

# Using Oracle<sup>®</sup> FastFormula

**RELEASE 11*i***  
December 2001

**ORACLE<sup>®</sup>**

Using Oracle FastFormula Release 11i

The part number for this book is A73295-04.

**Copyright © 1996, 2000, 2001 Oracle Corporation. All rights reserved.**

Major Contributors: Louise Raffo, Julia Margetson, Rebecca Peters, John Woodward, Andrew Moran, Juliette Fleming, Michael O'Shea, Mark Rowe, Mark Swaine.

Contributors: Kevin Kelley, Michael Laverty, Keith Ekiss, Charles Hudson, Janet McCandless.

The Programs contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent and other intellectual and industrial property laws. Reverse engineering, disassembly or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

Program Documentation is licensed for use solely to support the deployment of the Programs and not for any other purpose.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the US Government or anyone licensing or using the Programs on behalf of the US government, the following notice is applicable:

#### **RESTRICTED RIGHTS LEGEND**

Programs delivered subject to the DOD FAR Supplement are 'commercial computer software' and use, duplication and disclosure of the Programs including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are 'restricted computer software' and use, duplication and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be licensee's responsibility to take all appropriate fail-safe, back-up redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark, and Oracle Financials, Oracle Discoverer, Oracle *Metalink*, PL/SQL, and SQL\*Plus, are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.



# Contents

<b>Preface</b>	<b>Preface</b> .....	<b>v</b>
	Audience for This Guide .....	vi
	How To Use This Guide .....	vi
	Do Not Use Database Tools to Modify Oracle Applications	
	Data .....	xiv
	About Oracle .....	xv
	Your Feedback .....	xv
<b>Chapter 1</b>	<b>Oracle FastFormula</b> .....	<b>1 – 1</b>
	Oracle FastFormula Overview .....	1 – 4
	Uses for Oracle FastFormula .....	1 – 5
	Introduction to Formula Components .....	1 – 8
	Formula Writing Techniques .....	1 – 15
	Writing or Editing a Formula .....	1 – 20
	Defining Global Values .....	1 – 22
	Registering a Function .....	1 – 23
	Writing Payroll Formulas for Elements .....	1 – 25
	Writing Formulas for Accrual Plans .....	1 – 26
	Writing Formulas To Calculate Absence Duration .....	1 – 32
	Writing Formulas for Element Skip Rules .....	1 – 33
	Copying and Adding Features to a QuickPaint Formula ...	1 – 34
	Writing Formulas for Validation .....	1 – 35
	Editing Assignment Set Formulas .....	1 – 36
	Bulk Compiling Formulas .....	1 – 37
	Editing a Quick Paint Formula .....	1 – 38

Checking an Element Entry .....	1 – 39
Checking a User Table Entry .....	1 – 40
Writing Formulas for Templates .....	1 – 41

## Chapter 2

<b>Oracle FastFormula Reference .....</b>	<b>2 – 1</b>
Formula Reference .....	2 – 2
Input Values in Payroll Formulas .....	2 – 3
Constants .....	2 – 5
Variables .....	2 – 7
Expressions .....	2 – 11
Arithmetic Operators .....	2 – 12
Functions .....	2 – 13
.....	2 – 20
Comments .....	2 – 33
Statements .....	2 – 34
Formula Compilation .....	2 – 40
Formula Errors .....	2 – 41
Database Items .....	2 – 43
Static Database Items .....	2 – 44
Dynamic Database Items .....	2 – 61
Formulas for Payroll Legislative Checks .....	2 – 67
Formulas for Benefits Administration .....	2 – 68
Total Compensation Formula Types .....	2 – 69
User Defined Functions .....	2 – 88
UK Only Functions .....	2 – 89
JP Functions .....	2 – 91
Sample Accrual Formula .....	2 – 93
Sample Proration Formulas (UK Only) .....	2 – 106
Sample Rates History Formulas (UK Only) .....	2 – 118

## Glossary

## Index

# Reader's Comment Form

## Using Oracle FastFormula, Release 11*i* A73295-04

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information we use for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual? What did you like least about it?

If you find any errors or have any other suggestions for improvement, please indicate the topic, chapter, and page number below:

---

---

---

---

---

---

---

---

---

---

Please send your comments to:

Oracle Applications Documentation Manager  
Oracle Corporation  
500 Oracle Parkway  
Redwood Shores, CA 94065  
Phone: (650) 506-7000 Fax: (650) 506-7200

If you would like a reply, please give your name, address, and telephone number below:

---

---

---

Thank you for helping us improve our documentation.





# Preface

---

## Audience for This Guide

Welcome to Release 11i of *Using Oracle FastFormula*

This guide assumes you have a working knowledge of the following:

- The principles and customary practices of your business area.
- Oracle FastFormula

If you have never used Oracle FastFormula, Oracle suggests you attend one or more of the Oracle FastFormula training classes available through Oracle University.

- The Oracle Applications graphical user interface.

To learn more about the Oracle Applications graphical user interface, read the *Oracle Applications User's Guide*.

See Other Information Sources for more information about Oracle Applications product information.

---

## How To Use This Guide

This guide contains the information you need to understand and use Oracle FastFormula. It provides details of syntax for each of the statements and functions available in FastFormula as well as examples of the different types of formulas you may want to write.

This preface explains how this user guide is organized and introduces other sources of information that can help you. This guide contains the following chapters:

- Chapter 1 contains an introduction to Oracle FastFormula and information about writing, editing, and using formulas to configure your system.
- Chapter 2 contains reference information for each command or function, sample formulas, and lists of all static and dynamic database items that are available to you in Oracle FastFormula. This list of items provides a simple method of accessing your HRMS data in FastFormula.

## Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that



make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle Corporation is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program web site at <http://www.oracle.com/accessibility/>.

### **Accessibility of Code Examples in Documentation**

JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

### **Accessibility of Links to External Web Sites in Documentation**

This documentation may contain links to web sites of other companies or organizations that Oracle Corporation does not own or control. Oracle Corporation neither evaluates nor makes any representations regarding the accessibility of these web sites.

## **Other Information Sources**

You can choose from many sources of information, including online documentation, training, and support services, to increase your knowledge and understanding of Oracle FastFormula.

If this guide refers you to other Oracle Applications documentation, use only the Release 11*i* versions of those guides.

## **Online Documentation**

All Oracle Applications documentation is available online (HTML and PDF).

- **Online Help** – The new features section in the HTML help describes new features in 11*i*. This information is updated for each new release of Oracle HRMS. The new features section also includes information about any features that were not yet available when this guide was printed. For example, if your administrator has installed software from a mini-pack as an upgrade, this document describes the new features. Online help patches are available on Metalink.

- **11i Features Matrix** – This document lists new features available by patch and identifies any associated new documentation. The new features matrix document is available on Metalink.
- **Readme File**– Refer to the readme file for patches that you have installed to learn about new documentation or documentation patches that you can download.

## Related User's Guides

Oracle FastFormula shares business and setup information with other Oracle Applications products. Therefore, you may want to refer to other user guides when you set up and use Oracle FastFormula.

You can read the guides online by choosing Library from the expandable menu on your HTML help window, by reading from the Oracle Applications Document Library CD included in your media pack, or by using a Web browser with a URL that your system administrator provides.

If you require printed guides, you can purchase them from the Oracle store at <http://oraclestore.oracle.com>.

## Guides Related to All Products

### Oracle Applications User's Guide

This guide explains how to enter data, query, run reports, and navigate using the graphical user interface (GUI) available with this release of Oracle HRMS (and any other Oracle Applications products). This guide also includes information on setting user profiles, as well as running and reviewing reports and concurrent processes.

You can access this user's guide online by choosing "Getting started with Oracle Applications" from any Oracle Applications help file.

### Oracle Alert User's Guide

This guide explains how to define periodic and event alerts to monitor the status of your Oracle Applications data.

### Oracle Applications Developer's Guide

This guide contains the coding standards followed by the Oracle Applications development staff. It describes the Oracle Application Object Library components needed to implement the Oracle

Applications user interface described in the *Oracle Applications User Interface Standards for Forms-Based Products*. It also provides information to help you build your custom Oracle Forms Developer 6i forms so that they integrate with Oracle Applications.

## **Oracle Applications User Interface Standards for Forms-Based Products**

This guide contains the user interface (UI) standards followed by the Oracle Applications development staff. It describes the UI for the Oracle Applications products and how to apply this UI to the design of an application built by using Oracle Forms.

### **Guides Related to This Product**

#### **Using Oracle HRMS – The Fundamentals**

Use this guide to learn about representing your enterprise on your application. This includes setting up your organization hierarchy, recording details about jobs and positions within your enterprise, defining a payroll, and also how to manage your costs.

#### **Managing Your Workforce Using Oracle HRMS**

Use this guide to learn about all aspects of managing your workforce. This includes how to represent your workforce on your application, recruiting new employees and developing their careers, and also defining and managing budgets.

#### **Running Your Payroll Using Oracle HRMS**

This user guide provides information about wage attachments, taxes and social insurance, the payroll run, and other processes.

#### **Managing Total Compensation Using Oracle HRMS**

Use this guide to learn about setting up the application to store and process information on your total compensation package, including salary administration, other earnings, deductions, benefits, absence management and PTO accrual plans. For US users, standard and advanced benefits is covered in a separate guide called *Managing Total Compensation Using Oracle HRMS*.

## **Configuring, Reporting and System Administration in Oracle HRMS**

This guide provides information about extending and configuring Oracle HRMS, managing security, auditing, information access, and letter generation.

## **Implementing Oracle HRMS**

This guide explains the setup procedures you need to carry out in order to successfully implement Oracle HRMS in your enterprise.

## **Implementing Oracle Self-Service Human Resources (SSHR)**

This guide provides information about setting up the self-service human resources management functions for managers and employees. Managers and employees can then use an intranet and Web browser to have easy and intuitive access to personal information and career management functionality.

## **Using Oracle Training Administration (OTA)**

This guide provides information about how to set up and use Oracle Training Administration to facilitate your training and certification business.

## **Using Oracle SSP/SMP**

This guide provides information about setting up and using Oracle SSP/SMP to meet your statutory sick pay and statutory maternity pay obligations.

