,EFFECTIVE_DATE=l_new_eff_date)
(
l_yr_perd_st_date = BEN_YRP_START_DATE
)
/* If enrolled in prior year, use this year period start date */ IF ( l_chk_date < l_yr_perd_st_date ) THEN
(
l_chk_date = l_yr_perd_st_date
)
l_pay_perd= to_number(BEN_FN_GET_CHAR_VALUE( 'PAY_TIME_PERIODS'
,'PERIOD_NUM'
,''
,''
,'START_DATE'
,to_char(l_chk_date)
))
/* Get the current running pay periods by passing the effective date */
l_pay_cur_perd= to_number(BEN_FN_GET_CHAR_VALUE( 'PAY_TIME_PERIODS'
,'PERIOD_NUM'
,''
,''
,'START_DATE'
,to_char(l_new_eff_date)
))
l_total_tp_in_cal_year = to_number(BEN_FN_GET_CHAR_VALUE( 'PAY_TIME_PERIODS'
,'TOTAL_TP_IN_CALENDAR_YEAR'
,''
,''
,'START_DATE'
,to_char(l_chk_date)
))
l_pay_perd_totper_rem =l_total_tp_in_cal_year - l_pay_cur_perd +1
IF (l_prev_bnft_amt <> 0 ) THEN
(
l_prev_cntbn = (l_prev_bnft_amt/(l_total_tp_in_cal_year-l_pay_perd+1))*(l_pay_cur_perd - l_pay_perd )
)

```

```
ELSE
(
  l_prev_cntbn = 0
)
CMCD_VAL = (l_bnft_amt - l_prev_cntbn )/l_pay_perd_totper_rem DFND_VAL = (CMCD_VAL *
  l_pay_perd_totper_rem)/12 ANN_VAL = l_bnft_amt
RETURN DFND_VAL,ANN_VAL,CMCD_VAL
```

## Dependent Certification Required

This formula can be used to determine the conditions under which a person must provide certification for his or her designated dependents.

### Contexts

The following contexts are available to formulas of this type:

- BUSINESS\_GROUP\_ID ( ENTERPRISE\_ID)
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- PERSON\_ID
- LER\_ID
- PGM\_ID
- PL\_ID
- PL\_TYP\_ID

### Database Items

Use only the following database items that are available to formulas of this type:

- Database items based on person's assignment:HR\_ASSIGNMENT\_ID AND PERSON\_ID

### Input Variables

The following input variable is available.

#### ***Input variables table***

Input	Data Type	Required	Description
CON_PERSON_ID	NUMBER	N	Dependent personID

### Return Variables

Use predefined names for return variables.

### Return variables table

Out Variable name	Data Type	Remarks
L_OUTPUT	Varchar2	Y or N

## Errors

If any other output value is returned then the participation process errors with BEN\_91329\_FORMULA\_RETURN.

### Sample Formula 1:

FORMULA NAME: BEN\_DPNT\_CERT

FORMULA TYPE: Dependent Certification Required

Returned Value: Returned l\_return is a Y/N value.

```

/
inputs are CON_PERSON_ID(number)
default for BEN_LER_NAME is 'NA'
l_ler_name=BEN_LER_NAME
l_dpnt_id=CON_PERSON_ID
l_return='Y'
l_chk_dpnt_ctfn='N'
l_chk_dpnt_ctfn2='N'
l_eff_date='1951/01/01 00:00:00' (date)
l_context_date='1951/01/01 00:00:00' (date)
l_eff_date = GET_CONTEXT(EFFECTIVE_DATE,to_date('1951/01/01 00:00:00') )
l_context_date=add_days(l_eff_date,-1)
change_contexts(EFFECTIVE_DATE=l_context_date)
(
if (l_ler_name = 'Open') then
(
l_chk_dpnt_ctfn=BEN_FN_GET_CHAR_VALUE(
'BEN_CVRD_DPNT_CTFN_PRVDD'
,'PROVIDED'
,'PlanA'
,'OptionA'
,to_char(l_dpnt_id)
,'Birth certificate'
)
l_chk_dpnt_ctfn2=BEN_FN_GET_CHAR_VALUE(
'BEN_CVRD_DPNT_CTFN_PRVDD'
,'PROVIDED'
,'PlanB'
,'OptionB'
,to_char(l_dpnt_id)
,'Birth certificate'
)
if (l_chk_dpnt_ctfn='Y' or l_chk_dpnt_ctfn2='Y' ) then (l_return='N')
)
)
return l_return

```

### Sample Formula 2:

FORMULA TYPE: Dependent Certification Required

```
DEFAULT FOR BEN_LER_NAME is 'Default'
```

```

l_ler_name = BEN_LER_NAME
l_cert_rqrd = 'Y'
/* Do not trigger certifications for Open/Salary Change life events
*/
IF ( l_ler_name = 'Open'
or l_ler_name = 'Salary Change'
)
then
(
l_cert_rqrd = 'N'
)
return l_cert_rqrd

```

## Sample Formula 3:

FORMULA TYPE: Dependent Certification Required

```

inputs are CON_PERSON_ID(number)
l_dpnt_id=CON_PERSON_ID
l_return='Y'
l_chk_dpnt_ctfn='N'
/* Check if the certificate is already provided in the past.
Possible values for the certificates are
1. Adoption certificate
2. Birth certificate
3. Domestic partner certificate
4. Legal custody certificate
5. Marriage certificate
*/
l_chk_dpnt_ctfn=BEN_FN_GET_CHAR_VALUE(
'BEN_CVRD_DPNT_CTFN_PRVDD'
,'PROVIDED'
,'NA_CHECK_ONCE'
,'NA'
,to_char(l_dpnt_id)
,'Marriage certificate'
)
if (l_chk_dpnt_ctfn='Y') then
(l_return='N')
return l_return

```

## Length of Service Date to Use

This formula type is used to determine the date that will be used in the calculation of length of service.

### Contexts

The following contexts are available to formulas of this type:

- BUSINESS\_GROUP\_ID ( ENTERPRISE\_ID)
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- PGM\_ID
- PL\_ID
- PER\_IN\_LER\_ID

- LER\_ID
- BENEFIT\_RELATION\_ID
- PL\_TYP\_ID
- OPT\_ID
- ORGANIZATION\_ID

## Database Items

Use only the following database items that are available to formulas of this type:

- Database items based on person's assignment: HR\_ASSIGNMENT\_ID AND PERSON\_ID

## Input Variables

The following input variables are available:

### ***Input variables table***

Input	Data Type	Required	Description
BEN_IV_PERSON_ID	Number	N	PersonID of the participant
BEN_IV_RT_STRT_DT	Date	N	Rate start date
BEN_IV_CVG_STRT_DT	Date	N	Coverage start date

## Return Variables

The following return variables are available to formulas of this type. Any name can be used.

### ***Return variables table***

Out Variable Name	Data Type	Remarks
L_OUTPUT	Date	Date to be used while calculating the length of service

## Sample Formula 1:

FORMULA TYPE: Length of Service Date to Use

Description: Use the date stored in the Descriptive flex field.

```

DEFAULT FOR PER_ASG_ATTRIBUTE_DATE3 IS '1951/01/01 00:00:00' (DATE)
DEFAULT FOR PER_REL_ORIGINAL_DATE_OF_HIRE IS '1951/01/01 00:00:00' (DATE)
l_return = '1951/01/01 00:00:00' (DATE)
IF (PER_ASG_ATTRIBUTE_DATE3 WAS NOT DEFAULTED) THEN
(
l_return = PER_ASG_ATTRIBUTE_DATE3
)
ELSE
(

```

```
l_return = PER_REL_ORIGINAL_DATE_OF_HIRE
)
RETURN l_return
```

## Sample Formula 2:

FORMULA TYPE: Length of Service Date to Use

Description: Use the hire/adjusted service date.

```
DEFAULT FOR ACP_HIRE_DATE IS '1951/01/01 00:00:00' (date)
DEFAULT FOR PER_ASG_REL_ADJUSTED_SVC_DATE IS '1951/01/01 00:00:00' (date)
l_pps_date = ACP_HIRE_DATE
l_return = l_pps_date
IF PER_ASG_REL_ADJUSTED_SVC_DATE WAS NOT DEFAULTED THEN
(
l_adjusted_svc_dt = PER_ASG_REL_ADJUSTED_SVC_DATE
if (l_adjusted_svc_dt < l_pps_date) then
(
l_return = l_adjusted_svc_dt
)
)
RETURN l_return
```

## Enrollment Certification Required

This formula type is used to determine if a certification is required for enrollment.

### Contexts

The following contexts are available:

- BUSINESS\_GROUP\_ID ( ENTERPRISE\_ID)
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- PGM\_ID
- PL\_ID
- PER\_IN\_LER\_ID
- LER\_ID
- BENEFIT\_RELATION\_ID
- PL\_TYP\_ID
- OPT\_ID
- ORGANIZATION\_ID
- ELIG\_PER\_ELCTBL\_CHC\_ID
- ENRT\_CTFN\_TYP\_CD

## Database Items

Use only the following database items that are available to formulas of this type:

- Database items based on person's assignment: HR\_ASSIGNMENT\_ID AND PERSON\_ID

## Input Variables

Input variables are not available to formulas of this type.

## Return Variables

The following return variables are available to formulas of this type. Any name can be used for the return variable.

### ***Return variables table***

Out Variable Name	Data Type	Remarks
L_OUTPUT	Varchar	Y/N

## Sample Formula 1:

FORMULA TYPE: Enrollment Certification Required

Description: Trigger certification for self-assigned life events.

```
DEFAULT FOR BEN_PLN_NAME IS 'X'  
DEFAULT FOR BEN_OPT_NAME IS 'X'  
DEFAULT FOR BEN_LER_NAME is 'XXX'  
l_return='Y'  
l_emp_enrd='N'  
l_lf_evt_ocrd_dt = GET_CONTEXT(EFFECTIVE_DATE,to_date('1951/01/01 00:00:00'))  
l_self_assign_event = 'N'  
l_first_event = 'N'  
l_ler_name = BEN_LER_NAME  
l_pl_name = BEN_PLN_NAME  
l_opt_name = BEN_OPT_NAME  
IF  
(  
  l_ler_name = 'Marriage' OR  
  l_ler_name = 'Birth' OR  
  l_ler_name = 'Adoption' OR  
  l_ler_name = 'Divorce' OR  
  l_ler_name = 'Change Beneficiary' OR  
  l_ler_name = 'Death of a Dependent'  
) THEN (l_self_assign_event = 'Y')  
l_day_before = add_days(l_lf_evt_ocrd_dt,-1)  
change_contexts(EFFECTIVE_DATE=l_day_before)  
(  
l_emp_enrd = ben_fn_get_char_value('BEN_PRTT_ENRT_RSLT', 'ENROLLED',l_pl_name,l_opt_name)  
)  
if ( (l_emp_enrd = 'Y')  
or (l_ler_name like '%System Implementation%')  
or (l_self_assign_event = 'N' )  
) then  
(  
l_return = 'N'  
)
```

```
return l_return
```

## Sample Formula 2:

FORMULA TYPE : Enrollment Certification Required

DESCRIPTION : Do not trigger certification for employee life when participant elects to decrement coverage.

```

DEFAULT FOR BEN_OPT_NAME is 'ABC'
DEFAULT_DATA_VALUE FOR BEN_PIL_LF_EVT_OCRD_DT_TN IS '1951/01/01 00:00:00' (date)
DEFAULT_DATA_VALUE FOR BEN_PIL_PER_IN_LER_STAT_CD_TN IS 'NA'
l_opt_name = BEN_OPT_NAME
i=1
l_cert_reqd ='Y'
l_eff_date = GET_CONTEXT(EFFECTIVE_DATE, to_date('1951/01/01 00:00:00') )
/* get the life date for the started event */
WHILE BEN_PIL_LF_EVT_OCRD_DT_TN.exists(i) LOOP
( IF (BEN_PIL_PER_IN_LER_STAT_CD_TN[i] ='STRTD') THEN
(
l_eff_date = BEN_PIL_LF_EVT_OCRD_DT_TN[i]
EXIT
)
i=i+1
) /* end while */
/* get the enrolled option in the past life event*/
CHANGE_CONTEXTS(EFFECTIVE_DATE=add_days(l_eff_date,-1))
(
l_opt1_prev = ben_fn_get_char_value('BEN_PRTT_ENRT_RSLT','ENROLLED','Supplemental Employee Life','1X Annual
Salary')
l_opt2_prev = ben_fn_get_char_value('BEN_PRTT_ENRT_RSLT','ENROLLED','Supplemental Employee Life','2X Annual
Salary')
l_opt3_prev = ben_fn_get_char_value('BEN_PRTT_ENRT_RSLT','ENROLLED','Supplemental Employee Life','3X Annual
Salary')
l_opt4_prev = ben_fn_get_char_value('BEN_PRTT_ENRT_RSLT','ENROLLED','Supplemental Employee Life','4X Annual
Salary')
l_opt5_prev = ben_fn_get_char_value('BEN_PRTT_ENRT_RSLT','ENROLLED','Supplemental Employee Life','5X Annual
Salary')
) /* end change contexts*/
if (l_opt_name = 'Waive') then
(
l_cert_reqd='N'
)
else if ( l_opt_name = '5X Annual Salary'
and l_opt5_prev = 'Y'
) then
(
l_cert_reqd='N'
)
else if ( l_opt_name = '4X Annual Salary'
and ( l_opt5_prev = 'Y'
OR l_opt4_prev = 'Y'
)
) then
(
l_cert_reqd='N'
)
else if ( l_opt_name = '3X Annual Salary'
and ( l_opt5_prev = 'Y'
OR l_opt4_prev = 'Y'
OR l_opt3_prev = 'Y'
)
) then
(
l_cert_reqd='N'
)

```

```
else if ( l_opt_name = '2X Annual Salary'  
and ( l_opt5_prev = 'Y'  
OR l_opt4_prev = 'Y'  
OR l_opt3_prev = 'Y'  
OR l_opt2_prev = 'Y'  
)  
) then  
l_cert_reqd='N'  
)  
else if ( l_opt_name = '1X Annual Salary'  
and ( l_opt5_prev = 'Y'  
OR l_opt4_prev = 'Y'  
OR l_opt3_prev = 'Y'  
OR l_opt2_prev = 'Y'  
OR l_opt1_prev = 'Y'  
)  
) then  
l_cert_reqd='N'  
)  
RETURN l_cert_reqd
```

## Coverage Upper Limit

This formula type determines the upper bound on the benefit amount that a participant can elect as part of a compensation object.

### Contexts

The following contexts are available:

- BUSINESS\_GROUP\_ID ( ENTERPRISE\_ID)
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- PGM\_ID
- PL\_ID
- PER\_IN\_LER\_ID
- LER\_ID
- BENEFIT\_RELATION\_ID
- PL\_TYP\_ID
- OPT\_ID
- ORGANIZATION\_ID
- ELIG\_PER\_ELCTBL\_CHC\_ID
- ENRT\_CTFN\_TYP\_CD

## Database Items

Use only the following database items that are available to formulas of this type:

- Database items based on person's assignment: HR\_ASSIGNMENT\_ID AND PERSON\_ID

## Input Variables

The following input variables are available:

### ***Input variables table***

Input	Data Type	Required	Description
BEN_IV_RT_STRT_DT	Date	N	Rate start date
BEN_IV_CVG_STRT_DT	Date	N	Coverage start date

## Return Variables

The following return variables are available to formulas of this type. Any name can be used for the return variable.

### ***Return variables table***

Out Variable Name	Data Type	Remarks
L_OUTPUT	Number	Coverage upper limit

## Sample Formula 1:

FORMULA TYPE : Coverage Upper Limit

```
DEFAULT FOR CMP_ASSIGNMENT_SALARY_ANNUAL_AMOUNT is 0
1_sal= CMP_ASSIGNMENT_SALARY_ANNUAL_AMOUNT
L_MX_CVG_ALWD_AMT = LEAST((5*1_sal),2000000)
RETURN L_MX_CVG_ALWD_AMT
```

## Rate Start Date

This formula type is used to derive the rate start date for the enrollment into the selected compensation object.

## Contexts

The following contexts are available:

- BUSINESS\_GROUP\_ID ( ENTERPRISE\_ID)
- EFFECTIVE\_DATE

- HR\_ASSIGNMENT\_ID
- PGM\_ID
- PL\_ID
- PER\_IN\_LER\_ID
- PERSON\_ID
- LER\_ID
- BENEFIT\_RELATION\_ID
- ELIG\_PER\_ELCTBL\_CHC\_ID

## Database Items

Use only the following database items that are available to formulas of this type:

- Database items based on person's assignment: HR\_ASSIGNMENT\_ID AND PERSON\_ID

## Input Variables

Input variables are not available to formulas of this type.

## Return Variables

The following return variables are available to formulas of this type. Any name can be used for the return variable.

### ***Return variables table***

Out Variable Name	Data Type	Remarks
L_OUTPUT	Date	Calculated rate start date

## Sample Formula 1:

FORMULA TYPE: Rate Start Date

DESCRIPTION: For (New Hire) rate starts first-of-month after the enrollment end.

```
DEFAULT for PER_ASG_REL_ORIGINAL_DATE_OF_HIRE is '1951/01/01 00:00:00' (date)
l_enrt_perd_end = '1951/01/01 00:00:00' (date)
l_rt strt_dt = '1951/01/01 00:00:00' (date)
l_hire_dt=PER_ASG_REL_ORIGINAL_DATE_OF_HIRE
l_enrt_perd_end = LAST_DAY(ADD_MONTHS(l_hire_dt,1))
l_rt strt_dt = ADD_DAYS(l_enrt_perd_end,1)
return l_rt strt_dt
```

## Sample Formula 2:

FORMULA TYPE: Rate Start Date

```
DEFAULT for PER_ASG_REL_ACTUAL_TERMINATION_DATE is '1951/01/01 00:00:00' (date)
l strt_dt = '1951/01/01 00:00:00' (date)
l_fc_dt = '1951/01/01 00:00:00' (date)
l_term_dt = PER_ASG_REL_ACTUAL_TERMINATION_DATE
```

```
l_fc_dt = ADD_DAYS(LAST_DAY(ADD_MONTHS(l_term_dt,1)),1)
RETURN l_fc_dt
```

## Enrollment Coverage Start Date

This formula type is used to derive the enrollment coverage start date.

### Contexts

The following contexts are available:

- BUSINESS\_GROUP\_ID ( ENTERPRISE\_ID)
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- PGM\_ID
- PL\_ID
- PL\_TYP\_ID
- OPT\_ID
- PER\_IN\_LER\_ID
- PERSON\_ID
- LER\_ID
- BENEFIT\_RELATION\_ID
- ELIG\_PER\_ELCTBL\_CHC\_ID
- ORGANIZATION\_ID

### Database Items

Use only the following database items that are available to formulas of this type:

- Database items based on person's assignment: HR\_ASSIGNMENT\_ID AND PERSON\_ID

### Input Variables

Input variables are not available to formulas of this type.

### Return Variables

The following return variables are available to formulas of this type. Any name can be used for the return variable.

#### ***Return variables table***

Out Variable Name	Data Type	Remarks
L_OUTPUT	Date	Calculated enrollment coverage start date

## Sample Formula 1:

Formula Type: Enrollment Coverage Start Date

DESCRIPTION: For (New Hire) hired between 1 and 15 of the month, coverage starts (FONM), and those between 15 and 30/31 coverage starts First of the month following next month

```
l_eff_date =GET_CONTEXT(EFFECTIVE_DATE, to_date('1951/01/01 00:00:00'))
l_rt_start_dt='1951/12/31 00:00:00'(date)
l_day = to_num(to_char(l_eff_date,'DD'))
If l_day >=1 and l_day <=15 THEN
(
l_rt_start_dt = ADD_DAYS(LAST_DAY(l_eff_date),1)
)
ELSE
(
l_rt_start_dt = ADD_MONTHS(ADD_DAYS(LAST_DAY(l_eff_date),1),1)
)
Return l_rt_start_dt
```

## Sample Formula 2:

FORMULA TYPE: Enrollment Coverage Start Date

DESCRIPTION: Coverage starts (life date + 1) in a (Death of Dependent) life event.

```
DEFAULT FOR BEN_LER_NAME IS 'NA'
DEFAULT FOR BEN_PIL_LF_DT is '1951/01/01 00:00:00'(date)
l_ler_name = BEN_LER_NAME
l_lf_dt = BEN_PIL_LF_DT
l_cvg strt_dt = '1951/01/01 00:00:00'(date)
IF l_ler_name = 'Death of Dependent'
THEN
(
l_cvg strt_dt = ADD_DAYS(l_lf_dt,1)
)
RETURN l_cvg strt_dt
```

## Default to Assign Pending Action

This formula type is used to identify an interim choice in case of a suspension.

### Contexts

The following contexts are available:

- BUSINESS\_GROUP\_ID ( ENTERPRISE\_ID)
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- PGM\_ID
- PL\_ID
- PL\_TYP\_ID
- OPT\_ID

- PERSON\_ID
- LER\_ID
- BENEFIT\_RELATION\_ID
- ELIG\_PER\_ELCTBL\_CHC\_ID
- ORGANIZATION\_ID

## Database Items

Use only the following database items that are available to formulas of this type:

- Database items based on person's assignment: HR\_ASSIGNMENT\_ID AND PERSON\_ID

## Input Variables

Input variables are available to formulas of this type.

### *Input variables table*

Input	Data Type	Required	Description
BEN_PEN_IV_PRTT_ENRT_RSLT_ID	Number	N	Result ID of the choice that is suspended

## Return Variables

The formula can return either:

- Interim code
- Interim choice
- Amount corresponding to the coverage along with an interim electable choice

Please see samples for illustration.

### *Return variables table*

Out Variable Name	Data Type	Remarks
L_OUTPUT	Character	Interim code which can be used arrive at the interim choice. Possible values are listed below. CDND : Current - default; new - default CDNL : Current - default; new - next lower CDNM : Current - default; new - minimum CDNN : Current - default; new - nothing CLND : Current - next lower; new - default

Out Variable Name	Data Type	Remarks
		CLNL : Current - next lower; new - next lower CLNM : Current - next lower; new - minimum CLNN : Current - next Lower, new - nothing CMND : Current - minimum; new - apply default CMNL : Current - minimum; new - next lower CMNM : Current - minimum; new - minimum CMNN : Current - minimum, new - nothing CNND : Current - nothing; new - default CNNL : Current - nothing; new - next lower CNNM : Current - nothing; new - minimum CNNN : Current - nothing; new - nothing CSND : Current - same; new - default CSNL : Current - same; new - next lower CSNM : Current - same; new - minimum CSNN : Current - same; new – nothing
L_BNFT_AMOUNT	NUMBER	Benefit amount
L_EPE_ID	NUMBER	Electable choice

## Sample Formula 1:

Formula Name : STU\_DEFAULT\_TO\_ASSIGN\_PENDING\_ACTION

Formula Type : Default to Assign Pending Action

```

DEFAULT_DATA_VALUE FOR BEN_EXT_RSLT_BNFT_AMT is 0
DEFAULT_DATA_VALUE FOR BEN_EXT_RSLT_PER_IN_LER_ID is 0
DEFAULT_DATA_VALUE FOR BEN_EXT_RSLT_PRTT_ENRT_RSLT_ID is 0
DEFAULT_DATA_VALUE FOR BEN_EXT_RSLT_PL_NAME is 'NA'
l_eff_dt = GET_CONTEXT(EFFECTIVE_DATE,'1900/01/01 00:00:00'(date))
l_bnft_amt_num_o=0
l_bnft_amt_num_n=0
l_pil_o=0
l_pil_n=0
l_prtt_o=0
l_prtt_n=0
l_pndg_actn='CLNL'
l_debug='NA'
/* main() */
/* get the old benefit amount and pil */
l_eff_dt=add_days(l_eff_dt,-1)
l_debug=ESS_LOG_WRITE(' l_eff_dt: '||to_char(l_eff_dt))
l_debug=ESS_LOG_WRITE(' l_iv_pil: '||to_char(l_iv_pil))
i=1
CHANGE_CONTEXTS(EFFECTIVE_DATE=l_eff_dt)
(

```

```

while (BEN_EXT_RSLT_PL_NAME.exists(i) )
loop
(
if (BEN_EXT_RSLT_PL_NAME[i] = 'STU3_Stuller Supplemental Life - Employee')
then
(
l_bnft_amt_num_o=BEN_EXT_RSLT_BNFT_AMT[i]
l_pil_o=BEN_EXT_RSLT_PER_IN_LER_ID[i]
l_prtt_o=BEN_EXT_RSLT_PRTT_ENRT_RSLT_ID[i]
) /* end if */
i=i+1 /* increment loop counter */
) /* end loop */
) /* end change contexts*/
l_debug=ESS_LOG_WRITE(' l_bnft_amt_num_o: '||to_char(l_bnft_amt_num_o))
l_debug=ESS_LOG_WRITE(' l_pil_o: '||to_char(l_pil_o))
l_debug=ESS_LOG_WRITE(' l_prtt_o: '||to_char(l_prtt_o))
/* get the new benefit amount and pil */
l_eff_dt=add_days(l_eff_dt,90)
l_debug=ESS_LOG_WRITE(' l_eff_dt: '||to_char(l_eff_dt))
i=1
CHANGE_CONTEXTS(EFFECTIVE_DATE=l_eff_dt)
(
while (BEN_EXT_RSLT_PL_NAME.exists(i) )
loop
(
if (BEN_EXT_RSLT_PL_NAME[i] = 'STU3_Stuller Supplemental Life - Employee')
then
(
l_bnft_amt_num_n=BEN_EXT_RSLT_BNFT_AMT[i]
l_pil_n=BEN_EXT_RSLT_PER_IN_LER_ID[i]
l_prtt_n=BEN_EXT_RSLT_PRTT_ENRT_RSLT_ID[i]
) /* end if */
i=i+1 /* increment loop counter */
) /* end loop */
) /* end change contexts*/
l_debug=ESS_LOG_WRITE(' l_bnft_amt_num_n: '||to_char(l_bnft_amt_num_n))
l_debug=ESS_LOG_WRITE(' l_pil_n: '||to_char(l_pil_n))
l_debug=ESS_LOG_WRITE(' l_prtt_n: '||to_char(l_prtt_n))
/* conditional logic */
if ( (l_bnft_amt_num_o > 150000)
and (l_bnft_amt_num_n > 150000)
) then
(
if ( l_pil_o <> l_iv_pil ) then
(l_pndg_actn = 'CLNL')
else
(l_pndg_actn ='CSENLNLL')
)
else if ( (l_bnft_amt_num_o < 150000)
and (l_bnft_amt_num_n > 150000)
) then
(
l_pndg_actn = 'CLNL'
)
return l_pndg_actn

```

## Sample Formula 2:

FORMULA Name : MMO\_INTERIM\_RULE1

FORMULA TYPE : Default to Assign Pending Action

```

DEFAULT_DATA_VALUE FOR BEN_EPE_ELIG_PER_ELCTBL_CHC_ID_NN is 0
DEFAULT_DATA_VALUE FOR BEN_EPE_PL_NAME_TN is 'NA'
DEFAULT_DATA_VALUE FOR BEN_EPE_OPT_NAME_TN is 'NA'

```

```

/* initialization */
L_ELIG_PER_ELCTBL_CHC_ID=345
i=1
/*main()*/
while (BEN_EPE_ELIG_PER_ELCTBL_CHC_ID_NN.exists(i)) loop
(
if ( (BEN_EPE_PL_NAME_TN[i] = 'MMO_SUPP_LIFE_P')
and (BEN_EPE_OPT_NAME_TN[i] = 'MMO_GUARANTEE_ISSUE' )
)
then
( L_ELIG_PER_ELCTBL_CHC_ID = BEN_EPE_ELIG_PER_ELCTBL_CHC_ID_NN[i]
)
/* increment the counter */
i=i+1
) /* end loop */
return L_ELIG_PER_ELCTBL_CHC_ID

```

## Sample Formula 3:

FORMULA NAME : AGIOS\_SUPP\_EMP\_LIFE\_INTERIM\_LIMIT\_FF

FORMULA TYPE : Default to Assign Pending Action

DESCRIPTION : Validates the Employee supplemental life increases for Increment without a certification as long as the increase does not exceed the guaranteed issue.

```

DEFAULT FOR BEN_PIL_LF_EVT_OCRD_DT is '1951/01/01 00:00:00' (date)
DEFAULT_DATA_VALUE FOR BEN_EPE_ELIG_PER_ELCTBL_CHC_ID_NN is 0
DEFAULT_DATA_VALUE FOR BEN_EPE_PL_NAME_TN is 'WAIVE'
DEFAULT_DATA_VALUE FOR BEN_EPE_CRNTLY_ENRD_FLAG_TN is 'N'
DEFAULT_DATA_VALUE FOR BEN_EPE_CVRD_FLAG_TN is 'N'
DEFAULT_DATA_VALUE FOR BEN_EPE_INTERIM_FLAG_TN is 'N'
DEFAULT_DATA_VALUE FOR BEN_EPE_OPT_NAME_TN is 'NA'
DEFAULT_DATA_VALUE FOR BEN_EPE_ELCTBL_FLAG_TN is 'NA'
DEFAULT_DATA_VALUE FOR PER_HIST_ASG_ASSIGNMENT_ID is 0
DEFAULT_DATA_VALUE FOR PER_HIST_ASG_EFFECTIVE_START_DATE is '1951/01/01 00:00:00' (date)
DEFAULT_DATA_VALUE FOR PER_HIST_ASG_EFFECTIVE_END_DATE is '1951/01/01 00:00:00' (date)
DEFAULT_DATA_VALUE FOR PER_HIST_ASG_ASSIGNMENT_TYPE is 'NA'
DEFAULT_DATA_VALUE FOR PER_HIST_ASG_STATUS_TYPE is 'NA'
DEFAULT_DATA_VALUE FOR PER_HIST_ASG_PRIMARY_FLAG is 'NA'
DEFAULT_DATA_VALUE FOR PER_HIST_ASG_EFFECTIVE_LATEST_CHANGE is 'NA'
Default for l_sal_amount is 0
Default for l_sal_fctr is 0
Default for l_max is 0
default for l_enrld_flag is 'Y'
L_MN_CVG_RQD_AMT = 0 /*Minimum coverage amount*/
L_MX_CVG_ALWD_AMT = 150000 /*Maximum coverage amount*/
L_MX_CVG_INCR_ALWD_AMT = 50000 /*Maximum coverage increment*/
L_MX_CVG_INCR_WCF_ALWD_AMT = 500000 /*Maximum coverage increment with certification*/
L_MX_CVG_WCFN_AMT = 500000 /*Maximum coverage amount with certification*/
l_prev_enrld_amt = '0'
l_prev_enrld_amt1 = '0'
l_prev_enrld_amt_num1 = 0
l_curr_enrld_amt = '0'
l_curr_enrld_amt_num1 = 0
l_new_effective_date = '1951/01/01 00:00:00' (date)
l_actual_eff_date = '1951/01/01 00:00:00' (date)
l_asg_id=1234
l_sal_amount=0
l_sal_fctr=0
l_max = 500000
l_sal_amount1=BEN_FN_GET_CHAR_VALUE(
'CMP_SALARY'
,'SALARY_AMOUNT'

```

```

)
l_sal_fctrl=BEN_FN_GET_CHAR_VALUE(
'CMP_SALARY'
,'SALARY_ANNUALIZATION_FACTOR'
)
IF l_sal_amount1 <> 'NO_DATA_FOUND' AND l_sal_amount1 <> '' AND l_sal_amount1 <> 'N'
THEN
(
l_sal_amount=to_num(l_sal_amount1)
)
ELSE
(
l_sal_amount = 0
)
IF l_sal_fctrl <> 'NO_DATA_FOUND' AND l_sal_fctrl <> '' AND l_sal_fctrl <> 'N'
THEN
(
l_sal_fctr=to_num(l_sal_fctrl)
)
ELSE
(
l_sal_fctr=0
)
IF ( 5 * l_sal_amount * l_sal_fctr ) > 500000
THEN
(
L_MX_CVG_ALWD_AMT = 500000
)
ELSE
(
L_MX_CVG_ALWD_AMT = trunc((5*l_sal_amount * l_sal_fctr)/10000)*10000
)
l_actual_eff_date = GET_CONTEXT(EFFECTIVE_DATE, to_date('1951/01/01 00:00:00'))
l_new_effective_date = ADD_DAYS (l_actual_eff_date, - 1)
l_log_data = ESS_LOG_WRITE ('Start of the formula after initial ')
ret = HR_TRACE('Start Debugging AGIOS_SUPP_EMP_LIFE_INTERIM_LIMIT_FF')
/* get Previous Coverage Amount - Change the contexts since coverage is starting as of the processing end
date */
CHANGE_CONTEXTS(EFFECTIVE_DATE = l_new_effective_date )
(
l_enrld_amtl = ben_fn_get_char_value( 'BEN_PRTT_ENRT_RSLT'
,'BNFT_AMT'
,'Supplemental Employee Life'
,'Elected'
)
l_enrld_flag = ben_fn_get_char_value(
'BEN_PRTT_ENRT_RSLT'
,'ENROLLED'
,'Supplemental Employee Life'
,'Elected')
l_log_data = ESS_LOG_WRITE (' Previous Supp Life Insured Coverage Amount : ' ||l_enrld_amtl)
IF (l_enrld_amtl <> '' AND l_enrld_amtl <> 'NO_DATA_FOUND' AND l_enrld_amtl <> 'N')
THEN
(
l_prev_enrld_amt_num1 = to_number(l_enrld_amtl)
)
ELSE
(
l_prev_enrld_amt_num1 = 0
)
)
l_log_data = ESS_LOG_WRITE (' Previous Supp Life Insured Coverage Amount l_prev_enrld_amt_num1: ' ||
to_char(l_prev_enrld_amt_num1))
l_new_effective_date = GET_CONTEXT(EFFECTIVE_DATE,to_date('1951/01/01 00:00:00'))
CHANGE_CONTEXTS(EFFECTIVE_DATE = ADD_DAYS(l_new_effective_date,120))
(

```

```

l_curr_enrld_amtl = ben_fn_get_char_value( 'BEN_PRTT_ENRT_RSLT'
,'BNFT_AMT'
,'Supplemental Employee Life'
,'Elected'
)
l_enrld_flag = ben_fn_get_char_value(
'BEN_PRTT_ENRT_RSLT'
,'ENROLLED'
,'Supplemental Employee Life'
,'Elected')
l_log_data = ESS_LOG_WRITE (' Current Supp Life Insured Coverage Amount : ' ||l_curr_enrld_amtl)
IF (l_curr_enrld_amtl <> ' ' AND l_curr_enrld_amtl <> 'NO_DATA_FOUND' AND l_curr_enrld_amtl <> 'N')
THEN
(
l_curr_enrld_amt_numl = to_number(l_prev_enrld_amtl)
)
ELSE
(
l_curr_enrld_amt_numl = 0
)
)
/* reassign l to i */
i=1
l_asg_id = GET_CONTEXT(HR_ASSIGNMENT_ID,1)
CHANGE_CONTEXTS(HR_ASSIGNMENT_ID=l_asg_id)
(
while (BEN_EPE_ELIG_PER_ELCTBL_CHC_ID_NN.exists(i)) loop
(
if ( (BEN_EPE_PL_NAME_TN[i]='Supplemental Employee Life')
and (BEN_EPE_OPT_NAME_TN[i]='Elected')
and (BEN_EPE_ELCTBL_FLAG_TN[i]='Y')
) then
(
/* this will hold the choice id of the (elected) option */
l_elig_per_elctbl_chc_id = BEN_EPE_ELIG_PER_ELCTBL_CHC_ID_NN[i]
)
i=i+1
)
)
if (l_curr_enrld_amt_numl > 150000)
then
(
if (l_prev_enrld_amt_numl < 100000) then
(
l_b_bnft_amt = 150000
)
else if (l_prev_enrld_amt_numl > 100000) then
(
l_b_bnft_amt = l_prev_enrld_amt_numl
)
)
else if (l_curr_enrld_amt_numl<150000) then
(
if (l_prev_enrld_amt_numl>100000) then
(
l_b_bnft_amt = l_curr_enrld_amt_numl
)
else if (l_prev_enrld_amt_numl<100000) then
(
l_b_bnft_amt = 150000
)
)
)
return l_b_bnft_amt,l_elig_per_elctbl_chc_id

```

# Extra Input

This formula type is used to pass an extra value to the payroll element which is linked to the standard rate.

## Contexts

The following contexts are available:

- BUSINESS\_GROUP\_ID (ENTERPRISE\_ID)
- EFFECTIVE\_DATE
- HR\_ASSIGNMENT\_ID
- PGM\_ID
- PL\_ID
- PL\_TYP\_ID
- OPT\_ID
- PERSON\_ID
- LER\_ID
- BENEFIT\_RELATION\_ID
- ACTY\_BASE\_RT\_ID
- ORGANIZATION\_ID

## Database Items

Use only the following database items that are available to formulas of this type:

- Database items based on person's assignment: HR\_ASSIGNMENT\_ID AND PERSON\_ID

## Input Variables

Input variables are not available to formulas of this type.

## Return Variables

The following return variables are available to formulas of this type. Any name can be used for the return variable.

### ***Sample Formula 1:***

Out Variable Name	Data Type	Remarks
L_OUTPUT	Number/Character	Return a number/character value depending on the data type of the corresponding payroll element's input value

## Sample Formula 1:

FORMULA TYPE : Extra Input

DESCRIPTION : Formula to get the Medical Care Reimbursement Coverage amount

```

P_ENRLD_BNFT_AMT = 0
l_bnft_amt_char = '0'
l_lf_evt_dt_char = ben_fn_get_char_value('BEN_PER_IN_LER'
, 'LF_EVT_OCRD_DT'
, ''
, ''
, 'BENEFIT_RELATION_NAME'
, 'DFLT')
l_lf_evt_dt = to_date(l_lf_evt_dt_char)
l_pay_perd_end_date = TO_DATE(BEN_FN_GET_CHAR_VALUE( 'PAY_TIME_PERIODS'
, 'END_DATE'
, ''
, ''
, 'START_DATE'
, l_lf_evt_dt_char
)
)
CHANGE_CONTEXTS (EFFECTIVE_DATE = l_pay_perd_end_date)
(
l_bnft_amt_char = ben_fn_get_char_value('BEN_PRTT_ENRT_RSLT'
, 'BNFT_AMT'
, 'Medical Care Reimbursement'
, 'Coverage'
)
IF (l_bnft_amt_char <> 'N' AND l_bnft_amt_char <> 'NO_DATA_FOUND' AND l_bnft_amt_char <> ' ')
THEN
(
P_ENRLD_BNFT_AMT = TO_NUMBER(l_bnft_amt_char)
)
)
Return P_ENRLD_BNFT_AMT

```

## Sample Formula 2:

FORMULA TYPE: Extra Input

DESCRIPTION : Get HSA Coverage Amount