## **Using Application Data Exchange**

This guide provides information about using Application Data Exchange to view HRMS data with desktop tools, and upload revised data to your application.

## **Oracle Business Intelligence System Implementation Guide**

This guide provides information about implementing Oracle Business Intelligence (BIS) in your environment.

## **BIS User Guide 11i**

---

This guide is provided as online help only from the BIS application and includes information about intelligence reports, Discoverer workbooks, and the Performance Management Framework.

## **Using Oracle Time Management**

---

This guide provides information about capturing work patterns such as shift hours so that this information can be used by other applications such as General Ledger.

# **Installation and System Administration**

## **Oracle Applications Concepts**

---

This guide provides an introduction to the concepts, features, technology stack, architecture, and terminology for Oracle Applications Release 11i. It provides a useful first book to read before an installation of Oracle Applications. This guide also introduces the concepts behind Applications-wide features such as Business Intelligence (BIS), languages and character sets, and Self-Service Web Applications.

## **Installing Oracle Applications**

---

This guide provides instructions for managing the installation of Oracle Applications products. In Release 11i, much of the installation process is handled using Oracle Rapid Install, which minimizes the time to install Oracle Applications and the Oracle8 technology stack, and the Oracle8i Server technology stack by automating many of the required steps. This guide contains instructions for using Oracle Rapid Install and lists the tasks you need to perform to finish your installation. You should use this guide in conjunction with individual product user's guides and implementation guides.

## **Upgrading Oracle Applications**

---

Refer to this guide if you are upgrading your Oracle Applications Release 10.7 or Release 11.0 products to Release 11i. This guide describes the upgrade process and lists database and product-specific upgrade tasks. You must be either at Release 10.7 (NCA, SmartClient, or character mode) or Release 11.0, to upgrade to Release 11i. You cannot upgrade to Release 11i directly from releases prior to 10.7.

## **Maintaining Oracle Applications**

---

Use this guide to help you run the various AD utilities, such as AutoUpgrade, AutoPatch, AD Administration, AD Controller, AD Relink, License Manager, and others. It contains how-to steps, screenshots, and other information that you need to run the AD utilities. This guide also provides information on maintaining the applications file system and database.

## **Oracle Applications System Administrator's Guide**

---

This guide provides planning and reference information for the Oracle Applications System Administrator. It contains information on how to define security, customize menus and online help, and manage concurrent processing.

## **Other Implementation Documentation**

### **Oracle Applications Product Update Notes**

---

Use this guide as a reference for upgrading an installation of Oracle Applications. It provides a history of the changes to individual Oracle Applications products between Release 11.0 and Release 11i. It includes new features, enhancements, and changes made to database objects, profile options, and seed data for this interval.

### **Multiple Reporting Currencies in Oracle Applications**

---

If you use the Multiple Reporting Currencies feature to record transactions in more than one currency, use this manual before implementing Oracle HRMS. This manual details additional steps and setup considerations for implementing Oracle HRMS with this feature.

### **Multiple Organizations in Oracle Applications**

---

If you use the Oracle Applications Multiple Organization Support feature to use multiple sets of books for one Oracle HRMS installation, this guide describes all you need to know about setting up and using Oracle HRMS with this feature.

### **Oracle Workflow Guide**

---

This guide explains how to define new workflow business processes as well as customize existing Oracle Applications–embedded workflow processes. You also use this guide to complete the setup steps necessary

for any Oracle Applications product that includes workflow-enabled processes.

### **Oracle Applications Flexfields Guide**

---

This guide provides flexfields planning, setup, and reference information for the Oracle FastFormula implementation team, as well as for users responsible for the ongoing maintenance of Oracle Applications product data. This manual also provides information on creating custom reports on flexfields data.

### **Oracle Technical Reference Manuals**

---

The technical reference guides are now available in electronic format only. You can now access technical reference manuals for any Oracle Applications product you have licensed.

### **Oracle Manufacturing and Distribution Open Interfaces Manual**

---

This manual contains up-to-date information about integrating with other Oracle Manufacturing applications and with your other systems. This documentation includes open interfaces found in Oracle Manufacturing.

### **Oracle Applications Message Reference Manual**

---

This manual describes all Oracle Applications messages. This manual is available in HTML format on the documentation CD-ROM for Release 11i.

## **Training and Support**

### **Training**

---

Oracle offers a complete set of training courses to help you and your staff master Oracle HRMS and reach full productivity quickly. These courses are organized into functional learning paths, so you take only those courses appropriate to your job or area of responsibility.

You have a choice of educational environments. You can attend courses offered by Oracle University at any one of our many Education Centers, you can arrange for our trainers to teach at your facility, or you can use Oracle Learning Network, Oracle University's online education utility. In addition, Oracle Training professionals can tailor standard courses or develop custom courses to meet your needs. For

example, you may want to use your organization structure, terminology, and data as examples in a customized training session delivered at your own facility.

## **Support**

---

From on-site support to central support, our team of experienced professionals provides the help and information you need to keep Oracle FastFormula working for you. This team includes your Technical Representative, Account Manager, and Oracle's large staff of consultants and support specialists with expertise in your business area, managing an Oracle8i server, and your hardware and software environment.

---

## **Do Not Use Database Tools to Modify Oracle Applications Data**

***Oracle STRONGLY RECOMMENDS that you never use SQL\*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle Applications data unless otherwise instructed.***

Oracle provides powerful tools you can use to create, store, change, retrieve, and maintain information in an Oracle database. But if you use Oracle tools such as SQL\*Plus to modify Oracle Applications data, you risk destroying the integrity of your data and you lose the ability to audit changes to your data.

Because Oracle Applications tables are interrelated, any change you make using an Oracle Applications form can update many tables at once. But when you modify Oracle Applications data using anything other than Oracle Applications, you may change a row in one table without making corresponding changes in related tables. If your tables get out of synchronization with each other, you risk retrieving erroneous information and you risk unpredictable results throughout Oracle Applications.

When you use Oracle Applications to modify your data, Oracle Applications automatically checks that your changes are valid. Oracle Applications also keeps track of who changes information. If you enter information into database tables using database tools, you may store invalid information. You also lose the ability to track who has changed your information because SQL\*Plus and other database tools do not keep a record of changes.



---

## About Oracle

Oracle Corporation develops and markets an integrated line of software products for database management, applications development, decision support and office automation, as well as Oracle Applications, an integrated suite of more than 160 software modules for financial management, supply chain management, manufacturing, project systems, human resources and customer relationship management.

Oracle products are available for mainframes, minicomputers, personal computers, network computers and personal digital assistants, allowing organizations to integrate different computers, different operating systems, different networks, and even different database management systems, into a single, unified computing and information resource.

Oracle is the world's leading supplier of software for information management, and the world's second largest software company. Oracle offers its database, tools, and application products, along with related consulting, education and support services, in over 145 countries around the world.

---

## Your Feedback

Thank you for using Oracle FastFormula and this user's guide.

Oracle values your comments and feedback. This guide contains a Reader's Comment Form you can use to explain what you like or dislike about Oracle FastFormula or this user's guide. Mail your comments to the following address or call us directly at (650) 506-7000.

Oracle Applications Documentation Manager  
Oracle Corporation  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

Or, send electronic mail to [appsdoc\\_us@oracle.com](mailto:appsdoc_us@oracle.com).



CHAPTER

# *1*

# Oracle FastFormula

---

# Oracle FastFormula

## What is Oracle FastFormula?

Oracle FastFormula 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.

## How can you use formulas?

You can use formulas to:

- Calculate element pay values and run results during payroll processing.
- Check that element entry values are valid for an assignment.
- Check that entries made to a user table are valid.
- Specify the criteria for including an assignment in an assignment set.
- Create rules for benefits administration, such as eligibility determination
- Select the database information you want to display in a QuickPaint report, and perform calculations for the report.
- Specify the rules for skipping an element during payroll processing.
- Prepare records in the format you require for the magnetic tape writer. For most countries, the Oracle localization team has written the required formulas.
- Perform legislative checks during a payroll run.
- Specify the rules for Paid Time Off accrual plans, such as how much time is accrued and over what period, when new starters are eligible to begin accruing time, and how much time can be carried over to the next accrual term.
- Calculate the duration of an absence, given the start and end dates and times.
- Configure the people management templates in a number of ways, including supplying additional information to be available from fields on the template, and validating field entries.
- Validate forms by calling formulas from the Custom Library. Calling formulas from the Custom Library also allows you to

easily create and maintain business rules. For more information about calling formulas from PL/SQL, refer to the technical essay *Calling FastFormula from PL/SQL*.

When you write a formula, you specify for which of these purposes you will use it.

### **Are any formulas supplied?**

Yes, formulas are predefined for all the tax calculations required for Oracle Payroll. You should not edit these formulas; you will automatically receive updates when tax rules change.

There are also a number of formulas predefined for accrual plans. You can use these formulas as supplied, or customize them to match the rules of your own plans.

When US and Canadian Payroll users define earnings and deductions, the system automatically generates the formulas required to process these elements. When necessary, you can edit these generated payroll formulas. In other legislations, some formulas may be predefined, and you can create as many other formulas as you require to process the elements you define.

Formulas for QuickPaint reports and assignment sets can be generated from criteria you enter in windows. You can edit these generated formulas to add more functionality.

---

## Oracle FastFormula Overview

Formulas are generic expressions of calculations or comparisons you want to repeat with different input values. They take input from a window, a database, or a process, such as a payroll run and they return values or messages. See *Uses for Oracle FastFormula*: page 1 – 5 for a description of the different ways to use formulas.

Oracle FastFormulas are made up of a number of different *components*. These can include assignment statements, different types of input and expressions.

When writing formulas there are standard *formula writing techniques* you should use. These ensure your formulas are easy to read, use and understand, not only by yourself but also by other people.

Finally, there are strict rules for components as to the words and punctuation you must use in a formula. These ensure Oracle FastFormula can correctly perform the calculations and validation you require. See *Formula Reference*: page 2 – 2 to find out the rules for a particular component.

---

## Uses for Oracle FastFormula

You can use Oracle FastFormula to:

- Calculate your payrolls
- Define the rules for PTO accrual plans
- Define custom calculations for benefits administration
- Define QuickPaint reports
- Validate element inputs or user tables
- Edit assignment sets
- Calculate absence duration
- Configure people management templates
- Set up business rules and call them from other PL/SQL applications
- Define your Oracle Business Intelligence Systems reports.

## Payroll Calculations

You can use predefined payroll formulas. When you receive Oracle Payroll, some formulas are predefined in your system. You cannot make any changes to these formulas, but you can copy them and modify the copies.

US Payroll users can use Oracle FastFormula to edit the formulas the system generates for the earnings types, other payments and deductions you initiate in Oracle Payroll. You make these edits directly to the generated formula (not to a copy) using the Formula window.

All Oracle Payroll users can use FastFormula to write payroll calculations and skip rules for elements you define yourself to represent earnings and deductions. You can associate more than one formula with each element, to perform different processing for employee assignments with different statuses. US Payroll users need only define their own elements and formulas for earnings and deductions with highly complex calculations requiring a number of different calls to the database.

## PTO Accrual Plans

You can use Oracle FastFormula to edit the seeded Accrual type formulas, or to write your own. Each accrual plan needs two formulas: one to calculate gross accrual, and the other to return information to

the PTO Carry Over process. You can optionally create a third formula if you want to use the Batch Element Entry (BEE) to make absence entries against accrual plans. This formula is called by BEE to check whether an employee is eligible to use accrued PTO. See: Accrual Formulas in *Managing Total Compensation Using Oracle HRMS*.

## **Benefits Administration**

You can use Oracle FastFormula to augment the system's benefits administration features. Use formulas to configure your plan design to the requirements of your enterprise. For example, you can write a formula to calculate benefits eligibility for those cases where eligibility determination is most complex.

## **QuickPaint Reports**

In the Define QuickPaint Report window, you can paste database items from a list into the Report Definition area and enter free text to label the data. When you save your QuickPaint Report definition, a formula is generated automatically. Formulas generated from QuickPaint do not include conditional logic or calculations. You may want to add these features, for example to sum up compensation elements or to report different data for different assignments.

## **Validation of Element Inputs or User Tables**

You can use Oracle FastFormula to validate user entries into element input values using lookups or maximum and minimum values. However, if you need more complex validation, you can write a formula to check the entry.

You can also use a formula to validate entries into user tables that you define.

## **Assignment Sets**

When you define assignment sets in the Assignment Set window, Oracle FastFormula generates a formula to define an assignment set from the criteria entered. However, you may want to change the sequence in which the set criteria are checked for each assignment.

## **Absence Duration**

You can write a formula to calculate the duration of an absence from the start and end dates and times. Your localization team may have



created a default formula, but you can override this with your own version.

## Configuring People Management Templates

There are several ways you can use formulas to configure the people management templates:

- A Template Validation formula can check values entered in a field.
- A Template Information formula can specify information to be displayed from the right mouse menu when a user right-clicks in a field.
- A QuickPaint formula can return a text string to display in the Assignment field on the Maintenance window and in the Data Organizer.
- A QuickPaint formula can return message tokens that you can use in a notification message issued from template forms.

## Calling FastFormula from PL/SQL

You can call formulas from PL/SQL applications. This enables direct access to data items and makes it possible to develop custom code for localized business rules. More detailed information about calling FastFormula from PL/SQL can be found in the technical essay *Calling FastFormula from PL/SQL*.

## Oracle Business Intelligence Systems (BIS) Reports

Using formulas you can configure your HRMS BIS reports so that they answer the business questions which are important to your enterprise. You can:

- Define how workforce should be counted within your enterprise for the Workforce reports.
- Set up information about how you want to record and report on regular and overtime hours for the Hours Worked Analysis report.
- Set up the standard hours for your enterprise for the Absence Analysis report.

---

## Introduction to Formula Components

The following information uses a series of examples to help you understand how to use different components to build your Oracle formulas.

To start with a simple example, suppose you wanted to calculate the pay value for the element *Wage* by multiplying the number of hours an employee works each week by hourly rate. You could write this formula:

```
wage = hours_worked * hourly_rate  
RETURN wage
```

**Note:** If you want to know the rules which govern the use of a specific component, refer to *Formula Reference*.

## Assignment and Return Statements

The first line is an *Assignment statement* that simply assigns a value to the element *Wage*. The second line is a *Return statement* that passes back the *Wage* value to the payroll run.

## Constants and Variables

In this example, the *Wage* value is calculated, but it could be a *constant* value, such as: `wage = 200`. To calculate the *Wage* value, Oracle FastFormula needs to get values for the *variables* `hours_worked` and `hourly_rate`. They are called variables because they can contain different values depending, in this example, on which assignment the payroll run is processing.

## Data Types

Both variables and constants can be one of three data types:

- numeric
- text
- date

The variables in the *Wage* example are numeric.

## Types of Input

We have said that Oracle FastFormula needs to get values for the variables `hours_worked` and `hourly_rate`. There are three ways it can do this:

- Receiving them as input when the formula is called.
- Finding the values in the database from *database items*.
- Using *global values*, which you enter in the Globals window.

To use a database item or global value in your formula, you simply refer to it by name. You can browse through lists of database items in the Formulas window. To use a value passed to the formula at run time, you must write an *Inputs statement*.

### Input Statements

In our Wage example, suppose that `hours_worked` is an input value to the element Wage. To pass the element input values to the formula during a payroll run, you define an Inputs statement, as follows:

```
INPUTS ARE hours_worked
wage = hours_worked * hourly_rate
RETURN wage
```

The name you use in the Inputs statement must be the same as the name of the element input value, and multiple words must be joined by underscores. In this example, the input value `hours_worked` is numeric. If the input value is not numeric, you must tell Oracle FastFormula whether it is text or date. For example:

```
INPUTS ARE start_date (date)
```

### Database Items

Suppose that `hourly_rate` is a standard rate taken from the Grade Rates table. This is an example of a database item. A database item has a label, or special piece of code, telling Oracle FastFormula the path to take to access the data. Oracle HRMS produces most of the database items you need without you taking any special action. These items include both information unique to your enterprise, which you hold in flexfield segments, and standard information such as assignment numbers and grades.

In the Formulas window, you pick database items from a list. You will see that the name of the database item for a grade rate called `hourly_rate` is actually `grade_hourly_rate_value`. This is the name you must use in your formula.

By this naming convention, Oracle FastFormula knows that `hourly_rate` is a database item from the Grade Rate table. But how does it know

which `hourly_rate` to use from this table? It works this out from the *context* the payroll run provides for each element entry. The context identifies:

- the Business Group
- the element and element link
- the payroll and payroll run
- the employee and employee assignment.



**Attention:** You should use an Inputs statement in preference to database items where possible because this is more efficient.

See: Writing Efficient Payroll Calculation Formulas: page 1 – 17.

### Global Variables

Use global values to store information that does not change often, but you refer to frequently, such as Company Name, or company-wide percentages used to calculate certain types of bonus. You define the global value and change its value using the Globals window.

See: Defining Global Values: page 1 – 22

### Local Variables

Local variables exist in one formula only. You can change the value of a local variable by assigning it a value in an Assignment statement. In the Wage example, the variable `wage` itself is a local variable. It receives a value within the formula by the Assignment statement:

```
wage = hours_worked * hourly_rate
```

**Note:** You cannot change the value of input values, database items, or global values within a formula.

## Functions

The Assignment statement in the wages example above uses a simple multiplication to calculate the value of the wages element. You can also use addition, subtraction, division, and a number of *functions*. For example:

```
bonus = GREATEST(days_at_work,163) + bonus_rate
```

Here the function `GREATEST` tells Oracle FastFormula to use the value of the variable `days_at_work`, if it is greater than 163, and otherwise to use the constant 163.

The data type of variables and constants determines how operators and functions act on the values. For example, the addition operator (+) can

be used with numeric or text data, while division can be used with numeric data only.

There are special functions that convert variables from:

- numbers to text (TO\_TEXT)
- dates to text (TO\_TEXT)
- text to date (TO\_DATE)
- text to number (TO\_NUM)

See: Functions: page 2 – 13

## Nested Expressions

The Assignment statement can use as many arithmetic operators and functions as you require. Each function or calculation is one *expression*, and you can nest expressions to create more complex calculations. You must use brackets to make clear to Oracle FastFormula the order in which the calculations are performed. For example:

```
ANNUAL_BONUS = trunc((((salary_amount/100)*
bonus_percentage)/183)*(days_between(end_period_date,
start_date) + 1)), 2)
```

Oracle FastFormula begins calculating inside the brackets and from left to right, in the following steps:

1. salary\_amount/100
2. 1. \* bonus\_percentage
3. 2. / 183
4. days\_between (end\_period\_date, start\_date)
5. 4. + 1
6. 3. \* 5.
7. TRUNC(6.,2)

Notice that TRUNC is another function. It rounds a numeric value down to the number of decimal places specified after the comma (two in this case).

## Incorporating Conditions

In our simple Wage element example, only one value is returned, and it is calculated in the same way for every assignment. However you may need to perform different calculations depending on the particular group of employee assignments, or the time of the year, or some other factors. You can do this by incorporating *conditions* in your formula.

## Simple Conditions

For example:

```
IF age < 20 THEN
    training_allowance = 30
ELSE
    training_allowance = 0
```

The formula checks whether the condition (age < 20) is true or false. If it is true, the formula processes the statement that follows the word THEN. If the condition is not true, the formula ignores this statement and processes any statement that follows the word ELSE. The ELSE clause is optional.

## Complex Conditions

In the example above, the condition compares a variable (age) to a constant (20). The condition can be more complex, comparing expressions that contain functions and arithmetic operators. For example:

```
IF (DAYS_BETWEEN(end_period_date, start_date)+1) >= threshold
```

DAYS\_BETWEEN is another function.

We have seen two *comparators*: less than (<) and greater than or equal to (>=). A full list of the comparators you can use appears in the Reference section.