```

COVERAGE_AMT = 0
l_eff_dt = GET_CONTEXT(EFFECTIVE_DATE, to_date('1951/01/01 00:00:00'))
CHANGE_CONTEXTS(EFFECTIVE_DATE=l_eff_dt)
(
l_cvg_amt_char = ben_fn_get_char_value('BEN_PRTT_ENRT_RSLT'
, 'BNFT_AMT'
, 'Health Savings Account'
, 'Elect')
) /* end change contexts */
if (l_cvg_amt_char <> 'NO_DATA_FOUND' AND l_cvg_amt_char <> 'N' AND l_cvg_amt_char <> ' ')
then
(
COVERAGE_AMT = to_number(l_cvg_amt_char)
)
else
(
COVERAGE_AMT = 0
)
return COVERAGE_AMT

```

# Common Mistakes in Formulas

Here are some common mistakes that occur while using formulas:

## Sample 1

```
CHANGE_CONTEXTS(EFFECTIVE_DATE=l_eff_date)
(
  WHILE PER_EXT_CONT_CONTACT_TYPE.exists(i)
  LOOP
    (
      IF (PER_EXT_CONT_PER_PERSON_ID[i]=CON_PERSON_ID) AND
        ( PER_EXT_CONT_CONTACT_TYPE[i]='C AND
        (PER_ASG_ADD_REGION2='NJ' or upper(PER_ASG_ADD_REGION2)='NEW JERSEY') AND
        (MONTHS_BETWEEN(l_if_evt_ocrd_dt,PER_EXT_CONT_DATE_OF_BIRTH[i]))<372 THEN
        ( ELIGIBLE='Y' )
      i=i+1 ②
    )
  )
  RETURN INELIGIBLE ③
```

### Mistake description

Callout	Description
1	Ensure that you include all the relevant DB items within the change context scope. Not including them within the changed scope leads to inconsistent results.
2	Ensure that you increment or initialize the index
3	Ensure that you enter correct return variable names for certain formula types instead of specific variable names.

## Sample 2

```

DEFAULT FOR l_orgnl_enrt_dt is '2012/03/01 00:00:00'(date)
DEFAULT FOR l_return_value is -9
DEFAULT FOR pay_periods_per_period is 999

l_bnft_amt_number = 0
l_orgnl_enrt_dt_char = 'N'

l_orgnl_enrt_dt_char = ben_fn_get_char_value('BEN_PRTT_ENRT_RSLT','ORGNL_ENRT_DT','Health Care FSA',
'Annual Goal Amount')

l_bnft_amt_char = ben_fn_get_char_value('BEN_PRTT_ENRT_RSLT','BNFT_AMT','Health Care FSA', 'Annual Goal
Amount')

if l_bnft_amt_char <> 'N' then
  ( l_bnft_amt_number = to_number(l_bnft_amt_char) )
  1 [ if (l_orgnl_enrt_dt_char <> 'N' AND l_orgnl_enrt_dt_char <> ' ') then
    ( l_orgnl_enrt_dt = to_date(l_orgnl_enrt_dt_char)
    )
  else
    ( l_orgnl_enrt_dt = to_date('2012/03/01 00:00:00')
    )
  ]

```

### Mistake description

Callout	Description
1	Benefits functions returns amounts and dates in character format. Ensure that you convert values to date and numbers.

## Sample 3

```

1 [ WHILE BEN_PEN_PL_NAME_TN.exists(i)
  LOOP
  (
    IF(BEN_PEN_PL_NAME_TN[i] = 'Health Savings Account'
    OR BEN_PEN_PL_NAME_TN[i] = 'Health Care Flexible Spending Account'
    OR BEN_PEN_PL_NAME_TN[i] = 'Dependent Care Flexible Spending Account')
    THEN
    (
      l_orgnl_enrt_dt_char = BEN_FN_GET_CHAR_VALUE('BEN_PRTT_ENRT_RSLT',
          'ORGNL_ENRT_DT',
          BEN_PEN_PL_NAME_TN[i],
          'ENROLLED')

      l_bnft_amt_char = ben_fn_get_char_value('BEN_PRTT_ENRT_RSLT',
          'BNFT_AMT',
          BEN_PEN_PL_NAME_TN[i],
          'ENROLLED')

      l = l + 1
    )
    *** Below 3 statements can well fetch you the above details without using the formula functions
    l_bnft_amt=BEN_PEN_BNFT_AMT_NN[j]
    l_enrt_cvg strt_dt=BEN_PEN_ENRT_CVG_STRT_DT_DN[j]
    l_ori_enrt_cvg strt_dt=BEN_PEN_ORGNL_ENRT_DT_DN[j]
  )

```

### Mistake description

Callout	Description
1	When you use array DB items, you need not use the below functions, as the array DB items allow you to access other enrollment details like original enrollment dates and, benefit amounts.

## Sample 4

- default for PER\_ASG\_COUNTRY\_OF\_BIRTH is 'XX'
- default for PER\_ASG\_JOB\_NAME is 'ABC'
- ①  • DEFAULT FOR CMP\_ASSIGNMENT\_SALARY\_AMOUNT IS 0
- DEFAULT FOR CMP\_ASSIGNMENT\_SALARY\_ANNUALIZATION\_FACTOR IS 0
- ②  • l\_effective\_date = GET\_CONTEXT(EFFECTIVE\_DATE,to\_date('1951/01/01 00:00:00'))
- 
- l\_cntry = PER\_ASG\_COUNTRY\_OF\_BIRTH
- l\_job = PER\_ASG\_JOB\_NAME
- 
- if ( l\_cntry <> 'US' and l\_job = 'Content Writer' ) then
- (WAIT\_PERD\_VAL = 6
- WAIT\_PERD\_UOM = 'MO' )
- Else
- (WAIT\_PERD\_VAL = 1
- WAIT\_PERD\_UOM = 'MO' )
- 
- return WAIT\_PERD\_VAL, WAIT\_PERD\_UOM

### Mistake description

Callout	Description
1	Ensure that you don't have unused DB items and variables in your formula. They can throw errors at run time, requiring you to recompile formulas.
2	Use canonical date formats while initializing date variables to avoid date format related errors at run time.

## Sample 5

**Formula type = Post election edit rule**

**1**

```
=====
WHILE BEN_PEN_PL_NAME_TN.exists(i)
LOOP
(
  IF(BEN_PEN_PL_NAME_TN[i] = 'Health Savings Account')
  THEN
  (
    l_bnft_amt=BEN_PEN_BNFT_AMT_NN[j]
    l_enrt_cvg strt_dt=BEN_PEN_ENRT_CVG_STRT_DT_DN[j]
    l_ori_enrt_cvg strt_dt=BEN_PEN_ORGNL_ENRT_DT_DN[j]
  )
  l = l + 1
)
```

```
l_eff_date = GET_CONTEXT(EFFECTIVE_DATE,to_date('1951/01/01 00:00:00') )
```

```
l_context_date=add_days(l_eff_date,30)
```

**2**

```
change_contexts(EFFECTIVE_DATE=l_context_date)
```

```
(
```

```
/* put your while loop here */
```

```
)
```

### **Mistake description**

Callout	Description
1	DB items don't return enrolled plan or option names when enrollments begin from a later date in future due to first of next month or other similar coverage codes.
2	Manipulate the effective date as per your needs and enclose the above WHILE loop within the change context call.

## Sample 6

```

Formula type = Extract Rule
Formula Name = GET_COMM_VAL
=====
① Default for BEN_PBB_VAL is 0
l_val = 0

CHANGE_CONTEXTS(BUSINESS_GROUP_ID = 1)
(
CHANGE_CONTEXTS(BNFTS_BAL_ID = 3001)
(
  l_val = BEN_PBB_VAL
)
)
return l_val

Formula type = Coverage Amount Calculation
Formula Name = calc_coverage_val
=====
l_cvg = 100
l_value = 0

② if IS_EXECUTABLE('GET_COMM_VAL') then
(
  SET_INPUT('BUSINESS_GROUP_ID', 1)
  SET_INPUT('EFFECTIVE_DATE', GET_CONTEXT(EFFECTIVE_DATE,to_date('2013/01/01
12:00:00','YYYY/MM/DD HH24:MI:SS')) )
③ SET_INPUT('HR_ASSIGNMENT_ID', GET_CONTEXT(HR_ASSIGNMENT_ID, -1))
  EXECUTE ('GET_COMM_VAL')
  l_value = GET_OUTPUT('l_val' , 120)
)
l_cvg = l_cvg + l_value
return l_cvg

```

### Mistake description

Callout	Description
1	DB items are available for formulas based on data partitioning and formula context usages. So not all formulas can access all DB items. In this example, BEN_PBB_VAL DB item is not available to coverage calculation formula type. We recommend that you use Enterprise level formulas within Benefits. Don't set LDG while creating formulas.
2	Ensure that the called function is executable before you call the function.
3	Use the extract rule which has access to BEN_PBB_VAL DB item.

# Formula Debugging and Troubleshooting

You need to place the following debug statements as necessary:

```

l_log_data = ESS_LOG_WRITE('*****Start ff debug - messages appear in
the batch process log files')
ret = HR_TRACE('*****Start ff debug - messages appear in the PLSQL
trace file')
.

.

.

l_log_data = ess_log_write('end ff debug: L_MX_CVG_ALWD_AMT'||to_char(L_MAX_CVG_ALWD_AMT))
ret = HR_TRACE('end ff debug: L_MX_CVG_ALWD_AMT'||to_char(L_MX_CVG_ALWD_AMT))

```

## Formula Debugging

```

DEFAULTFOR CMP_ASSIGNMENT_SALARY_AMOUNT IS 0
DEFAULTFOR CMP_ASSIGNMENT_SALARY_ANNUALIZATION_FACTOR IS 0
L_MN_CVG_ROD_AMT=0
L_MX_CVG_ALWD_AMT=0
L_MX_CVG_WCFN_AMT=30000
L_MX_CVG_INCR_ALWD_AMT=10000
L_MX_CVG_INCR_WCF_ALWD_AMT=10000

```

```

l_log_data = ESS_LOG_WRITE('*****Start ff debug')
ret = HR_TRACE('*****Start ff debug')

IF(3 * CMP_ASSIGNMENT_SALARY_AMOUNT * CMP_ASSIGNMENT_SALARY_ANNUALIZATION_FACTOR) > 200000
THEN (L_MX_CVG_ALWD_AMT = 200000)
ELSE (L_MX_CVG_ALWD_AMT = 3 * CMP_ASSIGNMENT_SALARY_AMOUNT * CMP_ASSIGNMENT_SALARY_ANNUALIZATION_FACTOR)

l_log_data = ess_log_write('end ff debug: L_MX_CVG_ALWD_AMT'||to_char(L_MX_CVG_ALWD_AMT))

ret = HR_TRACE('end ff debug: L_MX_CVG_ALWD_AMT'||to_char(L_MX_CVG_ALWD_AMT))

Return L_MN_CVG_ROD_AMT, L_MX_CVG_ALWD_AMT, L_MX_CVG_WCFN_AMT, L_MX_CVG_INCR_ALWD_AMT, L_MX_CVG_INCR_WCF_ALWD_AMT

```

Place debug statements  
as necessary

Note: The log files that the benefits batch jobs create will have the above 2 statements printed in the log file which can be accessed from the Monitor requests UI.

## Formula Troubleshooting

If the formula doesn't return the correct values or if the end result is not as expected, then here's how you can diagnose the issue:

1. Test with the seeded codes first. You need to remove the formula before testing the seeded functionality.
2. Use hard coded Values and test a simple fast formula.
3. If you used any function, then replace them with hard coded values and test.
4. Trace the formula to gather details of the values passed.
5. If the above steps don't help you resolve the problem, then provide the results of the same for oracle to review.

While troubleshooting the formula, you need to ensure the following:

1. Don't delete a formula which has been attached to benefits plan design, especially after the plan design has been processed, and participants have been found eligible for the plan design.
2. Don't recreate the formula with the same name after deleting. It doesn't reinstate the older behavior. You would face errors at run time. Instead, edit the formula or create a new formula to replace the older one.

## Formula – FAQ

The following sections will provide you information about using formulas.

### How you use arrays within formulas

FORMULA Name : ELIG1

FORMULA TYPE : Participation and Rate Eligibility

```
Default for LOCATIONS is EMPTY_NUMBER_NUMBER
I=1
ELIGIBLE='N'
LOCATIONS[1]=1625
LOCATIONS[2]=1630
l_job_code=123 /* to be obtained from a database item */
WHILE (LOCATIONS.EXISTS(I)) LOOP
(
if (LOCATIONS[I] = l_job_code) then
(ELIGIBLE='Y'
/* If found eligible you need not iterate */
EXIT
)
/* increment the counter*/
I=I+1
) /* end loop */
Return ELIGIBLE
```

### How you use User defined tables within benefits formulas

Formula Name: HRA\_ER\_CONTRIBUTION

Formula Type: Post Enrollment Rate Calculation

```
er_contribution = 0 hra_single_amt = 0 hra_family_amt = 0
BEN_TABLE = 'ORCL_GLOBAL_VALUES' BEN_TABLE_COLUMN = 'Global_Value' BEN_ROW_VALUE = 'HRA_SINGLE_AMT'
hra_single_amt = to_num(get_table_value ( BEN_TABLE, BEN_TABLE_COLUMN, BEN_ROW_VALUE))
/* determine medical plan option */
EE_only = ben_fn_get_char_value('BEN_PRTT_ENRT_RSLT', 'ENROLLED', 'Medical', 'Employee') EE_Waive =
ben_fn_get_char_value('BEN_PRTT_ENRT_RSLT', 'ENROLLED', 'Medical', 'Waive') /* Use the option from above
code to determine the ER contribution */ IF EE_Waive = 'Y' THEN
( er_contribution = 0 )
ELSE
( er_contribution = hra_single_amt )
l_result = er_contribution / 12
Return l_result
```

### How you use element entry values within benefits rates

FORMULA NAME: RT\_VALUE\_CALC

FORMULA TYPE: Rate Value Calculation

```
Default_data_value for ORCL_PP_ELEMENT_2_AMOUNT_ENTRY_VALUE is 0
Default for BEN_OPT_NAME is 'NA'
/* Initialization section */
i=1
l_val=0
```

```

l_ret=0
if (ORCL_PP_ELEMENT_2_AMOUNT_ENTRY_VALUE.exists(i)) then (l_val=ORCL_PP_ELEMENT_2_AMOUNT_ENTRY_VALUE[i])
if (BEN_OPT_NAME='ACT_ZHRX_PLAN_1_EMP') then ( l_ret=750-l_val )
else if (BEN_OPT_NAME='ACT_ZHRX_PLAN_1_FAM' OR BEN_OPT_NAME='ACT_ZHRX_PLAN_1_CHILD') then ( l_ret=1500-
l_val )
if (l_ret<0) then
( l_ret=0 )
return l_ret

```

## How you access pay periods within rate periodization rules to compute remaining contributions

FORMULA NAME: RT\_PERIOD\_CALC

FORMULA TYPE: Rate Periodization Rule

```

Inputs are BEN_IV_CONVERT_FROM (text), BEN_IV_CONVERT_FROM_VAL (text)
DEFAULT_DATA_VALUE FOR BEN_PEN_PL_NAME_TN is 'My-Default'
DEFAULT_DATA_VALUE FOR BEN_PEN_ENRT_CVG_STRT_DT_DN is '1951/01/01 0:00:00' (date)
DEFAULT_DATA_VALUE FOR BEN_PEN_ORGNL_ENRT_DT_DN is '1951/01/01 0:00:00' (date)
DEFAULT FOR BEN_YRP_START_DATE is '1951/01/01 0:00:00' (date) DEFAULT_DATA_VALUE FOR BEN_PEN_OPT_NAME_TN is
'My-Default' DEFAULT_DATA_VALUE FOR BEN_PEN_BNFT_AMT_NN is 0
l_eff_date='1951/01/01 0:00:00' (date)
l_new_eff_date='1951/01/01 0:00:00' (date)
l_enrt_cvg strt_dt='1951/01/01 0:00:00' (date)
l_ori_enrt_cvg strt_dt='1951/01/01 0:00:00' (date)
l_chk_date='1951/01/01 0:00:00' (date)
l_yr_perd_st_date='1951/01/01 0:00:00' (date)
l_bnft_amt=0
l_prev_bnft_amt=0
l_prev_cntbn=0
DFND_VAL=0
ANN_VAL=0
CMCD_VAL=0
i=1
j=1
l_total_tp_in_cal_year=1
l_pay_perd_totper_rem=99
l_pay_cur_perd=1
/* input values */
l_iv_period=BEN_IV_CONVERT_FROM
l_iv_val=to_number(BEN_IV_CONVERT_FROM_VAL)
/* change contexts for accomodating waiting period */
l_eff_date=GET_CONTEXT(EFFECTIVE_DATE, to_date('1951/01/01 0:00:00'))
l_eff_date=add_days(l_eff_date,30)
l_new_eff_date=add_days(last_day(l_eff_date),1)
CHANGE_CONTEXTS(EFFECTIVE_DATE=l_new_eff_date)
/* get the bnft amount and the enrt cvg strt date within this block */
while (BEN_PEN_PL_NAME_TN.exists(j)) loop
(
if ((BEN_PEN_PL_NAME_TN[j]='Dependent Care FSA') and (BEN_PEN_OPT_NAME_TN[j]='Annual Goal Amount')) then (
l_bnft_amt=BEN_PEN_BNFT_AMT_NN[j]
l_enrt_cvg strt_dt=BEN_PEN_ENRT_CVG_STRT_DT_DN[j] l_ori_enrt_cvg strt_dt=BEN_PEN_ORGNL_ENRT_DT_DN[j] )
j=j+1
) /*end loop plan option */
) /*end change contexts */
j=1
CHANGE_CONTEXTS(EFFECTIVE_DATE=l_new_eff_date-1)
/* get the previous bnft amount within this block */
while (BEN_PEN_PL_NAME_TN.exists(j)) loop
(
if ((BEN_PEN_PL_NAME_TN[j]='Dependent Care FSA') and (BEN_PEN_OPT_NAME_TN[j]='Annual Goal Amount')) then (
l_prev_bnft_amt=BEN_PEN_BNFT_AMT_NN[j]
)

```

```

j=j+1
) /*end loop plan option */
) /*end change contexts */
IF (l_bnft_amt = 0 and l_iv_period = 'ANNUAL' ) THEN ( l_bnft_amt = l_iv_val)
/* check if enrolled earlier in same year */
IF (l_ori_enrt_cvg strt_dt <= l_enrt_cvg strt_dt) THEN ( l_chk_date = l_ori_enrt_cvg strt_dt )
ELSE
( l_chk_date = l_enrt_cvg strt_dt )
/* substitute the program id in below clause to pick year period start date */ CHANGE_CONTEXTS(PGM_ID =
202,EFFECTIVE_DATE=l_new_eff_date) (
l_yr_perd_st_date = BEN_YRP_START_DATE
)
/* If enrolled in prior year, use this year period start date */ IF ( l_chk_date < l_yr_perd_st_date ) THEN
(
l_chk_date = l_yr_perd_st_date
)
l_pay_cur_perd= to_number(BEN_FN_GET_CHAR_VALUE('PAY_TIME_PERIODS','PERIOD_NUM',' ',' '
,'START_DATE',to_char(l_chk_date)))
l_total_tp_in_cal_year =
to_number(BEN_FN_GET_CHAR_VALUE('PAY_TIME_PERIODS','TOTAL_TP_IN_CALENDAR_YEAR',' ',' '
,'START_DATE',to_char(l_chk_date)))
l_pay_perd_totper_rem=l_total_tp_in_cal_year - l_pay_cur_perd +1
IF (l_prev_bnft_amt <> 0 and l_bnft_amt <> l_prev_bnft_amt ) THEN
(
l_prev_cntbn = (l_prev_bnft_amt/l_total_tp_in_cal_year)*(l_pay_cur_perd - 1)
)
ELSE
(
l_prev_cntbn = 0
)
CMCD_VAL = (l_bnft_amt - l_prev_cntbn )/l_pay_perd_totper_rem DFND_VAL = (CMCD_VAL *
l_pay_perd_totper_rem)/12 ANN_VAL = l_bnft_amt
RETURN DFND_VAL,ANN_VAL,CMCD_VAL

```

## Associating HR actions to benefits life events

FORMULA NAME : New Hire Event

FORMULA TYPE : Person Changes Causes Life Event

```

/*===== DATABASE ITEM DEFAULTS BEGIN =====*/ DEFAULT for PER_ASG_ACTION_CODE is 'AA'
/*===== DATABASE ITEM DEFAULTS ENDS===== */ /*===== INPUT VALUES DEFAULT BEGIN
===== */ /*===== INPUT VALUES DEFAULT ENDS ===== */
/*===== INPUTS SECTION BEGIN =====*/
/*===== INPUTS SECTION ENDS =====*/
/*===== FORMULA SECTION BEGIN =====*/
l_action_code = PER_ASG_ACTION_CODE
if NOT PER_ASG_ACTION_CODE was defaulted then
(
if ((l_action_code='HIRE') OR (l_action_code='EMPL_REHIRE') )then
(l_create_ptnl = 'Y')
else
(l_create_ptnl = 'N')
)
return l_create_ptnl
/*===== FORMULA SECTION END =====*/

```

## Accessing contacts or family member details to determine eligibility

FORMULA NAME : DPNT\_ELIG

FORMULA TYPE : Dependent eligibility

```
DEFAULT_DATA_VALUE for PER_EXT_CONT_CONTACT_TYPE is 'NA'
```

```

DEFAULT_DATA_VALUE FOR PER_EXT_CONT_DATE_OF_BIRTH IS '1951/01/01' (DATE)
DEFAULT_DATA_VALUE FOR PER_EXT_CONTACT_ATTRIBUTE1 IS 'NA' DEFAULT_DATA_VALUE FOR PER_EXT_CONT_SEX IS 'NA'
  ELIGIBLE = 'N'
l_eff_dt = GET_CONTEXT(EFFECTIVE_DATE, to_date('1951/01/01 00:00:00') )
/* S - Spouse(female), 35 yrs age as of event date */
i=1
WHILE PER_EXT_CONT_CONTACT_TYPE.exists(i)
LOOP
(
IF (PER_EXT_CONT_CONTACT_TYPE[i]='S'
AND PER_EXT_CONTACT_ATTRIBUTE1[i] <> 'NA' AND PER_EXT_CONT_SEX[i] = 'F' ) AND (MONTHS_BETWEEN(l_eff_dt,
PER_EXT_CONT_DATE_OF_BIRTH[i])) < 4200
THEN
( ELIGIBLE = 'Y' )
i=i+1
)
RETURN ELIGIBLE

```

## How you access benefits data within HCM extracts

FORMULA NAME : ben\_ext\_rule

FORMULA TYPE : Extract rule