See: Formula Reference: page 2 – 2

## WAS DEFAULTED

There is a special type of condition called WAS DEFAULTED. Use this to test whether a default value has been placed in an input value or database item. Default values are placed using the Default statement. For example:

```
DEFAULT FOR hourly_rate IS 3.00
X = hours_worked * hourly_rate
IF hourly_rate WAS DEFAULTED
    THEN
        MSG = 'Warning: hourly rate defaulted'
```

In this example, if the database item hourly\_rate is empty (NULL), the formula uses the default value of 3.00 and issues a warning message.

## Combined Conditions

You can combine conditions using the *logical operators* AND, OR, NOT.

- Use AND if you want an action to occur when more than one condition is true. For example:

```
IF (days_between(end_period_date, start_date) + 1) >= 183
AND employee_status = 'FULL TIME'
THEN . . .
```

- Use OR if you want an action to occur when any one of two or more conditions is true. For example:

```
IF stock_level < 10000
OR order_size >= 1500
THEN . . .
```

- Use NOT if you want an action to occur when a condition is *not* true. For example:

```
IF NOT (months_between(purchase_date, system_date) => 60
THEN . . .
```

As with Assignment statements, you may need to use brackets to tell Oracle FastFormula in which order to test conditions. By default, NOT has the highest precedence, followed by AND then OR. So the following condition:

```
IF X = 6 AND NOT Y = 7 OR P >= 6
```

is interpreted as:

```
IF ((X = 6) AND (NOT (Y = 7))) OR (P >= 6)
```

How you use brackets can change dramatically the meaning of a formula.



**Suggestion:** Use brackets whenever you create multiple conditions, so that the meaning of the formula is clear to other readers.

## Multiple Actions Based On Conditions

We have seen how to make conditions more complex. You can also make the actions performed as complex as you like. In our simple example above, the result of the condition was the assignment of a value to the variable `training_allowance`. As well as assigning values, you can perform calculations and return values.

For example, suppose you must check whether there are sufficient funds in a bank account before processing a withdrawal:

```
INPUTS ARE acct_balance,
           acct (text),
           debit_amt
IF  acct_balance >= debit_amt
THEN
  (
    new_balance = acct_balance - debit_amt
    RETURN new_balance
  )
ELSE
  (
    message = 'Account No. ' + acct + ' - Insufficient Funds'
    message2 = 'Account Balance is ' + TO_TEXT(acct_balance)
    RETURN message, message2
  )
)
```

Notice that you can return two variables in the same Return statement.



**Attention:** The brackets following THEN and ELSE are essential when you have multiple actions based on a condition. Without them, Oracle FastFormula processes the first statement conditionally and the other statements unconditionally.



---

## Formula Writing Techniques

When writing formulas there are a number of techniques you should follow to ensure your formulas are easy to read, use and understand.

### Commenting Formula

It is good practice to include comments in your formulas to explain to other people what the formula does.

So, for example, you can name your formula:

```
/* Formula: Attendance Bonus */
```

and write an explanation of your formula:

```
/* Use this formula to calculate the annual bonus for
clerical staff. Employees receive either a percentage of
their salary (if they have been present for 183 or more
days in the last six months), or a pro rata bonus (if they
have been in attendance for less than 183 days in the last
six months). */
```

Oracle FastFormula ignores everything between the comment delimiters: `/*` and `*/`. You can place comments anywhere in a formula without affecting the formula's performance.



**Warning:** Do not put a comment within a comment. This causes Oracle FastFormula to return a syntax error.

You can use a comment to explain what part of your formula does. So, for example, you might want a comment explaining who decides the bonus percentage:

```
INPUTS ARE salary_amount,
           start_date (date),
           end_period_date (date),
           bonus_percentage /* decided at board level */
```

You can also use comments to 'comment out' parts of the formula you do not currently want to use. So, for example, you can include a fifth input of `employee_status`, ensuring that employees with a status of full time are awarded a bonus. However, as you do not yet have a range of statuses, you do not currently need the fifth input.

```
INPUTS ARE salary_amount,
           start_date (date),
           end_period_date (date),
           bonus_percentage /* decided at board level */
           /* employee_status (text) */
```

Use comments and white space freely when entering formulas. This makes the formulas easier to read and understand, and has no effect on

performance or memory usage. Use indentation for the same reason, especially when you are using brackets to control the order of processing.

It is good practice to include the following information in a comment at the beginning of a formula:

- Formula title and short statement of its purpose
- Description of formula inputs
- List of variables and constants that may require updating
- Description of the input values of the element that receives the formula's direct result
- Explanation of the formula's calculations
- Administrative information such as the name, address and telephone number of an office administering the earnings, deduction, or charge the formula affects
- The dates of formula modifications, the names of those entering the edits, and possibly the reasons for change

## Alias Statements

Database items are named by the system when it creates them, and sometimes these names are too long to conveniently use in a formula. You cannot shorten the name of a database item (or a global value) itself, but you can set up an alternative shorter name to use within the formula. For example:

```
ALIAS as_overtime_qualifying_length_of_service AS ot_qls
```

In the rest of the formula, you can use the alias (in this example, `ot_qls`) as if it were the actual variable.



**Attention:** Using an Alias is more efficient than assigning the database item to a local variable with a short name.

## Default Statements

Use the Default statement to set a default value for an input value or a database item. The formula uses the default value if the database item is empty or no input value is provided when you run the formula. For example:

```
DEFAULT FOR hourly_rate IS 3.00  
X = hours_worked * hourly_rate  
IF hourly_rate WAS DEFAULTED
```

```
THEN  
  MSG = 'Warning: hourly rate defaulted'
```

This example sets a default of 3.00 for the database item `hourly_rate`. If `hourly_rate` is empty (NULL) in the database, the formula uses the default value of 3.00. The formula uses the 'WAS DEFAULTED' test to detect when a default value is used, in which case it issues a warning message.



**Attention:** You must use the Default statement for database items that can be empty. The Database Items window includes a check box labelled Default Required. This check box is checked for database items that can be empty. The Database Items window appears when you choose the Show Items button on the Formulas window.

## Writing Efficient Payroll Calculation Formulas

The following guidelines are generally true for typical payroll runs:

- The longer an element's formula, the longer its processing time.
- The more elements entered for an assignment, the longer its processing time.
- One element associated with a lengthy formula usually processes faster than two related elements each associated with a short formula.
- The overall number of elements and formulas in the system has little effect on processing efficiency. It is the number of elements per assignment that affects processing time.

### Variable Names and Aliases

To improve readability use names that are brief yet meaningful. Name length has no effect on performance or memory usage. Use Aliases if the names of database items or global values are long.

### Input Statements

Use Input statements rather than database items whenever possible. This improves formula processing by as much as a factor of ten. It speeds up the running of your payroll by eliminating the need to access the database for the input values.

***Inefficient:***

```
Salary = Salary_annual_salary / 12
RETURN Salary
```

***Efficient:***

```
INPUTS ARE Annual_salary
Salary = Annual_salary / 12
RETURN Salary
```

**Date Literals**

Use the TO\_DATE function only when the operand is a variable.

***Inefficient:***

```
Start_date = TO_DATE ( '1992-01-12 00:00:00' )
```

***Efficient:***

```
Start_date = '1992-01-12 00:00:00' (date)
```

**Single Expressions**

Use a single expression in straightforward formulas where this does not lead to confusion.

***Inefficient:***

```
Temp = Salary / Annualizing_factor
Tax = Temp * 3
```

***Efficient:***

```
Tax = (Salary / Annualizing_factor) * 3
```

**Database Items**

Do not refer to database items until you need them. People sometimes list at the top of a formula all the database items the formula might need, thinking this helps Oracle FastFormula process more quickly. However, this in fact slows processing by causing unnecessary database calls.

***Inefficient:***

```
S = Salary
A = Age
IF S < 20000 THEN
  IF A < 20 THEN
    Training_allowance = 30
  ELSE
    Training_allowance = 0
```

*Efficient:*

```
IF Salary < 20000 THEN
  IF Age < 20 THEN
    Training_allowance = 30
  ELSE
    Training_allowance = 0
```

The first example always causes a database fetch for Age whereas the second only fetches Age if Salary is less than 20000.

**Balance Dimensions for UK HRMS**

Wherever possible, use balance dimensions for single assignments only in formulas. Multiple assignments require more calculation, leading to slower processing time. The number of genuine multiple assignments in a payroll is not normally high, and the presence of a small number does not lead to any significant increase in overall processing time. There could be a problem, however, if you unnecessarily link balance dimensions for multiple assignments into general formulas.

**Proration Formulas for UK HRMS**

You set up proration formulas to enable element values to be calculated accurately if they change during a payroll period, for example, if an employee leaves the company or if their pay rate changes.

For more detailed information on proration, see the Technical Essay entitled Proration available on MetaLink (Technical Libraries/Human Resources Management Systems/Payroll/Documentation/United Kingdom).

---

## Writing or Editing a Formula

Use the Formula window to write and edit formulas.

**Note:** You cannot create formulas that exceed 64K in the Formula window. You must split longer formulas into two.

► **To write or edit a formula:**

1. Set your effective date to the date when you want to begin using the formula.
2. To write a new formula, enter the formula's name and select a type. To edit an existing formula, query its name.
3. Choose the Edit button to open a blank Edit Formula window where you can write a new formula, or modify an existing one.
4. If you want to select database items, choose the Show Items button to display the Database Items window and run a query. Copy and paste items from this window to the Edit Formula window.
5. When you finish writing or editing the formula, choose the Verify button to compile it.

This process identifies any syntax errors in your formula.

**Note:** Once you have compiled any formula, new functions or changes to existing functions made after the first time you compile, are not used. You must logout of Oracle HRMS and login again. You can now compile your formula and the new functions and/or changes to existing functions will be included.

6. When the formula is verified successfully, save it.
7. Your next step depends on the type of formula:
  - If the formula is of type Oracle Payroll, you must associate it with an element in the Formula Result Rules window.
  - If the formula is of type Element Skip, you select it in the Skip Rule field of the Element window.
  - If the formula is of type Element Input Validation, you select it in the Formula field of the Input Values window when you are defining an element.
  - If the formula is of type User Table Validation, you select it in the Formula field of the Columns window when you are defining a user table structure.
  - If the formula is of type Accrual, Accrual Carryover, or Accrual Ineligibility, you select it in the Accrual Plan window.