```

Default for l_Val is 'XX'
DEFAULT_DATA_VALUE for BEN_PBG_BENEFIT_GROUP_NAME_TN is 'aa1'
default for i is 1
i=1
L_BG_ID = GET_CONTEXT(BUSINESS_GROUP_ID, 1)
L_EFF_DATE = GET_CONTEXT(EFFECTIVE_DATE, to_date('1951/01/01 00:00:00'))
L_PERSON_ID = GET_CONTEXT(PERSON_ID, 9999)
CHANGE_CONTEXTS(EFFECTIVE_DATE = L_EFF_DATE, BUSINESS_GROUP_ID = L_BG_ID, PERSON_ID
= L_PERSON_ID )
(
while BEN_PBG_BENEFIT_GROUP_NAME_TN.exists(i) loop
(
l_Val=BEN_PBG_BENEFIT_GROUP_NAME_TN[i] i=i+1
)
)
return l_Val

```

## Appendix 1: Contexts

Table below describes the different Benefits formula types available in Oracle Fusion V1, where they can be located in UI, and the various contexts available for each formula type. Please note that a lot of formula types which are available at the plan level would also be available at other levels like program, plan in program etc.

### Formula types navigation table

FORMULA_TYPE_NAME	NAVIGATION DETAILS	CONTEXT_NAME
Action Type Due Date	Plan Configuration > Benefit Plan Details> Certification train stop	BUSINESS_GROUP_ID ( ENTERPRISE_ID)
		EFFECTIVE_DATE
		ELIG_PER_ELCTBL_CHC_ID

FORMULA_TYPE_NAME	NAVIGATION DETAILS	CONTEXT_NAME
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Age Calculation	Plan Configuration > Manage Derived Factors > Age tab	BUSINESS_GROUP_ID ( ENTERPRISE_ID) EFFECTIVE_DATE HR_ASSIGNMENT_ID LER_ID OPT_ID ORGANIZATION_ID PERSON_ID PGM_ID PL_ID PL_TYP_ID
Age Determination Date	Plan Configuration > Derived Factors > Age tab	BNFTS_BAL_ID BUSINESS_GROUP_ID ( ENTERPRISE_ID) EFFECTIVE_DATE HR_ASSIGNMENT_ID LER_ID OPT_ID ORGANIZATION_ID PERSON_ID PGM_ID PL_ID PL_TYP_ID
Automatic Enrollment Method	Plan Configuration > Benefit Plan Details > Enrollment train stop	BUSINESS_GROUP_ID ( ENTERPRISE_ID) EFFECTIVE_DATE

FORMULA_TYPE_NAME	NAVIGATION DETAILS	CONTEXT_NAME
		ELIG_PER_ID
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Beneficiary Certification Required	Plan Configuration > Benefit Plan Details > Dependent/ Beneficiary train stop	BUSINESS_GROUP_ID ( ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Change Dependent Coverage	Plan Configuration > BenefitPlan Details > Dependent/ Beneficiary train stop	BUSINESS_GROUP_ID ( ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Compensation Calculation	Plan Configuration > Derived Factors > Compensation tab	BUSINESS_GROUP_ID ( ENTERPRISE_ID)

FORMULA_TYPE_NAME	NAVIGATION DETAILS	CONTEXT_NAME
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Compensation Determination Date	Plan Configuration > Derived Factors > Compensation tab	BNFTS_BAL_ID
		BUSINESS_GROUP_ID ( ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PAYROLL_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Compensation Object Selection	Evaluation and Reporting > Processes> Evaluate Life Event Participation > Comp object selection parameter	BUSINESS_GROUP_ID ( ENTERPRISE_ID)
		EFFECTIVE_DATE
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Coverage Amount Calculation	Plan Configuration > Benefit Plan coverage	BUSINESS_GROUP_ID (ENTERPRISE_ID)

FORMULA_TYPE_NAME	NAVIGATION DETAILS	CONTEXT_NAME
		EFFECTIVE_DATE
		ELIG_PER_ELCTBL_CHC_ID
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Coverage Lower Limit	Plan Configuration > Benefit Plan Coverage	BUSINESS_GROUP_ID (ENTERPRISE_ID)
		EFFECTIVE_DATE
		ELIG_PER_ELCTBL_CHC_ID
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
		BUSINESS_GROUP_ID (ENTERPRISE_ID)
		EFFECTIVE_DATE
		ELIG_PER_ELCTBL_CHC_ID
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID

FORMULA_TYPE_NAME	NAVIGATION DETAILS	CONTEXT_NAME
		PL_TYP_ID
Default Enrollment	Plan Configuration > Benefit Plan Details > Enrollment train stop	BUSINESS_GROUP_ID (ENTERPRISE_ID)
		EFFECTIVE_DATE
		ELIG_PER_ID
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Default to Assign Pending Action	Plan Configuration > Benefit Plan Details > Enrollment train stop	BUSINESS_GROUP_ID (ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Dependent Certification Required	Plan Configuration > Benefit Plan Details > Dependent / beneficiary train stop	BUSINESS_GROUP_ID (ENTERPRISE_ID)
		EFFECTIVE_DATE
		ELIG_PER_ELCTBL_CHC_ID
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID

FORMULA_TYPE_NAME	NAVIGATION DETAILS	CONTEXT_NAME
		PL_ID
		PL_TYP_ID
Dependent Coverage End Date	Plan Configuration > Benefit Plan Details > Dependent / beneficiary train stop	BUSINESS_GROUP_ID (ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
		BUSINESS_GROUP_ID (ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Dependent Eligibility	Plan Configuration > Benefit Plan Details > Dependent / beneficiary train stop	BUSINESS_GROUP_ID ( ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID

FORMULA_TYPE_NAME	NAVIGATION DETAILS	CONTEXT_NAME
		PL_TYP_ID
Element Type and Input Value Determination	Plan Configuration > Benefit Rates	ACTY_BASE_RT_ID
		BUSINESS_GROUP_ID (ENTERPRISE_ID)
		EFFECTIVE_DATE
		ELIG_PER_ELCTBL_CHC_ID
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Eligibility Access Calculation Rule	Plan Configuration > Eligibility Profiles	BUSINESS_GROUP_ID (ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Enrollment Certification Required	Plan Configuration > Benefit Plan Details > Enrollment train stop	BUSINESS_GROUP_ID (ENTERPRISE_ID)
		EFFECTIVE_DATE
		ELIG_PER_ELCTBL_CHC_ID
		ENRT_CTFN_TYP_CD
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID

FORMULA_TYPE_NAME	NAVIGATION DETAILS	CONTEXT_NAME
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Enrollment Coverage Start Date	Plan Configuration > Benefit Plan Details > Enrollment train stop	BUSINESS_GROUP_ID (ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Enrollment End	Plan Configuration > Benefit Plan Details > Enrollment train stop	BUSINESS_GROUP_ID (ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Enrollment Opportunity	Plan Configuration > Benefit Plan Details > Enrollment train stop	BUSINESS_GROUP_ID (ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID

FORMULA_TYPE_NAME	NAVIGATION DETAILS	CONTEXT_NAME
		PGM_ID
		PL_ID
		PL_TYP_ID
Enrollment Period End Date	Plan Configuration > Benefit Plan Details > Enrollment train stop	BUSINESS_GROUP_ID (ENTERPRISE_ID) EFFECTIVE_DATE HR_ASSIGNMENT_ID LER_ID OPT_ID ORGANIZATION_ID PERSON_ID PGM_ID PL_ID PL_TYP_ID
Enrollment Period Start Date	Plan Configuration > Benefit Plan Details > Enrollment train stop	BUSINESS_GROUP_ID (ENTERPRISE_ID) EFFECTIVE_DATE HR_ASSIGNMENT_ID LER_ID OPT_ID ORGANIZATION_ID PERSON_ID PGM_ID PL_ID PL_TYP_ID
Evaluate Life Event	Plan Configuration > Benefit Life Events	BUSINESS_GROUP_ID (ENTERPRISE_ID) EFFECTIVE_DATE HR_ASSIGNMENT_ID LER_ID OPT_ID ORGANIZATION_ID PERSON_ID
Extra Input	Plan Configuration > Manage Rates	ACTY_BASE_RT_ID

FORMULA_TYPE_NAME	NAVIGATION DETAILS	CONTEXT_NAME
		BUSINESS_GROUP_ID (ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Family Member Determination	Plan Configuration > Benefit Plan Details > Eligibility train stop	BUSINESS_GROUP_ID (ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Hours Worked Calculation	Plan Configuration > Derived Factors > Hours worked tab	BUSINESS_GROUP_ID (ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Hours Worked Determination Date	Plan Configuration > Derived Factors > Hours worked tab	BNFTS_BAL_ID

FORMULA_TYPE_NAME	NAVIGATION DETAILS	CONTEXT_NAME
		BUSINESS_GROUP_ID (ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PAYROLL_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Length of Service Calculation	Plan Configuration > Derived Factors > Length of Service tab	BUSINESS_GROUP_ID (ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Length of Service Date to Use	Plan Configuration > Derived Factors > Length of Service tab	BUSINESS_GROUP_ID (ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID

FORMULA_TYPE_NAME	NAVIGATION DETAILS	CONTEXT_NAME
Length of Service Determination Date	Plan Configuration > Derived Factors > Length of Service tab	BNFTS_BAL_ID
		BUSINESS_GROUP_ID (ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
Life Event Reason Timeliness	Plan Configuration > Benefit Life Events	PL_TYP_ID
		BUSINESS_GROUP_ID (ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID
Maximum Coverage Amount Calculation	Plan Configuration > Benefit Plan Coverage	PERSON_ID
		BUSINESS_GROUP_ID (ENTERPRISE_ID)
		EFFECTIVE_DATE
		ELIG_PER_ELCTBL_CHC_ID
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
Maximum Period of Enrollment	Plan Configuration > Benefit Plan Details > Enrollment train stop	PL_ID
		PL_TYP_ID
		BUSINESS_GROUP_ID (ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID

FORMULA_TYPE_NAME	NAVIGATION DETAILS	CONTEXT_NAME
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Maximum Period of Enrollment Determination Date	Plan Configuration > Benefit Plan Details > Enrollment train stop	BUSINESS_GROUP_ID ( ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Maximum Waiting Period Date to Use	Plan Configuration > Benefit Plan Details > Eligibility train stop	BUSINESS_GROUP_ID ( ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Minimum Coverage Amount Calculation	Plan Configuration > Benefit Plan Coverage	BUSINESS_GROUP_ID ( ENTERPRISE_ID)
		EFFECTIVE_DATE
		ELIG_PER_ELCTBL_CHC_ID
		HR_ASSIGNMENT_ID

FORMULA_TYPE_NAME	NAVIGATION DETAILS	CONTEXT_NAME
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Partial Month Effective Date Determination	Plan Configuration > Benefit Rates	ACTY_BASE_RT_ID
		BUSINESS_GROUP_ID ( ENTERPRISE_ID)
		EFFECTIVE_DATE
		ELIG_PER_ELCTBL_CHC_ID
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Partial Month Proration Method	Plan Configuration > Benefit Rates	ACTY_BASE_RT_ID
		BUSINESS_GROUP_ID ( ENTERPRISE_ID)
		EFFECTIVE_DATE
		ELIG_PER_ELCTBL_CHC_ID
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID

FORMULA_TYPE_NAME	NAVIGATION DETAILS	CONTEXT_NAME
Partial Month Proration Rule	Plan Configuration > Manage Rates	ACTY_BASE_RT_ID BUSINESS_GROUP_ID ( ENTERPRISE_ID) EFFECTIVE_DATE ELIG_PER_ELCTBL_CHC_ID HR_ASSIGNMENT_ID LER_ID OPT_ID ORGANIZATION_ID PERSON_ID PGM_ID PL_ID PL_TYP_ID
Participation Eligibility End Date	Plan Configuration > Benefit Plan Details > Eligibility train stop	BUSINESS_GROUP_ID ( ENTERPRISE_ID) EFFECTIVE_DATE HR_ASSIGNMENT_ID LER_ID OPT_ID ORGANIZATION_ID PERSON_ID PGM_ID PL_ID PL_TYP_ID
Participation Eligibility Start	Plan Configuration > Benefit Plan Details > Eligibility train stop	BUSINESS_GROUP_ID ( ENTERPRISE_ID) EFFECTIVE_DATE HR_ASSIGNMENT_ID LER_ID OPT_ID ORGANIZATION_ID PERSON_ID PGM_ID PL_ID

FORMULA_TYPE_NAME	NAVIGATION DETAILS	CONTEXT_NAME
		PL_TYP_ID
Participation and Rate Eligibility	Plan Configuration > Eligibility Profiles	BUSINESS_GROUP_ID (ENTERPRISE_ID)
		DATE_EARNED
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PAYROLL_ASSIGNMENT_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Payment Schedule	Plan Configuration > Benefit Rates	BUSINESS_GROUP_ID ( ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Person Change Causes Life Event	Plan Configuration > Benefit Life Events	BUSINESS_GROUP_ID ( ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		ORGANIZATION_ID
		PERSON_ID
Person Selection	Evaluation and Reporting > Submit any process > Person selection parameter	BUSINESS_GROUP_ID ( ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID

FORMULA_TYPE_NAME	NAVIGATION DETAILS	CONTEXT_NAME
		LER_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Postelection Edit	Plan Configuration > Benefit Plans > Enrollment train stop	BUSINESS_GROUP_ID ( ENTERPRISE_ID) EFFECTIVE_DATE HR_ASSIGNMENT_ID LER_ID ORGANIZATION_ID PERSON_ID
Prorate Annual Election Value	Plan Configuration > Benefit Rates	ACTY_BASE_RT_ID BUSINESS_GROUP_ID ( ENTERPRISE_ID) EFFECTIVE_DATE HR_ASSIGNMENT_ID LER_ID OPT_ID ORGANIZATION_ID PERSON_ID PGM_ID PL_ID PL_TYP_ID
Range of Scheduled Hours	Range of Scheduled Hours Profiles	BUSINESS_GROUP_ID ( ENTERPRISE_ID) EFFECTIVE_DATE HR_ASSIGNMENT_ID LER_ID OPT_ID ORGANIZATION_ID PERSON_ID PGM_ID

FORMULA_TYPE_NAME	NAVIGATION DETAILS	CONTEXT_NAME
		PL_ID
		PL_TYP_ID
Rate End Date	Plan Configuration > Benefit Plan Details > Enrollment train stop	ACTY_BASE_RT_ID
		BUSINESS_GROUP_ID ( ENTERPRISE_ID)
		EFFECTIVE_DATE
		ELIG_PER_ELCTBL_CHC_ID
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Rate Lower Limit	Plan Configuration > Benefit Rates	BUSINESS_GROUP_ID ( ENTERPRISE_ID)
		EFFECTIVE_DATE
		ELIG_PER_ELCTBL_CHC_ID
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Rate Periodization	Plan Configuration > Benefit Rates	BUSINESS_GROUP_ID ( ENTERPRISE_ID)
		EFFECTIVE_DATE
		ELIG_PER_ELCTBL_CHC_ID
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID

FORMULA_TYPE_NAME	NAVIGATION DETAILS	CONTEXT_NAME
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Rate Start Date	Plan Configuration > Benefit Plan Details > Enrollment train stop	ACTY_BASE_RT_ID BUSINESS_GROUP_ID ( ENTERPRISE_ID) EFFECTIVE_DATE ELIG_PER_ELCTBL_CHC_ID HR_ASSIGNMENT_ID LER_ID OPT_ID ORGANIZATION_ID PERSON_ID PGM_ID PL_ID PL_TYP_ID
Rate Upper Limit	Plan Configuration > Benefit Rates	ACTY_BASE_RT_ID BUSINESS_GROUP_ID ( ENTERPRISE_ID) EFFECTIVE_DATE ELIG_PER_ELCTBL_CHC_ID HR_ASSIGNMENT_ID LER_ID OPT_ID ORGANIZATION_ID PERSON_ID PGM_ID PL_ID PL_TYP_ID
Rate Value Calculation	Plan Configuration > Benefit Rates	ACTY_BASE_RT_ID BUSINESS_GROUP_ID ( ENTERPRISE_ID)

FORMULA_TYPE_NAME	NAVIGATION DETAILS	CONTEXT_NAME
		EFFECTIVE_DATE
		ELIG_PER_ELCTBL_CHC_ID
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Related Person Change Causes Life Event	Plan Configuration > Benefit Life Events	BUSINESS_GROUP_ID ( ENTERPRISE_ID) EFFECTIVE_DATE HR_ASSIGNMENT_ID ORGANIZATION_ID PERSON_ID
Required Period of Enrollment	Plan Configuration > Benefit Plan Details > Enrollment train stop	BUSINESS_GROUP_ID ( ENTERPRISE_ID) EFFECTIVE_DATE HR_ASSIGNMENT_ID LER_ID OPT_ID ORGANIZATION_ID PERSON_ID PGM_ID PL_ID PL_TYP_ID
Waiting Period Value and UOM	Plan Configuration > Benefit Plan Details > Additional Details train stop	BUSINESS_GROUP_ID ( ENTERPRISE_ID) EFFECTIVE_DATE HR_ASSIGNMENT_ID LER_ID OPT_ID

FORMULA_TYPE_NAME	NAVIGATION DETAILS	CONTEXT_NAME
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID
Waive Certification Required	Plan Configuration > Benefit Plan Details > Certification train stop	BUSINESS_GROUP_ID ( ENTERPRISE_ID)
		EFFECTIVE_DATE
		HR_ASSIGNMENT_ID
		LER_ID
		OPT_ID
		ORGANIZATION_ID
		PERSON_ID
		PGM_ID
		PL_ID
		PL_TYP_ID

## Appendix 2: Input Values

Input values are available for Person Changes causes life event formula type. Based on the table on which Person changes are defined, different set of input values are available for use in the formula. Input values NEW\_VAL, OLD\_VAL, COLUMN, PK\_ID are available irrespective of table on which data changes are defined.

Below given example explains what values these input variables contain.

Assume data changes are defined on table PER\_PEOPLE\_LEGISLATIVE\_F and Column MARITAL\_STATUS

A person's marital status is changed from S (single) to M (Married) COLUMN input value is set with column name i.e., MARITAL\_STATUS. NEW\_VAL input value is set with new value for column marital\_status.

For the current person record it's value is M. OLD\_VAL input value is set with old value for column marital\_status for the current person record. Its value will be set to S.

### ***Input variable values table***

Table Name	Input Values
CMP_SALARY	BEN_SAL_IN_PERSON_ID, BEN_SAL_IO_PERSON_ID,

Table Name	Input Values
	BEN_SAL_IN_SALARY_AMOUNT, BEN_SAL_IO_SALARY_AMOUNT, BEN_SAL_IN_ACTION_REASON_ID, BEN_SAL_IO_ACTION_REASON_ID, BEN_SAL_IN_SALARY_BASIS_ID, BEN_SAL_IO_SALARY_BASIS_ID, BEN_SAL_IN_ATTRIBUTE1, BEN_SAL_IO_ATTRIBUTE1, BEN_SAL_IN_ATTRIBUTE2, BEN_SAL_IO_ATTRIBUTE2, BEN_SAL_IN_SALARY_REASON_CODE, BEN_SAL_IO_SALARY_REASON_CODE, BEN_SAL_IN_DATE_FROM, BEN_SAL_IO_DATE_FROM, BEN_SAL_IN_DATE_TO, BEN_SAL_IO_DATE_TO, BEN_SAL_IN_ELEMENT_ENTRY_ID, BEN_SAL_IO_ELEMENT_ENTRY_ID, BEN_SAL_IN_FORCED_RANKING, BEN_SAL_IO_FORCED_RANKING, BEN_SAL_IN_PERFORMANCE_RATING, BEN_SAL_IO_PERFORMANCE_RATING, BEN_SAL_IN_PERFORMANCE REVIEW_ID, BEN_SAL_IO_PERFORMANCE REVIEW_ID, BEN_SAL_IN REVIEW_DATE,

Table Name	Input Values
	<p>BEN_SAL_IO_REVIEW_DATE, BEN_SAL_IN_SALARY_APPROVED, BEN_SAL_IO_SALARY_APPROVED</p>
PER_PASSPORTS	<p>BEN_PSS_IN_PERSON_ID, BEN_PSS_IO_PERSON_ID, BEN_PSS_IN_BUSINESS_GROUP_ID, BEN_PSS_IO_BUSINESS_GROUP_ID, BEN_PSS_IN_EXPIRATION_DATE, BEN_PSS_IO_EXPIRATION_DATE, BEN_PSS_IN_ISSUE_DATE, BEN_PSS_IO_ISSUE_DATE, BEN_PSS_IN_ISSUING_AUTHORITY, BEN_PSS_IO_ISSUING_AUTHORITY, BEN_PSS_IN_ISSUING_COUNTRY, BEN_PSS_IO_ISSUING_COUNTRY, BEN_PSS_IN_ISSUING_LOCATION, BEN_PSS_IO_ISSUING_LOCATION, BEN_PSS_IN_LEGISLATION_CODE, BEN_PSS_IO_LEGISLATION_CODE</p>
PER_PERIODS_OF_SERVICE	<p>BEN_PSS_IN_PASSPORT_ID, BEN_PSS_IO_PASSPORT_ID, BEN_PSS_IN_PASSPORT_NUMBER, BEN_PSS_IO_PASSPORT_NUMBER, BEN_PSS_IN_PASSPORT_TYPE,</p>

Table Name	Input Values
	<p>BEN_PSS_IO_PASSPORT_TYPE, BEN_PSS_IN_PERSON_ID, BEN_PSS_IO_PERSON_ID, BEN_PSS_IN_PROFESSION, BEN_PSS_IO_PROFESSION, BEN_PPS_IN_DATE_START, BEN_PPS_IO_DATE_START, BEN_PPS_IN_ACCEPTED_TERMINATION_DATE, BEN_PPS_IO_ACCEPTED_TERMINATION_DATE</p>
PER_PEOPLE_LEGISLATIVE_F	<p>BEN_PPS_IN_ACTION_OCCURRENCE_ID, BEN_PPS_IO_ACTION_OCCURRENCE_ID, BEN_PPS_IN_ACTUAL_TERMINATION_DATE, BEN_PPS_IO_ACTUAL_TERMINATION_DATE, BEN_PPS_IN_ADJUSTED_SVC_DATE, BEN_PPS_IO_ADJUSTED_SVC_DATE, BEN_PPS_IN_DATE_START, BEN_PPS_IO_DATE_START, BEN_PPS_IN_LAST_WORKING_DATE, BEN_PPS_IO_LAST_WORKING_DATE, BEN_PPS_IN_LEGAL_ENTITY_ID, BEN_PPS_IO_LEGAL_ENTITY_ID, BEN_PPS_IN_LEGISLATION_CODE, BEN_PPS_IO_LEGISLATION_CODE, BEN_PPS_IN_NOTIFIED_TERMINATION_DATE, BEN_PPS_IO_NOTIFIED_TERMINATION_DATE</p>

Table Name	Input Values
	BEN_PPS_IN_ON_MILITARY_SERVICE, BEN_PPS_IO_ON_MILITARY_SERVICE, BEN_PPS_IN_ORIGINAL_DATE_OF_HIRE, BEN_PPS_IO_ORIGINAL_DATE_OF_HIRE, BEN_PPS_IN_PRIMARY_FLAG, BEN_PPS_IO_PRIMARY_FLAG, BEN_PPS_IN_PROJECTED_TERMINATION_DATE, BEN_PPS_IO_PROJECTED_TERMINATION_DATE, BEN_PPS_IN_WORKER_NUMBER, BEN_PPS_IO_WORKER_NUMBER, BEN_PPL_IN_PERSON_ID, BEN_PPL_IO_PERSON_ID, BEN_PPL_IN_BUSINESS_GROUP_ID, BEN_PPL_IO_BUSINESS_GROUP_ID, BEN_PPL_IN_EFFECTIVE_END_DATE, BEN_PPL_IO_EFFECTIVE_END_DATE, BEN_PPL_IN_EFFECTIVE_START_DATE, BEN_PPL_IO_EFFECTIVE_START_DATE, BEN_PPL_IN_HIGHEST_EDUCATION_LEVEL, BEN_PPL_IO_HIGHEST_EDUCATION_LEVEL, BEN_PPL_IN_LEGISLATION_CODE, BEN_PPL_IO_LEGISLATION_CODE
PER_ALL_PEOPLE_F	BEN_PPL_IN_MARITAL_STATUS, BEN_PPL_IO_MARITAL_STATUS, BEN_PPL_IN_MARITAL_STATUS_DATE,

Table Name	Input Values
	BEN_PPL_IO_MARITAL_STATUS_DATE, BEN_PPL_IN_PERSON_ID, BEN_PPL_IO_PERSON_ID, BEN_PPL_IN_PERSON_LEGISLATIVE_ID, BEN_PPL_IO_PERSON_LEGISLATIVE_ID, BEN_PPL_IN_SEX, BEN_PPL_IO_SEX, BEN_PPF_IN_ATTRIBUTE1, BEN_PPF_IO_ATTRIBUTE1, BEN_PPF_IN_ATTRIBUTE2
PER_PERSONS	BEN_PPF_IO_ATTRIBUTE2, BEN_PPF_IN_ATTRIBUTE3, BEN_PPF_IO_ATTRIBUTE3, BEN_PPF_IN_ATTRIBUTE4, BEN_PPF_IO_ATTRIBUTE4, BEN_PPF_IN_ATTRIBUTE5, BEN_PPF_IO_ATTRIBUTE5, BEN_PPF_IN_ATTRIBUTE6, BEN_PPF_IO_ATTRIBUTE6, BEN_PPF_IN_ATTRIBUTE_CATEGORY, BEN_PPF_IO_ATTRIBUTE_CATEGORY, BEN_PPF_IN_MAILING_ADDRESS_ID, BEN_PPF_IO_MAILING_ADDRESS_ID, BEN_PPF_IN_PERSON_NUMBER, BEN_PPF_IO_PERSON_NUMBER,

Table Name	Input Values
	BEN_PPF_IN_PRIMARY_EMAIL_ID, BEN_PPF_IO_PRIMARY_EMAIL_ID, BEN_PPF_IN_PRIMARY_NID_ID, BEN_PPF_IO_PRIMARY_NID_ID, BEN_PPF_IN_PRIMARY_NID_NUMBER, BEN_PPF_IO_PRIMARY_NID_NUMBER, BEN_PPF_IN_PRIMARY_PHONE_ID, BEN_PPF_IO_PRIMARY_PHONE_ID, BEN_PPF_IN_START_DATE, BEN_PPF_IO_START_DATE, BEN_PPE_IN_PERSON_ID, BEN_PPE_IO_PERSON_ID, BEN_PPE_IN_BLOOD_TYPE, BEN_PPE_IO_BLOOD_TYPE, BEN_PPE_IN_BUSINESS_GROUP_ID, BEN_PPE_IO_BUSINESS_GROUP_ID, BEN_PPE_IN_CORRESPONDENCE_LANGUAGE, BEN_PPE_IO_CORRESPONDENCE_LANGUAGE, BEN_PPE_IN_COUNTRY_OF_BIRTH, BEN_PPE_IO_COUNTRY_OF_BIRTH, BEN_PPE_IN_DATE_OF_BIRTH, BEN_PPE_IO_DATE_OF_BIRTH, BEN_PPE_IN_DATE_OF_DEATH, BEN_PPE_IO_DATE_OF_DEATH, BEN_PPE_IN_PARTY_ID,

Table Name	Input Values
	<p>BEN_PPE_IO_PARTY_ID,</p> <p>BEN_PPE_IN_PERSON_ID,</p> <p>BEN_PPE_IO_PERSON_ID,</p> <p>BEN_PPE_IN_REGION_OF_BIRTH,</p> <p>BEN_PPE_IO_REGION_OF_BIRTH,</p> <p>BEN_PPE_IN_START_DATE,</p> <p>BEN_PPE_IO_START_DATE,</p> <p>BEN_PPE_IN_TOWN_OF_BIRTH,</p> <p>BEN_PPE_IO_TOWN_OF_BIRTH,</p> <p>BEN_PPE_IN_USER_GUID,</p> <p>BEN_PPE_IO_USER_GUID</p>
PAY_ELEMENT_ENTRY_VALUES_F	<p>BEN_PEE_IN_PERSON_ID,</p> <p>BEN_PEE_IO_PERSON_ID,</p> <p>BEN_PEE_IN_EFFECTIVE_END_DATE,</p> <p>BEN_PEE_IO_EFFECTIVE_END_DATE,</p> <p>BEN_PEE_IN_EFFECTIVE_START_DATE,</p> <p>BEN_PEE_IO_EFFECTIVE_START_DATE,</p> <p>BEN_PEE_IN_ELEMENT_ENTRY_ID,</p> <p>BEN_PEE_IO_ELEMENT_ENTRY_ID,</p> <p>BEN_PEE_IN_ELEMENT_ENTRY_VALUE_ID,</p> <p>BEN_PEE_IO_ELEMENT_ENTRY_VALUE_ID,</p> <p>BEN_PEE_IN_ENTRY_USAGE_ID,</p> <p>BEN_PEE_IO_ENTRY_USAGE_ID,</p> <p>BEN_PEE_IN_INPUT_VALUE_ID,</p> <p>BEN_PEE_IO_INPUT_VALUE_ID,</p>

Table Name	Input Values
	BEN_PEE_IN_SCREEN_ENTRY_VALUE, BEN_PEE_IO_SCREEN_ENTRY_VALUE
BEN_PER_BNFTS_BAL_F	BEN_PBB_IN_PERSON_ID, BEN_PBB_IO_PERSON_ID, BEN_PBB_IN_EFFECTIVE_START_DATE, BEN_PBB_IO_EFFECTIVE_START_DATE, BEN_PBB_IN_EFFECTIVE_END_DATE, BEN_PBB_IO_EFFECTIVE_END_DATE, BEN_PBB_IN_VAL, BEN_PBB_IO_VAL, BEN_PBB_IN_BNFTS_BAL_ID, BEN_PBB_IO_BNFTS_BAL_ID, BEN_IV_LER_ID
PER_PERSON_ADDR_USAGES_F	BEN_ADD_IN_ADDRESS_ID, BEN_ADD_IO_ADDRESS_ID, BEN_ADD_IN_ADDRESS_TYPE, BEN_ADD_IO_ADDRESS_TYPE, BEN_ADD_IN_BUSINESS_GROUP_ID, BEN_ADD_IO_BUSINESS_GROUP_ID, BEN_ADD_IN_CREATED_BY, BEN_ADD_IO_CREATED_BY, BEN_ADD_IN_EFFECTIVE_END_DATE, BEN_ADD_IO_EFFECTIVE_END_DATE, BEN_ADD_IN_EFFECTIVE_START_DATE, BEN_ADD_IO_EFFECTIVE_START_DATE

Table Name	Input Values
	<p>BEN_ADD_IN_LAST_UPDATED_BY,          BEN_ADD_IO_LAST_UPDATED_BY,          BEN_ADD_IN_LAST_UPDATE_LOGIN,          BEN_ADD_IO_LAST_UPDATE_LOGIN,          BEN_ADD_IN_OBJECT_VERSION_NUMBER,          BEN_ADD_IO_OBJECT_VERSION_NUMBER,          BEN_ADD_IN_PERSON_ADDR_USAGE_ID,          BEN_ADD_IO_PERSON_ADDR_USAGE_ID,          BEN_ADD_IN_PERSON_ID,          BEN_ADD_IO_PERSON_ID</p>
BEN_PER_LE_HABITS_COV_F	<p>BEN_LEH_IN_PERSON_ID,          BEN_LEH_IO_PERSON_ID,          BEN_LEH_IN_EFFECTIVE_START_DATE,          BEN_LEH_IO_EFFECTIVE_START_DATE,          BEN_LEH_IN_EFFECTIVE_END_DATE,          BEN_LEH_IO_EFFECTIVE_END_DATE,          BEN_LEH_IN_LE_HABITS_COV_ID,          BEN_LEH_IO_LE_HABITS_COV_ID,          BEN_LEH_IN_LEGISLATIVE_DATA_GROUP_ID,          BEN_LEH_IO_LEGISLATIVE_DATA_GROUP_ID,          BEN_IV_LER_ID</p>
PER_CONTACT_RELATIONSHIPS	<p>BEN_CON_IN_BENEFICIARY_FLAG,          BEN_CON_IO_BENEFICIARY_FLAG,          BEN_CON_IN_CONTACT_PERSON_ID,</p>

Table Name	Input Values
	BEN_CON_IO_CONTACT_PERSON_ID, BEN_CON_IN_CONTACT_TYPE, BEN_CON_IO_CONTACT_TYPE, BEN_CON_IN_DEPENDENT_FLAG, BEN_CON_IO_DEPENDENT_FLAG, BEN_CON_IN_EFFECTIVE_END_DATE, BEN_CON_IO_EFFECTIVE_END_DATE, BEN_CON_IN_EFFECTIVE_START_DATE, BEN_CON_IO_EFFECTIVE_START_DATE, BEN_CON_IN_EMERGENCY_CONTACT_FLAG, BEN_CON_IO_EMERGENCY_CONTACT_FLAG, BEN_CON_IN_END_LIFE_REASON_ID, BEN_CON_IO_END_LIFE_REASON_ID, BEN_CON_IN_PERSONAL_FLAG, BEN_CON_IO_PERSONAL_FLAG, BEN_CON_IN_PERSON_ID, BEN_CON_IO_PERSON_ID, BEN_CON_IN_PRIMARY_CONTACT_FLAG, BEN_CON_IO_PRIMARY_CONTACT_FLAG, BEN_CON_IN_RLTD_PER_RSRS_W_DSGNTR_FLAG, BEN_CON_IO_RLTD_PER_RSRS_W_DSGNTR_FLAG, BEN_CON_IN_START_LIFE_REASON_ID, BEN_CON_IO_START_LIFE_REASON_ID, BEN_CON_IN_THIRD_PARTY_PAY_FLAG, BEN_CON_IO_THIRD_PARTY_PAY_FLAG

Table Name	Input Values
BEN_BENEFIT_RELATIONS_F	<p>BEN_BRE_IN_PERSON_ID, BEN_BRE_IO_PERSON_ID, BEN_BRE_IN_EFFECTIVE_START_DATE, BEN_BRE_IO_EFFECTIVE_START_DATE, BEN_BRE_IN_EFFECTIVE_END_DATE, BEN_BRE_IO_EFFECTIVE_END_DATE, BEN_BRE_IN_BENEFIT_RELATION_ID, BEN_BRE_IO_BENEFIT_RELATION_ID, BEN_BRE_IN_LEGAL_ENTITY_ID, BEN_BRE_IO_LEGAL_ENTITY_ID, BEN_IV_LER_ID, BEN_IN_UPDATED_SOURCE, BEN_IO_UPDATED_SOURCE, BEN_IN_ORIGINAL_SOURCE, BEN_IO_ORIGINAL_SOURCE</p>
BEN_PER_BENEFIT_GROUP_F	<p>BEN_BBG_IN_PERSON_ID, BEN_BBG_IO_PERSON_ID, BEN_bbg_IN_EFFECTIVE_START_DATE, BEN_bbg_IO_EFFECTIVE_START_DATE, BEN_bbg_IN_EFFECTIVE_END_DATE, BEN_bbg_IO_EFFECTIVE_END_DATE, BEN_bbg_IN_LE_BENEFIT_GROUP_ID, BEN_bbg_IO_BENEFIT_GROUP_ID, BEN_bbg_IN_LEGISLATIVE_DATA_GROUP_ID, BEN_bbg_IO_LEGISLATIVE_DATA_GROUP_ID,</p>

Table Name	Input Values
	BEN_IV_LER_ID
PER_ASSIGN_WORK_MEASURES_F	BEN_AWM_IN_PERSON_ID, BEN_AWM_IO_PERSON_ID, BEN_AWM_IN_ACTION_OCCURRENCE_ID, BEN_AWM_IO_ACTION_OCCURRENCE_ID, BEN_AWM_IN_ADDS_TO_BUDGET, BEN_AWM_IO_ADDS_TO_BUDGET, BEN_AWM_IN_ASSIGNMENT_ID, BEN_AWM_IO_ASSIGNMENT_ID, BEN_AWM_IN_ASSIGN_WORK_MEASURE_ID, BEN_AWM_IO_ASSIGN_WORK_MEASURE_ID, BEN_AWM_IN_BUSINESS_GROUP_ID, BEN_AWM_IO_BUSINESS_GROUP_ID, BEN_AWM_IN_EFFECTIVE_END_DATE, BEN_AWM_IO_EFFECTIVE_END_DATE, BEN_AWM_IN_EFFECTIVE_START_DATE, BEN_AWM_IO_EFFECTIVE_START_DATE, BEN_AWM_IN_UNIT, BEN_AWM_IO_UNIT, BEN_AWM_IN_VALUE, BEN_AWM_IO_VALUE
PER_ALL_ASSIGNMENTS_M	BEN_ASG_IN_EFFECTIVE_START_DATE, BEN_ASG_IO_EFFECTIVE_START_DATE, BEN_ASG_IN_EFFECTIVE_END_DATE,

Table Name	Input Values
	BEN_ASG_IO_EFFECTIVE_END_DATE, BEN_ASG_IN_ORGANIZATION_ID, BEN_ASG_IO_ORGANIZATION_ID, BEN_ASG_IN_GRADE_ID, BEN_ASG_IO_GRADE_ID, BEN_ASG_IN_JOB_ID, BEN_ASG_IO_JOB_ID, BEN_ASG_IN_WORK_TERMS_ASSIGNMENT_ID, BEN_ASG_IO_WORK_TERMS_ASSIGNMENT_ID, BEN_ASG_IN_LOCATION_ID, BEN_ASG_IO_LOCATION_ID, BEN_ASG_IN_ASSIGNMENT_STATUS_TYPE_ID, BEN_ASG_IO_ASSIGNMENT_STATUS_TYPE_ID, BEN_ASG_IN_ASSIGNMENT_TYPE, BEN_ASG_IO_ASSIGNMENT_TYPE, BEN_ASG_IN_PRIMARY_FLAG, BEN_ASG_IO_PRIMARY_FLAG, BEN_ASG_IN_EMPLOYMENT_CATEGORY, BEN_ASG_IO_EMPLOYMENT_CATEGORY, BEN_ASG_IN_BARGAINING_UNIT_CODE, BEN_ASG_IO_BARGAINING_UNIT_CODE, BEN_ASG_IN_HOURLY_SALARIED_CODE, BEN_ASG_IO_HOURLY_SALARIED_CODE, BEN_ASG_IN_NORMAL_HOURS, BEN_ASG_IO_NORMAL_HOURS,

Table Name	Input Values
	BEN_IV_LER_ID BEN_ASG_IN_COLLECTIVE AGREEMENT_ID BEN_ASG_IO_COLLECTIVE AGREEMENT_ID BEN_ASG_IN_REASON_CODE BEN_ASG_IO_REASON_CODE BEN_ASG_IN_ASSIGNMENT_NUMBER BEN_ASG_IO_ASSIGNMENT_NUMBER BEN_ASG_IN_ASSIGNMENT_SEQUENCE BEN_ASG_IO_ASSIGNMENT_SEQUENCE BEN_ASG_IN_ACTION_OCCURRENCE_ID BEN_ASG_IO_ACTION_OCCURRENCE_ID BEN_ASG_IN_CONTRACT_ID BEN_ASG_IO_CONTRACT_ID BEN_ASG_IN_FULL_PART_TIME BEN_ASG_IO_FULL_PART_TIME BEN_ASG_IN_GRADE_LADDER_PGM_ID BEN_ASG_IO_GRADE_LADDER_PGM_ID BEN_ASG_IN_LEGAL_ENTITY_ID BEN_ASG_IO_LEGAL_ENTITY_ID BEN_ASG_IN_LEGISLATION_CODE BEN_ASG_IO_LEGISLATION_CODE BEN_ASG_IN_MANAGER_FLAG BEN_ASG_IO_MANAGER_FLAG BEN_ASG_IN_PERMANENT_TEMPORARY_FLAG BEN_ASG_IO_PERMANENT_TEMPORARY_FLAG

Table Name	Input Values
	BEN_ASG_IN_PRIMARY_ASSIGNMENT_FLAG BEN_ASG_IO_PRIMARY_ASSIGNMENT_FLAG BEN_ASG_IN_PRIMARY_WORK_RELATION_FLAG BEN_ASG_IO_PRIMARY_WORK_RELATION_FLAG BEN_ASG_IN_PRIMARY_WORK_TERMS_FLAG BEN_ASG_IO_PRIMARY_WORK_TERMS_FLAG BEN_ASG_IN_RETIREMENT_DATE BEN_ASG_IO_RETIREMENT_DATE BEN_ASG_IN_SYSTEM_PERSON_TYPE BEN_ASG_IO_SYSTEM_PERSON_TYPE BEN_ASG_IN_UNION_ID BEN_ASG_IO_UNION_ID
PER_ADDRESSES_F	BEN_ADD_IN_ADDRESS_ID, BEN_ADD_IO_ADDRESS_ID, BEN_ADD_IN_BUSINESS_GROUP_ID, BEN_ADD_IO_BUSINESS_GROUP_ID, BEN_ADD_IN_EFFECTIVE_END_DATE, BEN_ADD_IO_EFFECTIVE_END_DATE, BEN_ADD_IN_EFFECTIVE_START_DATE, BEN_ADD_IO_EFFECTIVE_START_DATE
PER_ABSENCE_ATTENDANCES	BEN_ABA_IN_SICKNESS_START_DATE BEN_ABA_IN_SICKNESS_END_DATE BEN_ABA_IN_DATE_NOTIFICATION BEN_ABA_IO_DATE_NOTIFICATION BEN_ABA_IN_DATE_PROJECTED_END

Table Name	Input Values
	BEN_ABA_IO_DATE_PROJECTED_END BEN_ABA_IN_DATE_PROJECTED_START BEN_ABA_IO_DATE_PROJECTED_START BEN_ABA_IN_TIME_END BEN_ABA_IO_TIME_END BEN_ABA_IN_TIME_PROJECTED_END BEN_ABA_IO_TIME_PROJECTED_END BEN_ABA_IN_TIME_PROJECTED_START BEN_ABA_IO_TIME_PROJECTED_START BEN_LER_ID BEN_ABA_IN_ABS_INFORMATION_CATEGORY BEN_ABA_IN_ABS_INFORMATION1 BEN_ABA_IN_ABS_INFORMATION2 BEN_ABA_IN_ABS_INFORMATION3 BEN_ABA_IN_ABS_INFORMATION4 BEN_ABA_IN_ABS_INFORMATION5 BEN_ABA_IN_ABS_INFORMATION6 BEN_ABA_IN_ABS_INFORMATION7 BEN_ABA_IN_ABS_INFORMATION8 BEN_ABA_IN_ABS_INFORMATION9 BEN_ABA_IN_ABS_INFORMATION10 BEN_ABA_IN_ABS_INFORMATION11 BEN_ABA_IN_ABS_INFORMATION12 BEN_ABA_IN_ABS_INFORMATION13 BEN_ABA_IN_ABS_INFORMATION14

Table Name	Input Values
	BEN_ABA_IN_ABS_INFORMATION15 BEN_ABA_IN_ABS_INFORMATION16 BEN_ABA_IN_ABS_INFORMATION17 BEN_ABA_IN_ABS_INFORMATION18 BEN_ABA_IN_ABS_INFORMATION19 BEN_ABA_IN_ABS_INFORMATION20 BEN_ABA_IN_ABS_INFORMATION21 BEN_ABA_IN_ABS_INFORMATION22 BEN_ABA_IN_ABS_INFORMATION23 BEN_ABA_IN_ABS_INFORMATION24 BEN_ABA_IN_ABS_INFORMATION25 BEN_ABA_IN_ABS_INFORMATION26 BEN_ABA_IN_ABS_INFORMATION27 BEN_ABA_IN_ABS_INFORMATION28 BEN_ABA_IN_ABS_INFORMATION29 BEN_ABA_IN_ABS_INFORMATION30 BEN_ABA_IN_ATTRIBUTE_CATEGORY BEN_ABA_IN_ATTRIBUTE1 BEN_ABA_IN_ATTRIBUTE2 BEN_ABA_IN_ATTRIBUTE3 BEN_ABA_IN_ATTRIBUTE4 BEN_ABA_IN_ATTRIBUTE5 BEN_ABA_IN_ATTRIBUTE6 BEN_ABA_IN_ATTRIBUTE7 BEN_ABA_IN_ATTRIBUTE8

Table Name	Input Values
	<p>BEN_ABA_IN_ATTRIBUTE9</p> <p>BEN_ABA_IN_ATTRIBUTE10</p> <p>BEN_ABA_IN_ATTRIBUTE11</p> <p>BEN_ABA_IN_ATTRIBUTE12</p> <p>BEN_ABA_IN_ATTRIBUTE13</p> <p>BEN_ABA_IN_ATTRIBUTE14</p> <p>BEN_ABA_IN_ATTRIBUTE15</p> <p>BEN_ABA_IN_ATTRIBUTE16</p> <p>BEN_ABA_IN_ATTRIBUTE17</p> <p>BEN_ABA_IN_ATTRIBUTE18</p> <p>BEN_ABA_IN_ATTRIBUTE19</p> <p>BEN_ABA_IN_ATTRIBUTE20</p> <p>BEN_ABA_IO_ABS_INFORMATION_CATEGORY</p> <p>BEN_ABA_IO_ABS_INFORMATION1</p> <p>BEN_ABA_IO_ABS_INFORMATION2</p> <p>BEN_ABA_IO_ABS_INFORMATION3</p> <p>BEN_ABA_IO_ABS_INFORMATION4</p> <p>BEN_ABA_IO_ABS_INFORMATION5</p> <p>BEN_ABA_IO_ABS_INFORMATION6</p> <p>BEN_ABA_IO_ABS_INFORMATION7</p> <p>BEN_ABA_IO_ABS_INFORMATION8</p> <p>BEN_ABA_IO_ABS_INFORMATION9</p> <p>BEN_ABA_IO_ABS_INFORMATION10</p> <p>BEN_ABA_IO_ABS_INFORMATION11</p> <p>BEN_ABA_IO_ABS_INFORMATION12</p>

Table Name	Input Values
	<p>BEN_ABA_IO_ABS_INFORMATION13</p> <p>BEN_ABA_IO_ABS_INFORMATION14</p> <p>BEN_ABA_IO_ABS_INFORMATION15</p> <p>BEN_ABA_IO_ABS_INFORMATION16</p> <p>BEN_ABA_IO_ABS_INFORMATION17</p> <p>BEN_ABA_IO_ABS_INFORMATION18</p> <p>BEN_ABA_IO_ABS_INFORMATION19</p> <p>BEN_ABA_IO_ABS_INFORMATION20</p> <p>BEN_ABA_IO_ABS_INFORMATION21</p> <p>BEN_ABA_IO_ABS_INFORMATION22</p> <p>BEN_ABA_IO_ABS_INFORMATION23</p> <p>BEN_ABA_IO_ABS_INFORMATION24</p> <p>BEN_ABA_IO_ABS_INFORMATION25</p> <p>BEN_ABA_IO_ABS_INFORMATION26</p> <p>BEN_ABA_IO_ABS_INFORMATION27</p> <p>BEN_ABA_IO_ABS_INFORMATION28</p> <p>BEN_ABA_IO_ABS_INFORMATION29</p> <p>BEN_ABA_IO_ABS_INFORMATION30</p> <p>BEN_ABA_IO_ATTRIBUTE_CATEGORY</p> <p>BEN_ABA_IO_ATTRIBUTE1</p> <p>BEN_ABA_IO_ATTRIBUTE2</p> <p>BEN_ABA_IO_ATTRIBUTE3</p> <p>BEN_ABA_IO_ATTRIBUTE4</p> <p>BEN_ABA_IO_ATTRIBUTE5</p> <p>BEN_ABA_IO_ATTRIBUTE6</p>

Table Name	Input Values
	BEN_ABA_IO_ATTRIBUTE7 BEN_ABA_IO_ATTRIBUTE8 BEN_ABA_IO_ATTRIBUTE9 BEN_ABA_IO_ATTRIBUTE10 BEN_ABA_IO_ATTRIBUTE11 BEN_ABA_IO_ATTRIBUTE12 BEN_ABA_IO_ATTRIBUTE13 BEN_ABA_IO_ATTRIBUTE14 BEN_ABA_IO_ATTRIBUTE15 BEN_ABA_IO_ATTRIBUTE16 BEN_ABA_IO_ATTRIBUTE17 BEN_ABA_IO_ATTRIBUTE18 BEN_ABA_IO_ATTRIBUTE19 BEN_ABA_IO_ATTRIBUTE20

## Appendix 3: Formula Function

Here's how you use the BEN\_FN\_GET\_CHAR\_VALUE formula function.

### Mandatory Contexts:

- HR\_ASSIGNMENT\_ID
- EFFECTIVE\_DATE