- If the formula is of type Accrual Subformula, you call it from another formula of type Accrual.
- If the formula is used for benefits administration, you select the formula in the Rules field of the appropriate benefits window.

Test your formula in the situation you intend to use it (such as, in a test payroll run) to ensure it contains no logical errors.

---

## Defining Global Values

Use global values to store information that does not change often, but you refer to frequently, as for example Company Name, or company-wide percentages used in the calculation of certain bonuses. You can use global values as variables in formulas by simply referring to the global value by name.

You can never change a global value using a formula. You change global values in the Globals window. Global values are date-tracked so you can make date effective changes ahead of time.

Global values are available to all formulas within a Business Group.

► **To define a global value:**

1. Set your effective date to the date when you want to begin using the global value.
2. Enter a name, data type (number, text, or date), and value. You can also enter a description.



---

## Registering a Function

You register a new function by naming and defining it, then creating contexts and parameters for it. Contexts are environment values that do not get passed to the function.

Where a function requires a mixture of contexts (from the FF\_CONTEXTS table) and parameters, the contexts should be listed first in the function header followed by the function parameters. Only the function parameters, however, need to be used to call the function from FastFormula.

For example, a function requires eight values: three contexts and five parameters. All eight values are listed in the function header but only the five parameters are used to call the function.

There are two classes of functions you can register:

- **External function:** These are further PL/SQL functions in addition to the ones already delivered with FastFormula. External functions can use contexts and parameters.
- **User Defined function:** It is suggested you do not use this class as it will be removed in future releases. Define new functions as External function class.

► **To register a new function for Oracle FastFormula:**

1. Enter a unique name for the new function.
2. Select date, number or text as its data type.
3. Select external or user defined as the class of the function.
4. Enter an alias for the function name if you require an alternative name for it. You can also enter a description to explain what the function is for. The Alias and Description fields are both optional.
5. Enter the definition of the function. Use the format: <package name>.<function name>.
6. Save your entries.

► **To enter context usage and parameter information:**

1. Choose the Context Usages button.
2. In the Context Usages window, select as many context items as you require for the function. The data type for each context displays automatically.

**Note:** The functionality that calls FastFormula, that is, QuickPaint or Payroll Processing, determines what contexts FastFormula has access to from the FF\_CONTEXTS table.

3. Save your entries. The sequence number of each context is entered automatically when you do this.
4. Close the Context Usages window and choose the Parameters button.
5. In the Parameters window, select the parameters, or operands, you require to define the function. Type and class display automatically.
6. Check the Optional check box if you want the corresponding parameter to be optional.
7. Check the Continuing check box if you want the function to make more than one call to the parameter.

**Note:** You cannot define a parameter as continuing unless you also make it optional. However, an optional parameter does not have to be continuing.

8. Save your entries.

---

## Writing Payroll Formulas for Elements

If you have defined your own payroll elements, you can write formulas to calculate earnings and deductions.

For guidance on writing efficient payroll calculation formulas, see: Formula Writing Techniques: page 1 – 15. For important information about using element input values in payroll formulas, see: Input Values in Payroll Formulas: page 2 – 3.

► **To define elements and their formulas:**

1. Design your element and how it will be calculated.
2. Write any formulas required to validate input values (formula type = Element Input Validation).
3. Write a formula, if required, to define the rules for skipping the element during payroll processing (formula type = Element Skip).
4. Define the element, referencing any formulas written in steps 2 and 3.
5. Write the formula or formulas for calculating the run results (formula type = Oracle Payroll).
6. Associate each Oracle Payroll type formula with the element in the Formula Result Rules window, and specify what happens to the formula results for this element.

**Note:** You can associate several formulas with a single element, each one for use with a different employee assignment status. You can also use the same formula for more than one element. In this case, if the formula references pay or input values (through the Inputs statement), each element must have pay and input values with the same names.

## Writing Formulas for Accrual Plans

Each accrual plan needs to be associated with two formulas: an accrual formula to calculate gross PTO entitlement to date and a Carry Over formula to be called by the carry over process at the end of the accrual term.

You can also associate a third formula to be called by BEE (Batch Element Entry) validation for entries to the absence element associated with the accrual plan. This Ineligibility formula checks whether an assignment is eligible to use accrued PTO. It must calculate the end of the ineligibility period in the same way as the Accrual formula for the plan. This formula is not required if you enter the ineligibility period for a plan in the Accrual Plan window.

Some formulas are seeded, see *Seeded Accrual Type Formulas in Managing Total Compensation Using Oracle HRMS*. You can use these as supplied, edit them, or write your own formulas to provide the plan rules you require.

This topic explains:

- The formula types for formulas associated with accrual plans
- The required inputs and outputs for Accrual, Carry Over, and Ineligibility formulas
- Checks you must include in your Accrual formulas to avoid errors

For a sample Accrual formula and suggestions on how to change it to incorporate a whole range of plan rules, see: *Sample Accrual Formula: page 2 – 93*.

## Formula Types

There are a number of formula types for formulas associated with accrual plans. You must define your formulas as the appropriate types or they will not be available for selection in the Accrual Plan window.

Formula Type	Displays on list of values for . . .	Use for . . .
Accrual	Accrual Formula field	The top level formula that calculates PTO entitlement for a plan
Accrual Subformula	--	Any formulas called by the top level formula, such as formulas for calculating the entitlement per period.

Formula Type	Displays on list of values for . . .	Use for . . .
Accrual Carryover	Carryover Formula field	The formula to be called by the Carry Over process.
Accrual Ineligibility	Ineligibility Formula field	The formula to be called by BEE (if required) to specify whether an assignment is eligible to use accrued PTO.

## Required Inputs and Outputs

If you write your own formulas for accrual plans, you **must** use the following INPUTS and RETURN statements. **Do not add extra lines to these statements.** You can use functions or database items to get extra inputs. The following values are available as contexts for all the accrual formula types (and therefore you do not need to retrieve them as inputs or database items):

- ASSIGNMENT\_ID
- DATE\_EARNED
- ACCRUAL\_PLAN\_ID
- BUSINESS\_GROUP\_ID
- PAYROLL\_ID

Some formula functions have been defined specially for Accrual type formulas. They require some or all of these contexts and they return values need for the accrual calculation, such as total absences, accrual band, and period dates. See: Functions for Accrual Type Formulas: page 2 – 26. You can define and register any other functions you require.

### Inputs and Outputs for Accrual Formulas

INPUTS ARE

```
Calculation_Date (date)
ACCRUAL_START_DATE (date)
ACCRUAL_LATEST_BALANCE
```

```
/* Formula body */
```

```
RETURN total_accrued_pto, effective_start_date,
effective_end_date, accrual_end_date
```

Input	Description
Calculation_Date	The date up to which accrual will be calculated
Accrual_Start_Date	The date to begin calculating accrual. If null, accruals are calculated from beginning of the accrual term.
Accrual_Latest_Balance	The latest balance for the accrual term up to the day before Accrual_Start_Date. The latest balance is held in a payroll balance.

**Note:** The Accrual\_Start\_Date and Accrual\_Latest\_Balance inputs are required only if your accrual plan uses a payroll balance to store gross accruals.

Output	Description
total_accrued_pto	Gross accrued PTO this term
effective_start_date	Start date of accrual, which is normally the start of this accrual term, but may be plan enrollment date, hire date, adjusted service date, or other, depending on plan rules.
effective_end_date	Normally the calculation date, but should be the termination date if the employee has been terminated, or the end date of the plan element entry if the employee has left the plan.
accrual_end_date	This is an optional output. In the seeded formulas it is the end of the last full accrual period before the calculation date (because these formulas do not take account of partial accrual periods).

### Inputs and Outputs for Carry Over Formula

```

INPUTS ARE
calculation_date (date),
accrual_term (text)

/* formula body */

RETURN max_carryover, effective_date, expiry_date

```

Input	Description
calculation_date	Any date falling within an accrual term
accrual_term	'PREVIOUS' or 'CURRENT' indicating whether to return the last date of the accrual term spanning calculation_date, or the accrual term previous to that spanning calculation date.

Output	Description
max_carryover	Maximum amount the employee can carry over, which may be dependent on an accrual band.
effective_date	The last date of an accrual term (either current or previous, as determined by the input). For example, this would be 31-DEC-YYYY for an accrual plan based on calendar years.
expiry_date	The date by which employees must use carried over PTO, otherwise they lose it. This output is optional.

### Inputs and Outputs for Ineligibility Formula

```

INPUTS ARE
calculation_date (date),

/* formula body */

RETURN assignment_eligible

```

Input	Description
calculation_date	The effective date of the element entry.

Output	Description
assignment_eligible	'Y' or 'N'. If Y, BEE creates the entry to the absence element (assuming all other validation is successful). If N, BEE creates a warning on the batch line for the absence entry.

## Checks You Must Include In Your Accrual Formulas

You may notice that the seeded formulas contain statements to check a number of dates to see whether an employee is eligible to accrue any PTO. Be sure to include these checks in your formulas too.

### Termination Date

Check whether there is a termination date for the assignment. If the termination date is before the calculation date, calculate accrual as of the termination date. If your formula does not handle partial accrual periods, check whether the termination date is before the end of the first accrual period; if yes, set gross accrual to zero.

### Enrollment End Date

Check whether there is an end date for the assignment's enrollment in the plan. If the end date is before the calculation date, calculate accrual as of the end date. If your formula does not handle partial accrual periods, check whether the end date is before the end of the first accrual period; if yes, set gross accrual to zero.

### Calculation Date

If the calculation date is before the end of the first accrual period, set gross accrual to zero (unless your formula handles partial accrual periods).

### Hire Date

Check the employee's hire date or continuous service date. If your formula handles partial accrual periods, check that this date is before the calculation date, and if not, set the gross accrual to zero. If your formula does not handle partial periods, check that this date is before the start of the last full accrual period used in the current calculation. If the employee has not worked for a full accrual period before the calculation date, set the gross accrual to zero.

### Start Date for New Plan Participants

Check when the employee should start to accrue time. This is typically the date of enrollment in the plan or (if your formula does not handle



partial accrual periods) the first period starting on or after the date of enrollment in the plan. If this date (or period) is after the calculation date (or period), set the accrual to zero.