- BUSINESS\_GROUP\_ID

### Input Parameters:

Data type is Char for all parameters:

- TABLE\_NAME

- COLUMN\_NAME
- PLAN\_NAME
- OPTION\_NAME
- ADD\_KEY\_NAME
- ADD\_KEY\_VALUE

The ADD\_KEY\_NAME and ADD\_KEY\_VALUE parameters are conditionally supported for limited use cases. Refer to example 3.

## Return Variables

Variable name could be anything. Data type is CHAR.

### Sample Function Call:

Example 1 : ben\_fn\_get\_char\_value('BEN\_PRTT\_ENRT\_RSLT', 'ENROLLED', 'Medical', 'Employee')

Example 2 : l\_if\_pl\_enrolled = ben\_fn\_get\_char\_value('BEN\_PRTT\_ENRT\_RSLT', 'ENROLLED', 'med\_plan', 'NA')

Example 3:

l\_dpnt\_ctfn=BEN\_FN\_GET\_CHAR\_VALUE('BEN\_CVRD\_DPNT\_CTFN\_PRVDD','PROVIDED','PlanA','OptionA',to\_char(l\_dpnt\_id),'Birth certificate')

## Table And Columns Supported By This Function

### ***Table and columns supported by BEN\_FN\_GET\_CHAR\_VALUE***

Table Name	Column Name	Usage
PER_JOB_EXTRA_INFO_F	JEI_ATTRIBUTE1, JEI_ATTRIBUTE2, JEI_ATTRIBUTE3, JEI_ATTRIBUTE4, JEI_ATTRIBUTE5, JEI_ATTRIBUTE6, JEI_ATTRIBUTE7, JEI_ATTRIBUTE8, JEI_ATTRIBUTE9, JEI_ATTRIBUTE10, JEI_ATTRIBUTE11,	BEN_FN_GET_CHAR_VALUE('PER_JOB_EXTRA_INFO_F','JEI_ATTRIBUTE1')  This function call returns the value stored in JEI_ATTRIBUTE1 of the table PER_JOB_EXTRA_INFO_F.

Table Name	Column Name	Usage
	JEI_ATTRIBUTE12, JEI_ATTRIBUTE13, JEI_ATTRIBUTE14, JEI_ATTRIBUTE15, JEI_ATTRIBUTE16, JEI_ATTRIBUTE17, JEI_ATTRIBUTE18, JEI_ATTRIBUTE19, JEI_ATTRIBUTE20, JEI_ATTRIBUTE21, JEI_ATTRIBUTE22, JEI_ATTRIBUTE23, JEI_ATTRIBUTE24, JEI_ATTRIBUTE25, JEI_ATTRIBUTE26, JEI_ATTRIBUTE27, JEI_ATTRIBUTE28, JEI_ATTRIBUTE29, JEI_ATTRIBUTE30, JEI_INFORMATION1, JEI_INFORMATION2, JEI_INFORMATION3, JEI_INFORMATION4, JEI_INFORMATION5, JEI_INFORMATION6,	

Table Name	Column Name	Usage
	JEI_INFORMATION7, JEI_INFORMATION8, JEI_INFORMATION9, JEI_INFORMATION10, JEI_INFORMATION11, JEI_INFORMATION12, JEI_INFORMATION13, JEI_INFORMATION14, JEI_INFORMATION15, JEI_INFORMATION16, JEI_INFORMATION17, JEI_INFORMATION18, JEI_INFORMATION19, JEI_INFORMATION20, JEI_INFORMATION21, JEI_INFORMATION22, JEI_INFORMATION23, JEI_INFORMATION24, JEI_INFORMATION25, JEI_INFORMATION26, JEI_INFORMATION27, JEI_INFORMATION28, JEI_INFORMATION29, JEI_INFORMATION30	

Table Name	Column Name	Usage
BEN_PRTT_ENRT_RSLT	ADMIN_CATEGORY_CD, BNFT_AMT, BNFT_NNMNTRY_UOM, BNFT_TYP_CD, COMP_LVL_CD, ENROLLED, ENRT_CVG_STRT_DT, ENRT_CVG_THRU_DT, IMPTD_INCM_CALC_CD, PRTT_ENRT_RSLT_ID, RPLCS_SSPNDD_RSLT_ID, SS_CATEGORY_CD, SSPNDD_FLAG, SVNGS_PLN_FLAG, UOM,ORGNL_ENRT_DT, ERLST_DEENRT_DT, ENRT_OVRID_THRU_DT, INTERIM_FLAG, ENRT_OVRIDN_FLAG, ENRT_MTHD_CD	BEN_FN_GET_CHAR_VALUE('BEN_PRTT_ENRT_RSLT','ENROLLED','Healthy HMO' /* plan name */,'Employee Only' /* option name */)  This function call will return Y if the participant is enrolled in the plan (Healthy HMO) and option (Employee Only). If the participant is not enrolled then 'N' will be the return value.
BEN_ELIG_CVRD_DPNT	ENROLLED, CVG_STRT_DT, CVG_THRU_DT, DPNT_PERSON_ID, RLNSHP_CD	BEN_FN_GET_CHAR_VALUE('BEN_ELIG_CVRD_DPNT','ENROLLED','Employee Basic Life' /* Plan Name */,\$50,000' /* Option Name */,'DEPENDENT_FULL_NAME' /* Key name */,'Jones, Fred' /* dependent name (value) */)  This function call returns 'Y' if the dependent is enrolled by the participant in the plan (Employee Basic Life) and in option (\$50,

Table Name	Column Name	Usage
		000). If there is no enrollment found then 'N' is returned.
BEN_ELIG_DPNT	CVRD_FLAG, DPNT_INELIG_FLAG, RLNSHP_CD, ELIG_STRT_DT, ELIG_THRU_DT, OVRDN_FLAG, INELG_RSN_CD, ELIG_PER_ELCTBL_CHC_ID, PER_IN_LER_ID, ELIG_CVRD_DPNT_ID, INELG_RSN_CD, DPNT_PERSON_ID, CTFN_REQD_FLAG, PCP_DSGN_CD	BEN_FN_GET_CHAR_VALUE('BEN_ELIG_DPNT','CVRD_FLAG','Employee Basic Life' /* Plan Name */,'\$50,000' /* Option Name */,'DEPENDENT_FULL_NAME' /* Key name */,'Jones, Fred' /* dependent name (value) */)  This function call returns 'Y' if the dependent is eligible to participate in plan (Employee Basic Life) and option (\$50,000). If there is no eligible dependent record found then 'N' is returned.
BEN_CVRD_DPNT_CTFN_PRVDD	PROVIDED	BEN_FN_GET_CHAR_VALUE('BEN_CVRD_DPNT_CTFN_PRVDD','PROVIDED','Healthy HMO' /* Plan Name */,'Employee + children' /* Option Name. Pass 'NA' if the plan is not associated with any option */,'to_char(l_dpnt_id) /* personId of the dependent */,'Legal custody certificate' /* certification type */)  This function call returns 'Y' if a certification of type (Legal custody certificate) is provided by the participant for the dependent identified by personId.  Possible certification types are: <ul style="list-style-type: none"> <li>• Adoption certificate</li> <li>• Birth certificate</li> <li>• Domestic partner certificate</li> </ul>

Table Name	Column Name	Usage
		<ul style="list-style-type: none"> <li>• Legal custody certificate</li> <li>• Marriage certificate</li> </ul> <p>Alternatively, the following parameters can be used to check if the certification was provided in a previous life event for the dependent. The determinant NA_CHECK_ONCE should be passed as the plan name.</p> <pre>BEN_FN_GET_CHAR_VALUE('BEN_CVRD_DPNT_CTFN_PRVDD','PROVIDED','NA_CHECK_ONCE' /* determinant */,'/* space */,to_char(l_dpnt_id) /* personId of the dependent */,'Legal custody certificate' /* certification type */)</pre>
BEN_PRTT_ENRT_CTFN_PRVDD	ENRT_CTFN_RECV_DT, ENRT_CTFN_DND_DT	<pre>BEN_FN_GET_CHAR_VALUE('BEN_PRTT_ENRT_CTFN_PRVDD','ENRT_CTFN_RECV_DT','Employee Supplemental Life' /* Plan Name */,'NA' /* you want to check against any option within the plan */,'Proof of good health' /* certification type */)</pre> <p>This function call returns the certification received date (of the participant) for the plan (Employee Supplemental Life) and any option under it. Possible certification types that can be checked for are:</p> <ul style="list-style-type: none"> <li>• Domestic partner affidavit</li> <li>• Proof of external coverage</li> <li>• Proof of good health</li> <li>• Proof of other coverage</li> </ul>
PER_CONTACT_RELSHIPS_F	DAUGHTER_COUNT, OTHERS_COUNT, SON_COUNT, SPOUSE_COUNT, DP_COUNT, DP_MALE_COUNT, DP_FEMALE_COUNT,	<pre>BEN_FN_GET_CHAR_VALUE('PER_CONTACT_RELSHIPS_F','DAUGHTER_COUNT','/* space (not used) */,'/* space (not used) */,'LEGISLATION_CODE' /* key */,'US' /* Pass the legislation code of the participant */)</pre> <p>This function call returns the aggregate counts of the dependents based on the contact type.</p>

Table Name	Column Name	Usage
	ADOPTED_CHILD_COUNT,  STEP_CHILD_COUNT,  FOSTER_CHILD_COUNT	
BEN_PER_IN_LER	PER_IN_LER_ID,  BENEFIT_RELATION_ID,  LF_EVT_OCRD_DT,  PROCD_DT,  STRTD_DT,  VOIDD_DT,  PERSON_ID,  BCKT_DT,  CLSD_DT,  NTFN_DT,  LEGAL_ENTITY_ID,  BENEFIT_REL_SYSTEM_CD,  PRIMARY_REL,  REL_PRMRY_ASG_ID,  NAME,  LER_ID	<p>BEN_FN_GET_CHAR_VALUE('BEN_PER_IN_LER','LF_EVT_OCRD_DT','' /* space (not used) */,'' /* space (not used) */,'BENEFIT_RELATION_NAME' /* key */,'Default' /* benefit relation name as seen in the UI */)</p> <p>This function call returns life event date of the started event corresponding to the benefit relation name which is passed as a parameter.</p>
PAY_ELEMENT_ENTRIES_F	AGGREGATE	<p>BEN_FN_GET_CHAR_VALUE('PAY_ELEMENT_ENTRIES_F','ER' /* element name */,'VAL' /* input value name */,'2014/01/01' /* from date */,'2014/12/31' /* to date */)</p> <p>This function call returns the aggregate of the element entry values for an element for the date range specified by from and to dates.</p>

Table Name	Column Name	Usage
BEN_PRTT_RT_VAL	AGG_RT_VAL, AGG_CMCD_RT_VAL	<pre>BEN_FN_GET_CHAR_VALUE('BEN_PRTT_RT_VAL','AGG_RT_VAL','Healthy HMO' /* plan name */,'Employee Only' /* option name */,'2014/01/01' /* from date */,'2014/12/31' /* to date */)</pre> <p>This function call returns the aggregate of the rates for the enrollment in Healthy HMO:Employee Only for the date range specified by from and to dates. All rates including primary/secondary/others will be considered.</p>
BEN_PRTT_RT_VAL	AGG_RT_VAL, AGG_CMCD_RT_VAL	<pre>BEN_FN_GET_CHAR_VALUE('BEN_PRTT_RT_VAL','AGG_RT_VAL','ACTIVITY_BASE_RATE' /* determinant to indicate that only a single rate should be summed up */,'Healthy_HMO_Employee_only_Vision' /* rate name */,'2014/01/01' /* from date */,'2014/12/31' /* to date */)</pre> <p>This function call returns the aggregate of the rate specified by the activity base rate name for the date range specified by from and to dates.</p>
PAY_TIME_PERIODS	PAYROLL_ID, PAYROLL_TYPE, PERIOD_NUM, PERIOD_NAME, START_DATE, END_DATE, CUT_OFF_DATE, DEFAULT_PAYDATE, TOTAL_TP_IN_CALENDAR_YEAR	<pre>BEN_FN_GET_CHAR_VALUE('PAY_TIME_PERIODS','DEFAULT_PAYDATE',' ' /* space (not used) */,' ' /* space (not used) */,'START_TIME' /* key */,'2014/12/05' /* date used to identify the pay period */)</pre> <p>This function call returns the pay date of the corresponding participant's payroll. The value for the START_TIME identifies the date which is used to identify the pay period. For example if the START_TIME 2014/12/05 and the participant is on a monthly payroll, then the default pay date for the December time period is returned.</p>

## Appendix 4: Formula Errors: Compile time

Here are the descriptions of the various formula errors during the compile time:

### **Formula errors table**

Formula Error	Description
Syntax Error	The formula text violates the grammatical rules for the formula language. Example: Using IF1 instead of IF for an IF statement.
Incorrect Statement Order	ALIAS, DEFAULT, or INPUT statements appear after other statements.
Misuse of ASSIGNMENT Statement	An ASSIGNMENT assigns a value to a database item. A context is assigned a value externally to a CHANGE-CONTEXTS statement. A non-context variable is assigned a value within a CHANGE-CONTEXTS statement.
Misuse of ALIAS Statement	An ALIAS statement may only be used for a database item.
Missing DEFAULT Statement	A database item with defaulting specified must have a DEFAULT statement.
Misuse of DEFAULT Statement	A DEFAULT statement is specified for a variable other than an input or database item.
Uninitialized Variable	The compiler detects that a variable is uninitialized when used. The compiler cannot do in all cases. This error often occurs when the intention is to use a database item, but the database item is not available to the formula (it does not exist, or partitioning restrictions mean it's not available to the formula).
Missing Function Call	A function call is not recognized. The combination of return type, function name, and parameter types does not match any available function.
Incorrect Operator Usage	An instance of formula operator use does not match the permitted uses of that operator. Example: The + operator has two permitted uses. The operands are both of data type NUMBER, or both of data type TEXT.
Inconsistent Data Type Usage	A formula variable is being used as if it is of more than one data type. Example: Variable A is assigned a NUMBER value at the start of the formula, but a TEXT value later in the formula.  A database item or context is being used with the wrong data type.
EXIT Statement Not Within WHILE Loop	
Mixed Use of Context	A variable is later used as a context or vice versa. Example: AREA1 is assigned a value so it's used as an ordinary variable, but later AREA1 is used as a context in a GET_CONTEXT call.

## Appendix 5: Formula Errors: Run time

Here are the descriptions of the various formula errors during the run time:

### **Formula errors table**

Formula Error	Description
Uninitialized Context	A context is not set when required for a database item or function call.
Uninitialized Variable	Where the formula compiler cannot fully determine whether or not a variable or context is initialized when used, it generates code to test whether or not the variable is initialized. When the formula executes and the variable or context is not initialized an error is raised.
Divide by Zero	This error is raised when a numeric value is divided by zero.
No Data Found	This error is raised when a database item of a non-array type unexpectedly fails to return any data. If the database item can return no data then it should allow defaulting. This error can also be raised from within a formula function. The cause is an error in the formula function code.
Too Many Rows	This error is raised when a database item of a non-array type unexpectedly returns more than a single row of data. The cause is an incorrect assumption made about the data being accessed. This error can also be raised from within a formula function. The cause is an error in the formula function code.
NULL Data Found	This error is raised when a database item unexpectedly returns a NULL data value. If the database item can return a NULL value then it should allow defaulting.
Value Exceeded Allowable Range	This error is raised for a variety of reasons such as exceeding the maximum allowable length of a string.
Invalid Number	This error is raised when an attempt is made to convert a non-numeric string to a number.
User Defined Function Error	This error is raised from within a formula function. The error message text is output as part of the formula error message.
External Function Call Error	A formula function returned an error, but did not provide any additional information to the formula code. The function might have output error information to the logging destination for the executing code.
Function Returned NULL Value	A formula function returned a NULL value.
Too Many Iterations	A single WHILE loop, or a combination of WHILE loops, has exceeded the maximum number of permitted iterations. The error is raised to terminate loops that could go on forever. This indicates a programming error within the formula.
Array Data Value Not Set	The formula attempted to access an array index that has no data value. This is an error in the formula code.
Invalid Type Parameter for WSA_EXISTS	An invalid data type was specified in the WSA_EXISTS call.
Incorrect Data Type For Stored Item	When retrieving an item using WSA_GET, the item's actual data type does not match that of the stored item. This is an error within the calling formula.

Formula Error	Description
Called Formula Not Found	The called formula could not be resolved when attempting to call a formula from a formula. This could be due to an error in the calling formula, or because of installation issues.
Recursive Formula Call	An attempt was made to call a formula from itself. The call could be directly or indirectly via another called formula. Recursive formula calling is not permitted.
Input Has Different Types In Called and Calling Formulas	When calling a formula from a formula, the actual formula input data type within the called formula does not match the data type specified from the calling formula.
Output Has Different Types In Called and Calling Formulas	When calling a formula from a formula, the actual formula output data type within the called formula does not match the data type specified from the calling formula.
Too Many Formula Calls	There are too many formula from formula calls. This is a problem with the formulas.

## Appendix 6: Database Items

The following table lists newly created database items that can be used in release 21A onward.

### ***Database items table***

User Entity	Required Contexts	Array Database Item	Database Item Name