**Note:** The seeded and sample formulas also show how to incorporate other start dates in your plan, such as six months after hire date, or start of calendar year after hire date.

### **Ineligibility Period**

Check any ineligibility period (which is a time when a new participant accrues time but cannot use it, so it does not appear credited to him or her until the end of the period). If the eligibility period is still in force at the calculation date (or, if your formula does not handle partial accrual periods, on the end date of the last accrual period used in the calculation) set the gross accrual to zero.

### **Inactive Assignment**

Check whether the employee's assignment has been active throughout the period for which you are calculating accruals. Depending on your plan rules, your employees might not accrue time when their assignments are inactive, or they might accrue time at a reduced rate. You can use the function `GET_ASG_INACTIVE_DAYS` to check the assignment status on each day from period start date to period end date and return the number of inactive working days.

---

## Writing Formulas To Calculate Absence Duration

You can write a formula to calculate absence duration automatically when a user enters an absence start and end date, or time. Your localization team may have written a formula, which the system will use by default. However, if you need a configured formula to take account of special work hours or shift patterns, you can create a new formula that will override the supplied one. You can create one formula for each Business Group.

Your formula must be called: `BG_ABSENCE_DURATION`. You must select the formula type QuickPaint.

The formula inputs must be:

- `days_or_hours` (units for the duration: D or H)
- `time_start`, `time_end`
- `date_start`, `date_end`

The formula outputs must be:

- `duration` (the calculated value or FAILED)
- `invalid_msg` (optional – an error message name)

Use the supplied example formula (`TEMPLATE_ABSENCE_DURATION`) as the basis for your formulas.

---

## Writing Formulas for Element Skip Rules

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 could specify:

- process the Union Fees element 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. Notice that 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.

► **To write a formula defining a skip rule:**

1. Select formula type Element Skip in the Formulas window.
2. Use as many input values as you require. The formula must set and return a local variable of type text, and this variable must be called `skip_flag`.

If the returned value of this variable begins with the letter y (such as 'Yes'), all processing for the element is skipped. Otherwise the element processes as normal.

The following example of a skip rule formula defines that the Union Fees element is not processed if the Union\_Fees\_Paid balance is greater than 10 000:

```
IF Union_Fees_Paid > 10000
  THEN
    skip_flag = 'yes'
  ELSE
    skip_flag = 'no'
RETURN skip_flag
```

---

## Copying and Adding Features to a QuickPaint Formula

When you save a QuickPaint Report definition, a formula is generated automatically. Formulas generated from QuickPaint do not include conditional logic or calculations. You may want to add these features, for example to sum up compensation elements or to report different data for different assignments.



**Attention:** If you want to add features to a generated QuickPaint formula, you must copy the formula and edit the copy. If you edit the original, your edits will be overwritten if the formula is regenerated from the QuickPaint definition.

► **To make a copy of a QuickPaint formula:**

1. In the Formula window, query your QuickPaint formula. It has the same name as your QuickPaint report.
2. Choose the Edit button. Select and copy the formula in the Edit Formula window.
3. Choose New Record from the Edit menu.
4. Enter a name for your edited copy and select the type QuickPaint.
5. Paste the text of the QuickPaint formula into the Edit Formula window.
6. Save your work.

---

## Writing Formulas for Validation

You can use Oracle FastFormula to validate user entries into the element input values, and to user tables that you define.

► **To write a formula for validation purposes:**

1. Write and validate the formula.

You must do this before you define the element or table, so that you can select the formula from a list in the Element window or Columns window.

2. Define the element or table.
3. Select formula type Element Input Validation or User Table Validation in the Formulas window.

### Rules to Observe

- There must be one input value, of type text, and it must be called `entry_value`.
- The formula must set and return a local variable giving the status of the validation (success or error). This variable must be called `formula_status` and have the value 's' (success) or 'e' (error).
- Optionally, the formula can also return a text variable giving an explanatory message. The returned message variable must be called `formula_message` and can contain any text. It can be returned with both successful and unsuccessful statuses.
- The formula must not return any other results.

For an element input value validation formula, you must also observe the following rules:

- You cannot use the element's other pay and input values in the formula.
- You cannot return a value to another pay or input value.

All entry values are stored in the database as text items. Therefore, if you want to validate an entry value as a date or number, you must use Oracle FastFormula's conversion function to convert the text into a date or number type variable. For example:

```
TO_NUM (entry_value)
TO_DATE(entry_value, 'DD-MON-YYYY')
```

---

## Editing Assignment Set Formulas

Assignment set formulas do not normally need to be edited. If, however, you enter multiple criteria to define an assignment set, with conditions joined by AND or OR, you may want to edit the formula to change the brackets in the generated conditions. This changes the order in which the conditions are tested.

To view an assignment set formula, query it in the Formula window. The formula type is Assignment Set and the formula name is the same as the assignment set name. To edit an assignment set formula, make a copy, as for a QuickPaint formula.

---

## Bulk Compiling Formulas

Where a formula has more than one version, you can compile all the versions in one process using the Bulk Compile Formulas process. For example, you run this process when you upgrade your legislative information, which contains formulas.

► **To bulk compile formulas**

1. Select Single Request in the Submit a New Request window.
2. In the Name field, select Bulk Compile Formulas.

You can define when you want to run this process using the schedule options.

**Note:** If you make any changes to a function after you have compiled a formula that uses it, you need to recompile the formula for the changes to take effect.

---

## Editing a Quick Paint Formula

If you want to add features to a generated QuickPaint formula, you must copy the formula and edit the copy. If you edit the original, your edits will be overwritten if the formula is regenerated from the QuickPaint definition.

In the following example, an automatically generated QuickPaint formula has been edited to add Line 09, which totals the input values used in the report.

```
LINE01=''
LINE02=' Pay Items      Value this Period
LINE03=''
LINE04=' Salary Value : ' + TO_TEXT(trunc((SALARY_ANNUAL/12),2))
LINE05=' Item 1 Value : ' + TO_TEXT(ITEM_1_PAY_VALUE)
LINE06=' Item 2 Value : ' + TO_TEXT(ITEM_2_PAY_VALUE)
LINE07=' Bonus Value  : ' + TO_TEXT(BONUS_AMOUNT)
LINE08='                _____'
LINE09='                Total : '+ TO_TEXT(trunc((
(SALARY_ANNUAL/12)+ITEM_1_PAY_VALUE+ITEM_2_PAY_VALUE+BONUS_AMOUNT)
,2))
LINE10=''
LINE11=''
LINE12=''
RETURN LINE01, LINE02, LINE03, LINE04, LINE05, LINE06, LINE07,
        LINE08, LINE09, LINE10, LINE11, LINE12,
```



---

## Checking an Element Entry

You can use FastFormula to validate user entries in element input values. For example, you can make sure that entries are within a specified range or do not exceed a predefined value.

The formula below checks that the entry value of the Salary element does not exceed 200,000.

**Figure 1 – 1 Salary Element**

Salary		
Classification	Priority	Type
Earnings	2500	R
Value Name	Type	Valid
– Pay Value	Money	F

```
/* Formula Name: Salary Range */
/* Formula Type: Element Input Validation */

INPUTS ARE entry_value (text)
IF TO_NUM(entry_value) > 200000
THEN
(
  formula_status = 'e'
  formula_message = 'Too much money . . . try again!'
)
ELSE
(
  formula_status = 's'
  formula_message = 'Fine'
)
RETURN formula_status, formula_message
```

---

## Checking a User Table Entry

You can use FastFormula to validate user entries into user tables that you define. For example, you can make sure that entries are between a specified range or not a negative amount.

The formula below checks that the deduction 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
```

---

## Writing Formulas for Templates

There are several ways you can use formulas to configure the people management templates:

- A Template Validation formula can check values entered in a field.
- A Template Information formula can specify information to be displayed from the right mouse menu when a user right-clicks in a field.
- A People Management Message formula can return a text string to display in the Assignment field on the Maintenance window and in the Data Organizer.
- A People Management Message formula can return message tokens that you can use in a notification message issued from template forms.

### Template Validation Formulas

If you use a formula to validate user entries in template fields, you must observe the following rules:

- Select the formula type Template Validation in the Formulas window.
- There can be up to five inputs, and they must be called item1, item2, item3, item4, and item5.
- The formula can return up to three outputs, which must be named as follows:
  - Status, which must have the value 's' if the validation was successful. Any other value is interpreted as an error.
  - Message, which is a text variable. The formula can return a message with validation statuses of success, failure, or both.
  - Item, which is the new value of the field that is being validated.

After creating the formula, you select it in the Validation Formula property for the field when you are setting up the template in Forms Configurator. You can also enter up to five parameters to be passed to the formula, including the value you are validating. For example, you might enter the name of another field on the template if you want the formula to cross-validate the value in one field against another.

#### Example

Here is a formula that validates the entry in the Organization field on the Maintenance window. It raises an error if the entry is not Corporate Finance.

```

/* Updateable Values Section */
/* Defaults Section */
/* Inputs Section */
INPUTS ARE item1
/* Main Body of Formula */
organization_name = item1
status = 's'
message = ' '
IF organization_name <> 'Corporate Finance' THEN
(
  message = 'Organization must be Corporate Finance'
  status = 'f'
)
RETURN message, status

```

When you are setting up the template in Forms Configurator, select the Organization (Maintain) item. Select the name of your formula as the Validation Formula property and select Organization (Maintain) as the Validation Formula Parameter 1 property. This passes and validates the value the user enters, not any codes or table identifiers to which it may be related.

## Template Information Formulas

If you write a formula to return additional information for a field, you must observe the following rules:

- Select the formula type Template Information in the Formulas window.
- There can be up to five inputs, and they must be called item1, item2, item3, item4, and item5.
- The formula can return one text output, which must be called Message. This is the information that is displayed when the user right clicks in the field and selects a prompt that you define in the right mouse menu.

After creating the formula, you select it in the Information Formula property for the field when you are setting up the template in Forms Configurator. You can also enter up to five parameters to be passed to the formula. For example, if you were specifying an information formula for the job field, you might enter the name of the organization field so that the formula can return different additional job information depending on the organization.

### Example

Here is a formula that returns contact information to be displayed from the Supervisor field.

```

/* Updateable Values Section */
/* Defaults Section */
DEFAULT FOR sup_work_phone IS ' '
DEFAULT FOR sup_email_address IS ' '
/* Inputs Section */
/* Main Body of Formula */
message = 'Supervisor Contact Details||CHR(10)||'Telephone:
' ||sup_work_phone||CHR(10)||'Email: ' ||sup_email_address
RETURN message

```

## People Management Message Formula for Assignment Field

Applicant and employee assignments are listed by name in the Data Organizer in the template Summary window and in the Assignment field on the Maintenance window. Since assignments do not have names, you can choose what assignment information is displayed as a name to help your users select the assignments they want to work with.

By default, assignments are identified as <job>.<organization>, such as Senior Manager.Engineering. However, you can choose any other database items to display.

If you want to override the default, write a formula (of type People Management Message) called ASSIGNMENT\_NAME. This formula must return a text string. If there is no formula called ASSIGNMENT\_NAME on the database, the system uses the predefined formula QH\_ASSIGNMENT\_NAME.

The QH\_ASSIGNMENT\_NAME formula is as follows:

```

/* Updateable Values Section */
/* Defaults Section */
DEFAULT FOR asg_job IS ' '
DEFAULT FOR asg_org IS ' '
/* Inputs Section */
/* Main Body of Formula */
assignment_name = asg_job||'.'||asg_org
RETURN assignment_name

```

## People Management Message Formulas for Message Tokens

If you write a formula to return message tokens, you must observe the following rules:

- Select the formula type People Management Message in the Formulas window.
- The formula must have the same name as the notification message that will use the tokens.

- The formula can return up to five text outputs, which must be named FF1, FF2, FF3, FF4 and FF5.

### Example

Suppose you want to include an employee's preferred name in a New Hire notification. The New Hire notification is called NEW\_STARTER, so you create a formula of the same name that returns this name in the variable FF1:

```
/* Updateable Values Section */
/* Defaults Section */
DEFAULT FOR per_known_as IS ' '
/* Inputs Section */
/* Main Body of Formula */
FF1 = per_known_as
RETURN FF1
```

Then you edit the New Hire notification to include the FF1 variable. For example:

Please note, that we have a new employee as of &HIRE\_DATE.

They are &FULL\_NAME (&EMPLOYEE\_NUMBER).

Known as: &FF1.

Their new job details are:

Position:	&POSITION
Job:	&JOB
Organization:	&ORGANIZATION
Location:	&LOCATION

CHAPTER

# 2



## Oracle FastFormula Reference

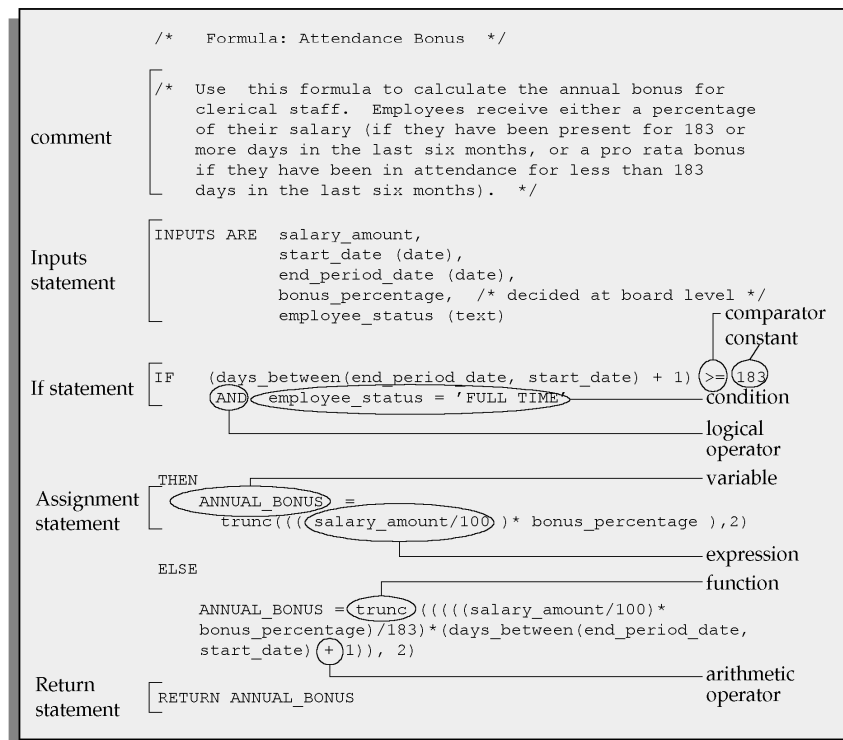
## Formula Reference

Formulas comprise *statements* and *comments*. Statements are instructions telling Oracle FastFormula how to process *constants* and *variables*, which are the basic information units in a formula. The two key types of statement, which describe the formula's calculations, are the Assignment statement and the If statement. These statements can include *expressions*, which manipulate constants and variables using *arithmetic operators* and *functions*. The operators and functions you can use and the results they give depend upon the data type of the constants and variables. In If statements, one expression can be compared to another using *comparators* to create a *condition*. Conditions can be combined using *logical operators*.

There are rules about how you use each of these components. Refer to the component description for more detailed information.

An example of each of these formula components is identified in the sample formula.

Figure 2 – 1 Components in a Sample Formula





---

## Input Values in Payroll Formulas

In many formulas for calculating pay, some or all of the information you need comes from the input values of the element.

For example, suppose you pay some salaried employees using a recurring element called Salary. The Salary element has an input value called `annual_salary`. You need a formula that divides the input value into twelve parts:

```
INPUTS ARE annual_salary
Salary = annual_salary/12
RETURN Salary
```



**Attention:** When you use an Inputs statement, you need to make sure that none of the input values can have a value of null because this causes the formula to fail. You can avoid this problem by using the Default statement.

Using an Inputs statement is the most efficient way to access the input values of the element with which the formula is associated. However, if the formula uses the input values of other elements, it must access the database items for them.

For example, if you want to use the input value `annual_salary` in the formula to calculate the element Bonus, you use the database item as follows:

```
IF Salary_annual_salary > 20000
THEN
Bonus = bonus_rate * (sales_achieved - sales_threshold)
```

Notice that the database item name is in two parts: the input value (`annual_salary`) name prefixed by the element name (`Salary`). This is the naming convention for the database items of element input values.

### Multiple Entries of Element Input Values

When you define an element, you can enable multiple entries of the element within a payroll period.

For example, suppose you use the element Wages to pay some weekly-paid employees. The Wages element has the input value `hours_worked`. Each week, you regularly make five entries for the input value `hours_worked`.

To calculate Wages, you can multiply the hours worked each day by the employee's standard rate from the grade rates table, so that your formula looks like this:

```
INPUTS ARE hours_worked
Wages = hours_worked * standard_rate
RETURN Wages
```

During the payroll run, the formula processes five times, creating five separate pay values, one for each entry.

Now consider using the database item `Wages_hours_worked` instead of an `Inputs` statement. The database item for an entry value **sums up** all the entries made in the payroll period.

This is a great convenience when referring to input value totals for a payroll period. However, you must be sure that it is the totals that you want to use. In this example, using the database item produces the wrong result.

`Wages_hours_worked` gives a single value that is the **sum** of the five entries in each weekly payroll period. When the payroll runs, the formula processes five times, each time calculating wages using the total hours worked in the week.



**Attention:** If multiple entries are enabled for an element, be careful when using database items for the element's entry values. These database items hold the sum of all the entries. This includes entries made as indirect formula results during the payroll run.

### Database Items for Numeric Values Only

Entry value database items are **not** created for input values with units of character, date, or time when multiple entries are enabled for the element. This is because entry values of these types cannot be summed over the payroll period. Only numeric values can be summed. Numeric input values are those with units of:

- Hours
- Integer
- Money
- Number

Notice that other database items, such as `default`, `minimum`, or `maximum`, may be created for non-numeric input values.

---

## Constants

Every piece of information that you can manipulate or use in a formula is a constant or a variable.

The data type of a constant or variable describes what kind of data the constant or variable holds. Generally, you use constant and variables of the same data type in an expression.

*Constants* are actual values you use in a formula. Oracle FastFormula uses constants directly rather than holding them in a variable.

There are three types of constant:

- numeric
- text
- date

### Numeric Constants

Enter numeric constants without quotes. Precede negative numbers with a minus sign (-). Numbers may have a decimal component after a decimal point. Do not use exponents and floating point (scientific) notations. So, for example, you cannot use  $2^2$  or  $10e^{1.24}$  as numeric constants. Do not use commas or spaces in a number. So, for example, you cannot use 10,000 or 10 000.00 numeric constants.

Examples of valid numeric constants are:

- 63
- 3.55
- -2.3
- - 0.33
- - .2
- 10000

### Text Constants

Enclose text constants in single quotes. They may contain spaces. You can represent the single quote character in a text constant by writing it twice ("). Note that this is not the same as the double quote (").

Examples of valid text constants are:

- 'J. Smith'
- 'P O'Donnell'
- '1234'

- 'Manager'
- '12 Union Road'
- 'The Bonus this year is 23%'

### **Date Constants**

Date constants contain a date. Enclose dates in single quotes and follow immediately with the word date, in brackets. Use the format YYYY-MON-DD HH24:MI:SS or DD-MON-YYYY. It is recommended that you use the first format if you want to compile the formula under different language settings.

Examples of valid date constants are:

- '1989-03-12 00:00:00' (date)
- '12-MAR-1989' (date)

---

## Variables

You use *variables* in a formula to access information. Variables can have frequently changing values.

The data type of a variable determines the type of information the variable holds:

- numeric
- text
- date

You do not have to specify what type you want a variable to be. Oracle FastFormula works out the type from how you use the variable. For example, if you set a variable to 'J. Smith', this is interpreted as a text variable.

The system also warns you if you try to perform any inconsistent operations, such as trying to add a number to a text string.

There are three classes of variable in Oracle FastFormula:

<b>Local Variable</b>	Variables that occur in a single formula only.
<b>Global Value</b>	Values that can occur in any formula.
<b>Database Item</b>	Items that exist in the application's database.

The variable class determines how a formula uses it.

### Local Variables

*Local variables* occur in a single formula only. You can change a local variable within the formula by assigning a value to it using an Assignment statement.

You can use local variables to store data in a formula. You might want to hold data temporarily while you perform some other calculations, or to pass data back to the application.

Below is an example showing the use of a local variable, `annual_leave`.

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

### Global Values

*Global values* are visible from within any formula. Use global values to store information that does not change often, but you refer to frequently,

such as company name, or a location allowance that applies to many employees. Global values are datetracked so you can make date effective changes ahead of time.

You can never change a global value using a formula. You alter global values using the Globals window. The global value is the same across all formulas within a Business Group.

See: Defining Global Values: page 1 – 22

Below is an example using a global value.