BEN_ELIG_PER_ELCTBL_CHC_TN_UE	HR_ASSIGNMENT_ID, BUSINESS_GROUP_ID, EFFECTIVE_DATE	Yes	BEN_EPE_INTERIM_FLAG_TN BEN_EPE_LER_CHG_DPNT_CVG_CD_TN BEN_EPE_MNDTRY_FLAG_TN BEN_EPE_MUST_ENRL_ANTHR_PL_ID_NN BEN_EPE_OIPL_ID_NN BEN_EPE_PER_IN_LER_ID_NN BEN_EPE_PGM_ID_NN BEN_EPE_PIL_ELCTBL_CHC_POPL_ID_NN BEN_EPE_PLIP_ID_NN BEN_EPE_PL_ID_NN BEN_EPE_PL_TYP_ID_NN BEN_EPE_PROCG_END_DT_DN BEN_EPE_PRTT_ENRT_RSLT_ID_NN BEN_EPE_PTIP_ID_NN BEN_EPE_ROLL_CRS_FLAG_TN

User Entity	Required Contexts	Array Database Item	Database Item Name
			BEN_EPE_SPCL_RT_OIPL_ID_NN BEN_EPE_SPCL_RT_PL_ID_NN BEN_EPE_YR_PERD_ID_NN BEN_EPE_ALWS_DPNT_DSGN_FLAG_TN BEN_EPE_APPROVAL_STATUS_CD_TN BEN_EPE_AUTO_ENRT_FLAG_TN BEN_EPE_BNFT_PRVDR_POOL_ID_NN BEN_EPE_BNF_DSGN_CD_TN BEN_EPE_BUSINESS_GROUP_ID_NN BEN_EPE_CMBN_PLIP_ID_NN BEN_EPE_CMBN_PTIP_ID_NN BEN_EPE_CMBN_PTIP_OPT_ID_NN BEN_EPE_COMP_LVL_CD_TN BEN_EPE_CRNTLY_ENRD_FLAG_TN BEN_EPE_CTFN_RQD_FLAG_TN BEN_EPE_CVRD_FLAG_TN BEN_EPE_DFLT_FLAG_TN BEN_EPE_DPNT_CVG_STRT_DT_CD_TN BEN_EPE_DPNT_CVG_STRT_DT_RL_NN BEN_EPE_DPNT_DSGN_CD_TN BEN_EPE_ELCTBL_FLAG_TN BEN_EPE_ELIG_FLAG_TN BEN_EPE_ELIG_PER_ELCTBL_CHC_ID_NN BEN_EPE_ENRT_CVG_STRT_DT_CD_TN BEN_EPE_ENRT_CVG_STRT_DT_DN BEN_EPE_ENRT_CVG_STRT_DT_RL_NN

User Entity	Required Contexts	Array Database Item	Database Item Name
			BEN_EPE_ERLST_DEENRT_DT_DN BEN_EPE_INELIG_RSN_CD_TN BEN_EPE_LER_NAME_TN BEN_EPE_LER_INSTRUCTION_TEXT_TN BEN_EPE_LER_ID_NN BEN_EPE_LER_GLOBAL_FLAG_TN BEN_EPE_LER_DESC_TXT_TN BEN_EPE_LER_QUALG_EVT_FLAG_TN BEN_EPE_LER_SELF_ASSIGNED_EVENT_FLAG_TN BEN_EPE_LER_SHORT_CODE_TN BEN_EPE_LER_SHORT_NAME_TN BEN_EPE_LER_SLCTBL_SLF_SVC_CD_TN BEN_EPE_LER_TYP_CD_TN BEN_EPE_OPT_NAME_TN BEN_EPE_PGM_NAME_TN BEN_EPE_PGM_GLOBAL_FLAG_TN BEN_EPE_PGM_EFFECTIVE_START_DATE_DN BEN_EPE_PGM_EFFECTIVE_END_DATE_DN BEN_EPE_PGM_ALWS_UNRSTRCTD_ENRT_FLAG_TN BEN_EPE_PGM_SHORT_CODE_TN BEN_EPE_PGM_RATE_LEVEL_TN BEN_EPE_PGM_STAT_CD_TN BEN_EPE_PGM_SHORT_NAME_TN BEN_EPE_PL_NAME_TN BEN_EPE_PL_RATE_LEVEL_TN BEN_EPE_PL_ALWS_UNRSTRCTD_ENRT_FLAG_TN BEN_EPE_PL_GLOBAL_FLAG_TN

User Entity	Required Contexts	Array Database Item	Database Item Name
			BEN_EPE_PL_EFFECTIVE_START_DATE_DN BEN_EPE_PL_SHORT_NAME_TN BEN_EPE_PL_STAT_CD_TN BEN_EPE_PL_FUNCTION_CODE_TN BEN_EPE_PL_EFFECTIVE_END_DATE_DN BEN_EPE_PL_SHORT_CODE_TN BEN_EPE_PTP_SS_CATEGORY_CD_TN BEN_EPE_PTP_NAME_TN BEN_EPE_PTP_GLOBAL_FLAG_TN BEN_EPE_PTP_CARRIER_PLAN_TYPE_NAME_TN BEN_EPE_PTP_ADMIN_CATEGORY_CD_TN
BEN_PRTT_LEG_DISCMR_ACTN_TN	PERSON_ID, PER_IN_LER_ID, BUSINESS_GROUP_ID, EFFECTIVE_DATE	Yes	BEN_PLD_ACTION_TIME_DN BEN_PLD_STATUS_TN BEN_PLD_BENEFIT_RELATION_NAME_TN BEN_PLD_LEGAL_DISCLAIMER_DESC_TN BEN_PLD_LER_NAME_TN
BEN_PER_BNFTS_BAL_F_TN	EFFECTIVE_DATE, BUSINESS_GROUP_ID, PERSON_ID	Yes	BEN_PBB_BNFTS_BAL_NAME_TN BEN_PBB_BUSINESS_GROUP_ID_NN BEN_PBB_VAL_NN BEN_PBB_EFFECTIVE_START_DATE_DN BEN_PBB_PERSON_ID_NN BEN_PBB_PER_BNFTS_BAL_ID_NN BEN_PBB_BNFTS_BAL_ID_NN BEN_PBB_EFFECTIVE_END_DATE_DN

User Entity	Required Contexts	Array Database Item	Database Item Name
BEN_PER_LE_HABITS_COV_F	PERSON_ID, BUSINESS_GROUP_ID, EFFECTIVE_DATE	No	BEN_PHB_COORD_MED_CVG_END_DT BEN_PHB_TOBACCO_TYPE_USAGE BEN_PHB_COORD_MED_EXT_ER BEN_PHB_COORD_MED_INSR_CRR_IDENT BEN_PHB_COORD_MED_INSR_CRR_NAM BEN_PHB_COORD_MED_PLN_NO BEN_PHB_COORD_MED_PL_NAME BEN_PHB_COORD_NO_CVG_FLAG BEN_PHB_CVRD_IN_ANTHR_PL BEN_PHB_DISABILITY_STATUS BEN_PHB_DPDNT_ADOPTION_DATE BEN_PHB_DPDNT_VLNTRY_SVCE_FLAG BEN_PHB_EFFECTIVE_END_DATE BEN_PHB_EFFECTIVE_START_DATE BEN_PHB_ON_MILITARY_SERVICE BEN_PHB_RECEIPT_OF_DEATH_CERT_DATE BEN_PHB_REGISTERED_DISABLED_FLAG BEN_PHB_STUDENT_STATUS BEN_PHB_COORD_MED_CVG_STRT_DT
BEN_PER_BNF_ORG_TN	PERSON_ID, BUSINESS_GROUP_ID, EFFECTIVE_DATE	Yes	BEN_PBO_BNF_ORGANIZATION_NAME_TN BEN_PBO_END_DATE_DN BEN_PBO_START_DATE_DN BEN_PBO_TRUSTEE_ORG_REG_CD_TN BEN_PBO_TRUSTEE_EXECUTOR_NAME_TN

User Entity	Required Contexts	Array Database Item	Database Item Name
			BEN_PBO_TRUSTEE_ORG_DESCRIPTION_TN BEN_PBO_TRUSTEE_ORG_NAME_TN BEN_PBO_BNF_TYP_CD_TN BEN_PBO_TRUSTEE_ADDL_DETAILS_TN
BEN_PER_BENEFIT_GROUP_F_TN	PERSON_ID, BUSINESS_GROUP_ID, EFFECTIVE_DATE	Yes	EN_PBG_BENEFIT_GROUP_NAME_TN
BEN_EXT_HIST_PLD	PERSON_ID, EFFECTIVE_DATE	Yes	BEN_EXT_HIST_PLD_ACTION_TIME_DN BEN_EXT_HIST_PLD_STATUS_TN BEN_EXT_HIST_PLD_BENEFIT_RELATION_ID_NN BEN_EXT_HIST_PLD_PRTT_LEG_DISCMR_ACTN_ID_NN BEN_EXT_HIST_PLD_PL_ID_NN BEN_EXT_HIST_PLD_PGM_ID_NN BEN_EXT_HIST_PLD_PER_IN_LER_ID_NN BEN_EXT_HIST_PLD_PERSON_ID_NN BEN_EXT_HIST_PLD_LER_ID_NN BEN_EXT_HIST_PLD_LEGAL_ENTITY_ID_NN BEN_EXT_HIST_PLD_LEGAL DISCLAIMER_ID_NN BEN_EXT_HIST_PLD_LAST_UPDATE_LOGIN_TN BEN_EXT_HIST_PLD_LAST_UPDATE_DATE_DN BEN_EXT_HIST_PLD_LAST_UPDATED_BY_TN BEN_EXT_HIST_PLD_CREATION_DATE_DN BEN_EXT_HIST_PLD_CREATED_BY_TN

User Entity	Required Contexts	Array Database Item	Database Item Name
			BEN_EXT_HIST_PLD_BRN_BENEFIT_REL_SYSTEM_CD_TN BEN_EXT_HIST_PLD_BRN_STATUS_TN BEN_EXT_HIST_PLD_BENEFIT_RELATION_NAME_TN BEN_EXT_HIST_PLD_LEG_DESCRIPTION_TN BEN_EXT_HIST_PLD_LEGAL_ENTITY_NAME_TN BEN_EXT_HIST_PLD_LER_SHORT_CODE_TN BEN_EXT_HIST_PLD_LER_TYP_CD_TN BEN_EXT_HIST_PLD_LER_GLOBAL_FLAG_TN BEN_EXT_HIST_PLD_LER_SLCTBL_SLF_SVC_CD_TN BEN_EXT_HIST_PLD_LER_SHORT_NAME_TN BEN_EXT_HIST_PLD_LER_LER_NAME_TN BEN_EXT_HIST_PLD_PERSON_NAME_TN BEN_EXT_HIST_PLD_PGM_NAME_TN BEN_EXT_HIST_PLD_PGM_SHORT_NAME_TN BEN_EXT_HIST_PLD_PGM_SHORT_CODE_TN BEN_EXT_HIST_PLD_PIL_LF_EVT_OCRD_DT_DN BEN_EXT_HIST_PLD_PIL_PER_IN_LER_STAT_CD_TN BEN_EXT_HIST_PLD_PL_NAME_TN BEN_EXT_HIST_PLD_PL_SHORT_NAME_TN BEN_EXT_HIST_PLD_PL_SHORT_CODE_TN
BEN_EXT_HIST_PTNL_LER_FOR_PER	PERSON_ID,	Yes	BEN_EXT_HIST_PTNL_BRN_BENEFIT_RELATION_NAME_TN

User Entity	Required Contexts	Array Database Item	Database Item Name
	EFFECTIVE_DATE		BEN_EXT_HIST_PTNL_BRN_BENEFIT_REL_SYSTEM_CD_TN BEN_EXT_HIST_PTNL_BRN_EFFECTIVE_END_DATE_DN BEN_EXT_HIST_PTNL_BRN_EFFECTIVE_START_DATE_DN BEN_EXT_HIST_PTNL_BRN_PRIMARY_REL_TN BEN_EXT_HIST_PTNL_BRN_REL_PRMRY_ASG_ID_NN BEN_EXT_HIST_PTNL_BRN_STATUS_TN BEN_EXT_HIST_PTNL_LEGAL_ENTITY_NAME_TN BEN_EXT_HIST_PTNL_LER_LF_EVT_OPER_CD_TN BEN_EXT_HIST_PTNL_LER_TMLNS_PERD_CD_TN BEN_EXT_HIST_PTNL_LER_TMLNS_EVAL_CD_TN BEN_EXT_HIST_PTNL_LER_TMLNS_DYS_NUM_NN BEN_EXT_HIST_PTNL_LER_SLCTBL_SLF_SVC_CD_TN BEN_EXT_HIST_PTNL_LER_SHORT_NAME_TN BEN_EXT_HIST_PTNL_LER_SHORT_CODE_TN BEN_EXT_HIST_PTNL_LER_PTNL_LER_TRMT_CD_TN BEN_EXT_HIST_PTNL_LER_OVRDGL_E_FLAG_TN BEN_EXT_HIST_PTNL_LER_OCRD_DT_DET_CD_TN BEN_EXT_HIST_PTNL_LER_NAME_TN BEN_EXT_HIST_PTNL_LER_GLOBAL_FLAG_TN BEN_EXT_HIST_PTNL_PERSON_NAME_TN BEN_EXT_HIST_PTNL_CREATED_BY_TN

User Entity	Required Contexts	Array Database Item	Database Item Name
			BEN_EXT_HIST_PTNL_LF_EVT_OCRD_DT_DN BEN_EXT_HIST_PTNL_MNL_DT_DN BEN_EXT_HIST_PTNL_NTFN_DT_DN BEN_EXT_HIST_PTNL_PERSON_ID_NN BEN_EXT_HIST_PTNL_PROCD_DT_DN BEN_EXT_HIST_PTNL_VOIDD_DT_DN BEN_EXT_HIST_PTNL_LER_ID_NN BEN_EXT_HIST_PTNL_PTNL_LER_FOR_PER_ID_NN BEN_EXT_HIST_PTNL_LER_FOR_PER_STAT_CD_TN BEN_EXT_HIST_PTNL_BENEFIT_RELATION_ID_NN BEN_EXT_HIST_PTNL_LEGAL_ENTITY_ID_NN BEN_EXT_HIST_PTNL_LAST_UPDATE_LOGIN_TN BEN_EXT_HIST_PTNL_LAST_UPDATE_DATE_DN BEN_EXT_HIST_PTNL_LAST_UPDATED_BY_TN BEN_EXT_HIST_PTNL_DTCTD_DT_DN BEN_EXT_HIST_PTNL_CREATION_DATE_DN BEN_EXT_HIST_PTNL_UNPROCD_DT_DN BEN_EXT_HIST_PTNL_LER_TYP_CD_TN
BEN_EXT_HIST_PCP	PERSON_ID, EFFECTIVE_DATE	Yes	BEN_EXT_HIST_PCP_DPNT_FULL_NAME_TN BEN_EXT_HIST_PCP_PRTTPCP_TN BEN_EXT_HIST_PCP_LEGAL_ENTITY_NAME_TN

User Entity	Required Contexts	Array Database Item	Database Item Name
			BEN_EXT_HIST_PCP_LER_SHORT_CODE_TN BEN_EXT_HIST_PCP_LER_ID_NN BEN_EXT_HIST_PCP_LER_NAME_TN BEN_EXT_HIST_PCP_LER_GLOBAL_FLAG_TN BEN_EXT_HIST_PCP_LER_SHORT_NAME_TN BEN_EXT_HIST_PCP_LER_TYP_CD_TN BEN_EXT_HIST_PCP_OPT_NAME_TN BEN_EXT_HIST_PCP_OPT_SHORT_NAME_TN BEN_EXT_HIST_PCP_OPT_SHORT_CODE_TN BEN_EXT_HIST_PCP_DPNT_RLNSHP_CD_TN BEN_EXT_HIST_PCP_DPNT_PERSON_ID_NN BEN_EXT_HIST_PCP_ENRT_MTHD_CD_TN BEN_EXT_HIST_PCP_ENRT_CVG_THRU_DT_DN BEN_EXT_HIST_PCP_ENRT_CVG_STRT_DT_DN BEN_EXT_HIST_PCP_ELECTION_DATE_DN BEN_EXT_HIST_PCP_COMP_LVL_CD_TN BEN_EXT_HIST_PCP_PL_TYP_ID_NN BEN_EXT_HIST_PCP_PL_ID_NN BEN_EXT_HIST_PCP_PGM_ID_NN BEN_EXT_HIST_PCP_ORGNL_ENRT_DT_DN BEN_EXT_HIST_PCP_SSPNDD_FLAG_TN BEN_EXT_HIST_PCP_PRTT_ENRT_RSLT_STAT_CD_TN

User Entity	Required Contexts	Array Database Item	Database Item Name
			BEN_EXT_HIST_PCP_OPT_ID_NN BEN_EXT_HIST_PCP_LEGAL_ENTITY_ID_NN BEN_EXT_HIST_PCP_PERSON_FULL_NAME_TN BEN_EXT_HIST_PCP_PGM_SHORT_NAME_TN BEN_EXT_HIST_PCP_PGM_NAME_TN BEN_EXT_HIST_PCP_PGM_SHORT_CODE_TN BEN_EXT_HIST_PCP_BENEFIT_RELATION_ID_NN BEN_EXT_HIST_PCP_LF_EVT_OCRD_DT_DN BEN_EXT_HIST_PCP_PER_IN_LER_ID_NN BEN_EXT_HIST_PCP_PER_IN_LER_STAT_CD_TN BEN_EXT_HIST_PCP_PERSON_ID_NN BEN_EXT_HIST_PCP_PL_SHORT_CODE_TN BEN_EXT_HIST_PCP_PL_NAME_TN BEN_EXT_HIST_PCP_PL_SHORT_NAME_TN BEN_EXT_HIST_PCP_PL_TYP_NAME_TN BEN_EXT_HIST_PCP_LAST_UPDATE_LOGIN_TN BEN_EXT_HIST_PCP_LAST_UPDATE_DATE_DN BEN_EXT_HIST_PCP_EXT_IDENT_TN BEN_EXT_HIST_PCP_ELIG_PER_ELCTBL_CHC_ID_NN BEN_EXT_HIST_PCP_ELIG_DPNT_ID_NN BEN_EXT_HIST_PCP_ELIG_CVRD_DPNT_ID_NN

User Entity	Required Contexts	Array Database Item	Database Item Name
			BEN_EXT_HIST_PCP_CVRD_FLAG_TN BEN_EXT_HIST_PCP_PRTT_ENRT_RSLT_ID_NN BEN_EXT_HIST_PCP_PRMRY_CARE_PRVDR_ID_NN BEN_EXT_HIST_PCP_PRMRY_CARE_PRVDR_TYP_CD_TN BEN_EXT_HIST_PCP_CREATED_BY_TN BEN_EXT_HIST_PCP_CREATION_DATE_DN BEN_EXT_HIST_PCP_NAME_TN BEN_EXT_HIST_PCP_LAST_UPDATED_BY_TN BEN_EXT_HIST_PCP_CVG_THRU_DT_DN BEN_EXT_HIST_PCP_CVG_STRT_DT_DN
BEN_EXT_HIST_PBB	PERSON_ID, EFFECTIVE_DATE	Yes	BEN_EXT_HIST_PBB_BNB_BNFTS_BAL_DESC_TN BEN_EXT_HIST_PBB_BNB_BNFTS_BAL_NAME_TN BEN_EXT_HIST_PBB_BNB_BNFTS_BAL_USG_CD_TN BEN_EXT_HIST_PBB_BNB_NNMNTRY_UOM_TN BEN_EXT_HIST_PBB_BRN_BRN_STATUS_TN BEN_EXT_HIST_PBB_BRN_BENEFIT_RELATION_NAME_TN BEN_EXT_HIST_PBB_BRN_BENEFIT_REL_SYSTEM_CD_TN BEN_EXT_HIST_PBB_BRN_BRN_PRIMARY_REL_TN BEN_EXT_HIST_PBB_LEGAL_ENTITY_NAME_TN BEN_EXT_HIST_PBB_PERSON_ID_NN BEN_EXT_HIST_PBB_LEGAL_ENTITY_ID_NN

User Entity	Required Contexts	Array Database Item	Database Item Name
			BEN_EXT_HIST_PBB_LAST_UPDATE_LOGIN_TN BEN_EXT_HIST_PBB_LAST_UPDATE_DATE_DN BEN_EXT_HIST_PBB_LAST_UPDATED_BY_TN BEN_EXT_HIST_PBB_CREATION_DATE_DN BEN_EXT_HIST_PBB_CREATED_BY_TN BEN_EXT_HIST_PBB_BNFTS_BAL_ID_NN BEN_EXT_HIST_PBB_VAL_NN BEN_EXT_HIST_PBB_ASSIGNMENT_ID_NN BEN_EXT_HIST_PBB_BENEFIT_RELATION_ID_NN BEN_EXT_HIST_PBB_PER_BNFTS_BAL_ID_NN BEN_EXT_HIST_PBB_UOM_TN BEN_EXT_HIST_PBB_PERSON_NAME_TN
BEN_EXT_HIST_FLEX_CREDT_LDGR	PERSON_ID, EFFECTIVE_DATE	Yes	BEN_EXT_HIST_FCL_CHILD_POOL_NAME_TN BEN_EXT_HIST_FCL_PRNT_POOL_NAME_TN BEN_EXT_HIST_FCL_POOL_LEGAL_ENTITY_NAME_TN BEN_EXT_HIST_FCL_POOL_LDG_NAME_TN BEN_EXT_HIST_FCL_ACTY_REF_PERD_CD_TN BEN_EXT_HIST_FCL_ANN_CASH_RECD_VAL_NN BEN_EXT_HIST_FCL_ANN_DONATED_TO_PRNT_VAL_NN BEN_EXT_HIST_FCL_ANN_FRFTD_VAL_NN BEN_EXT_HIST_FCL_ANN_PRVDD_VAL_NN

User Entity	Required Contexts	Array Database Item	Database Item Name
			BEN_EXT_HIST_FCL_ANN_REC_D_ FROM_CHLD_VAL_NN BEN_EXT_HIST_FCL_ANN_RLD_UP_VAL_NN BEN_EXT_HIST_FCL_ANN_USED_VAL_NN BEN_EXT_HIST_FCL_CASH_REC_D_VAL_NN BEN_EXT_HIST_FCL_CMCD_CASH_REC_D_VAL_NN BEN_EXT_HIST_FCL_CMCD_DONATED_TO_PRNT_VAL_NN BEN_EXT_HIST_FCL_CMCD_FRFTD_VAL_NN BEN_EXT_HIST_FCL_CMCD_PRVDD_VAL_NN BEN_EXT_HIST_FCL_CMCD_REC_D_FROM_CHLD_VAL_NN BEN_EXT_HIST_FCL_CMCD_REF_PERD_CD_TN BEN_EXT_HIST_FCL_CMCD_RLD_UP_VAL_NN BEN_EXT_HIST_FCL_CMCD_USED_VAL_NN BEN_EXT_HIST_FCL_CREATED_BY_TN BEN_EXT_HIST_FCL_CREATION_DATE_DN BEN_EXT_HIST_FCL_DONATED_TO_PRNT_POOL_VAL_NN BEN_EXT_HIST_FCL_FLEX_CREDT_LDGR_ID_NN BEN_EXT_HIST_FCL_FRFTD_VAL_NN BEN_EXT_HIST_FCL_LAST_UPDATED_BY_TN BEN_EXT_HIST_FCL_LAST_UPDATE_DATE_DN BEN_EXT_HIST_FCL_LAST_UPDATE_LOGIN_TN BEN_EXT_HIST_FCL_LDGR_CLASS_TN

User Entity	Required Contexts	Array Database Item	Database Item Name
			BEN_EXT_HIST_FCL_LDGR_END_DT_DN BEN_EXT_HIST_FCL_LDGR_STRT_DT_DN BEN_EXT_HIST_FCL_PRVDD_VAL_NN BEN_EXT_HIST_FCL_RECDFROM_CHLD_POOL_VAL_NN BEN_EXT_HIST_FCL_RLD_UP_VAL_NN BEN_EXT_HIST_FCL_USED_VAL_NN BEN_EXT_HIST_FCL_LER_SHORT_NAME_TN BEN_EXT_HIST_FCL_LER_ID_NN BEN_EXT_HIST_FCL_LER_GLOBAL_FLAG_TN BEN_EXT_HIST_FCL_LER_TYP_CD_TN BEN_EXT_HIST_FCL_LER_NAME_TN BEN_EXT_HIST_FCL_LER_SHORT_CODE_TN BEN_EXT_HIST_FCL_OPT_SHORT_NAME_TN BEN_EXT_HIST_FCL_OPT_NAME_TN BEN_EXT_HIST_FCL_OPT_SHORT_CODE_TN BEN_EXT_HIST_FCL_PRTT_ENRT_RS廖_ID_NN BEN_EXT_HIST_FCL_ENRT_MTHD_CD_TN BEN_EXT_HIST_FCL_ENRT_CVG_THRU_DT_DN BEN_EXT_HIST_FCL_ENRT_CVG_STRT_DT_DN BEN_EXT_HIST_FCL_ELECTION_DATE_DN BEN_EXT_HIST_FCL_PRTT_ENRT_RS廖_STAT_CD_TN

User Entity	Required Contexts	Array Database Item	Database Item Name
			BEN_EXT_HIST_FCL_COMP_LVL_CD_TN BEN_EXT_HIST_FCL_ORGNL_ENRT_DT_DN BEN_EXT_HIST_FCL_PERSON_FULL_NAME_TN BEN_EXT_HIST_FCL_PGM_RATE_LEVEL_TN BEN_EXT_HIST_FCL_PGM_SHORT_NAME_TN BEN_EXT_HIST_FCL_PGM_SHORT_CODE_TN BEN_EXT_HIST_FCL_PGM_NAME_TN BEN_EXT_HIST_FCL_PER_IN_LER_ID_NN BEN_EXT_HIST_FCL_PERSON_ID_NN BEN_EXT_HIST_FCL_BENEFIT_RELATION_ID_NN BEN_EXT_HIST_FCL_LF_EVT_OCRD_DT_DN BEN_EXT_HIST_FCL_PER_IN_LER_STAT_CD_TN BEN_EXT_HIST_FCL_PL_SHORT_CODE_TN BEN_EXT_HIST_FCL_PL_NAME_TN BEN_EXT_HIST_FCL_PL_SHORT_NAME_TN BEN_EXT_HIST_FCL_POOL_LEVEL_TN BEN_EXT_HIST_FCL_POOL_NAME_TN BEN_EXT_HIST_FCL_POOL_LDG_ID_NN BEN_EXT_HIST_FCL_POOL_LEGAL_ENTITY_ID_NN BEN_EXT_HIST_FCL_PROVIDER_OPTION_SHORT_NAME_TN BEN_EXT_HIST_FCL_PROVIDER_OPTION_NAME_TN

User Entity	Required Contexts	Array Database Item	Database Item Name
			BEN_EXT_HIST_FCL_PROVIDER_OPTION_SHORT_CODE_TN BEN_EXT_HIST_FCL_PROVIDER_PLAN_SHORT_NAME_TN BEN_EXT_HIST_FCL_PROVIDER_PLAN_NAME_TN BEN_EXT_HIST_FCL_PROVIDER_PLAN_SHORT_CODE_TN BEN_EXT_HIST_FCL_PRV_RATE_STRT_DT_DN BEN_EXT_HIST_FCL_PRV_MLT_CD_TN BEN_EXT_HIST_FCL_PRV_RATE_END_DT_DN BEN_EXT_HIST_FCL_PRV_RCRRG_FLAG_TN BEN_EXT_HIST_FCL_PRV_RATE_NAME_TN

# 9 Formulas for Performance Management

## How You Use Fast Formulas in Performance Documents

You can configure fast formulas to calculate section ratings for workers in a performance document.

### Fast Formulas and Database Items

Fast formulas contain the items and rules used to calculate the section rating, and can be based on ratings provided on the items by workers, managers, and participants. You create the fast formulas using the Performance Rating Calculation type. To the fast formula you add Database Items (DBI). These items can calculate ratings on attributes besides the overall item rating. For example, you can use DBIs to rate other attributes on a goal, such as the measurement attributes, rather than the overall goal. You create fast formulas using the Manage Fast Formulas task in the Setup and Maintenance work area.

For more information, see the Complete List of HRA Database Items for Calculated Ratings (Doc ID 2444793.1) on My Oracle Support at <http://support.oracle.com>.

### Calculated Rating Setup in the Performance Template Section and Performance Template

You configure the sections of the performance document to use fast formulas. Fast formulas can calculate ratings for the Overall Summary, Profile Content (Competencies), Performance Goals, and Development Goals sections. You can set up any, or all, these sections in a performance document to calculate ratings.

In the section, you must select the fast formula calculation rule. In addition, you select the fast formula rule. To use calculated ratings, you must also select the option to use calculated ratings on the Process tab of the performance template. You also must select which roles can see calculated ratings.



# 10 Formulas for Time and Labor and Workforce Management

## Where to find DBI and Fast Formula Info for Time and Labor and Workforce Management?

Go to the [\*How do I administer Time and Labor rules using fast formulas?\*](#) playbook.