```
/* Formula: HAZARD ALLOWANCE FORMULA */
IF basic_hours > hazard_limit
THEN
    hazard_allowance = 2.30
ELSE
    hazard_allowance = 2.00
RETURN hazard_allowance
```

In this example, `hazard_limit` is a global value, which has been preset to reflect the point at which workers' hazard payment increases.

### Database Items

*Database items* exist in the application database and have a label, hidden from users, that the system uses to find the data.

There are two types of database item:

- *Static* database items are predefined. They include standard types of information, such as the sex, birth date, and work location of an employee, or the start and end dates of a payroll period.
- *Dynamic* database items are generated from your definitions of:
  - elements
  - balances
  - absence types
  - grade rates and pay scale rates
  - flexfield segments

The name of your element, pay and input values, balance, absence type, grade rate, pay scale rate, or flexfield segment becomes part of the name of the generated database item. This helps you identify the database item you need when you display a list of all available items in the Formulas window.

Definitions of flexfield segments only become database items when you initiate the Declare Descriptive Flexfield process from the Submit

Requests window. The other definitions become database items immediately when you save them to the database.

See:

Static Database Items: page 2 – 44

Dynamic Database Items: page 2 – 61.

Database items are specific to the *context* in which you use them. For example, using the database item `hourly_rate` gives the appropriate hourly rate for the specific assignment being processed.

Like global values, database item values cannot be changed within a formula.

### Rules for Determining Variable Class and Data Type

The rules that determine the data type and class of variables in a formula are:

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

```
INPUTS ARE  salary_amount,
            start_date (date)
            frequency (text)
```

If you do not specify the variable type in the statement, Oracle FastFormula assumes it is numeric.

2. If the variable is not an input, Oracle FastFormula looks in the list of global values the first time the variable occurs. If the variable is in the list, Oracle FastFormula determines the data type from there.
3. If the variable is not in the list, Oracle FastFormula searches the list of database items. Again, if it is in the list, Oracle FastFormula knows the data type.
4. Finally, if Oracle FastFormula does not find the variable in either the global values or the database items, then it treats the variable as a local variable. It determines the data type from the way you use the variable.

Notice that if the variable is either a global value or a database item, then any attempt in your formula to alter the value of the variable causes an error.

If the variable is a local variable, it does not contain a value when it is first used in the formula. Therefore you must assign a value to the variable before you try to use it in a condition or expression. If you fail to assign a value, Oracle FastFormula fails when you attempt to verify or run the formula.



**Suggestion:** To avoid a failure, assign values to your local variables when they first appear in your formula.

### Naming Variables

Variables have names comprising one or more words. The words must be joined by underscores. The words must each start with an alphabetic character (A–Z) and can be followed by alphanumeric characters (A–Z, 0–9). The maximum size of a variable name is 80 characters.

Oracle FastFormula is not sensitive to case. So, for example, the variable called EMPLOYEE\_NAME is the same as the variable Employee\_name.

The following reserved words cannot be used as the names of variables:

ALIAS	AND	ARE	AS
DEFAULT	DEFAULTED	ELSE	EXECUTE
FOR	IF	INPUTS	IS
NOT	OR	RETURN	THEN
USING	WAS		

Also, any word consisting only of digits, as these could be mistaken for numbers.

You may find that the name of a database item or global value is too long to use conveniently in your formula. You can set up an alternative, shorter name for use within a formula. You set this up using the Alias statement.

See: Statements: page 2 – 34



---

## Expressions

Expressions combine constants and variables with arithmetic operators (+, -, \*, /) and functions to return a value of a certain data type. For example, the expression (3 + 2) returns a value of 5, and is of numeric data type.

The format of an expression is:

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

This means that a number of 'subexpressions' can combine in a single expression. For example, the subexpressions (3 + 2) and MONTHS\_BETWEEN(start\_date, end\_date) can combine in a single expression as follows:

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

Expressions can also be used inside functions, as in the following example:

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

### Data Type of Expressions

The rules for determining the data type of an expression are simple. Operands in an expression are normally of the same data type, and this is normally the data type of the expression as a whole. For example, in the following expression all the operands are numeric and the expression itself is numeric:

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

There are some exceptions to this. For example:

```
DAYS_BETWEEN(date1, date2)  
MONTHS_BETWEEN(date1, date2)
```

These have date operands, but return a numeric value.

So the expression:

```
4 + days_between(start_date, todays_date)
```

returns a numeric result.

---

## Arithmetic Operators

An expression may contain arithmetic operators, which determine how variables and constants are manipulated. For example, the operator "+" indicates that two items should be added together.

The division, subtraction, and multiplication operators can only be used with numeric operands. The operands can be variables, constants, or subexpressions. A formula error occurs if:

- the result of subtraction is too large a negative number
- the result of multiplication is too large
- the second operand of a division evaluates to zero

In both cases, 'too large' here is determined by the normal limits in the ORACLE database.

The addition operator can be used with numeric or text operands. If the result is greater than 240 characters, a formula error occurs.

Notice that you enclose text constants in single forward quotes only ('), not double quotes ("). For example, the statements:

```
string1 = 'will '  
result_string = 'Pigs ' + string1 + 'fly'
```

set the local variable `result_string` to 'Pigs will fly'.

---

## Functions

Oracle FastFormula provides functions that manipulate data in different ways. Some functions work on only one type of data, some can work on two, others work on all three data types.

The functions are described below, separated into the three data types and functions that convert between data types. Where a function returns a different data type result than the data type of its operands, the description explains this.

Some functions retrieve data from Oracle Applications tables. These are described in the sections: Functions To Get Values from Tables, and Functions for Accrual Type Formulas. Some functions enable you to set and use globals in SQL\*Plus. They are described in the section: Functions to Set and Get Globals.

The general form of a function is:

```
NAME OF FUNCTION(operand, operand, . . .)
```

Notice that, as with the operators, the operands of a function can be variables, constants, or complete expressions. If the operand is a text string, you must enclose it in quote marks.

## Text Functions

### DEBUG

DEBUG(*expr*)

This function accepts a string and uses a DBMS\_OUTPUT statement to output the string to the console. Use this function when you are testing a new formula to track its processing and identify where it is failing.

### CHR

CHR(*n*)

The CHR function returns the character having the binary equivalent to number operand *n* in the database character set.

Example

```
/* CHR (10) used to add a newline to the end of REPORT_TEXT2. */  
REPORT_TEXT2 = 'Warning the Transaction Limit has been exceeded'
```

## **GREATEST**

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

**GREATEST\_OF**(*expr*, *expr* [, *expr*] . . .)

The **GREATEST** function compares the values of all the text string operands. It returns the value of the operand that is alphabetically last. If there are two or more operands that meet the criteria, Oracle FastFormula returns the first.

## **INITCAP**

**INITCAP**(*expr*)

The **INITCAP** function returns the expression *expr* with the first letter of each word in uppercase, all other letters in lowercase. Words are delimited by white space or characters that are not alphanumeric.

## **INSTR**

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

The **INSTR** searches *expr1* beginning with its *n*th character for the *n*th occurrence of *expr2* and returns the position of the character in *expr1* that is the first character of this occurrence. If *n* is negative, Oracle FastFormula 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 Oracle FastFormula 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 (if *expr2* does not appear *m* times after the *n*th character of *expr1*) the return value is 0.

## **INSTRB**

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

The same as **INSTR**, except that *n* and the return value are expressed in bytes, rather than in characters. For a single-byte database character set, **INSTRB** is equivalent to **INSTR**.

## **LEAST**

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

**LEAST\_OF**(*expr*, *expr* [, *expr*] . . .)

The **LEAST** function compares the values of all the text string operands. It returns the value of the operand that is alphabetically first. If there are two or more operands that meet the criteria, Oracle FastFormula returns the first.

## LENGTH

LENGTH(*expr*)

The LENGTH function returns the number of characters in the text string operand *expr*.

**Note:** The data type of the result of this function is numeric.

## LENGTHB

LENGTHB(*char*)

The LENGTHB function returns the length of *char* in characters. If *char* has datatype CHAR, the length includes all trailing blanks. If *char* is null, this function returns null.

## LOWER

LOWER(*expr*)

The LOWER function returns the string operand *expr* with all letters lowercase. The return value has the same datatype as the argument *expr*.

## LPAD

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

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

Examples:

```
/* A is set to 'XYXYXhello' */  
A = LPAD ('hello', 10, 'XY')
```

```
/* A is set to 'hell' */  
A = LPAD ('hello', 4 )
```

## LTRIM

(*expr* [,*set*])

The LTRIM function returns the text string operand *expr* with all the leftmost characters that appear in *set* removed. The default for *set* is a blank. If none of the leftmost characters of *expr* appear in *set* then *expr* is returned

## Examples:

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

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

## REPLACE

*(expr, search\_string [,replacement\_string])*

The REPLACE function returns the text string operand *expr* with every occurrence of *search\_string* replaced with *replacement\_string*. If *replacement\_string* is omitted or null, all occurrences of *search\_string* are removed. If *search\_string* is NULL, *expr* is returned. REPLACE allows you to substitute one string for another as well as to remove character strings.

### Example:

```
SELECT REPLACE ('JACK and JUE', 'J', 'BL') "Changes"  
FROM DUAL  
Changes  
-----  
BLACK and BLUE
```

## RPAD

*(expr, n [,pad])*

The RPAD function returns the text string operand *expr* right-padded to length *n* with the sequence of characters in *pad*. The default 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 'helloXYXYX' */  
A = RPAD ('hello', 10, 'XY')
```

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

## RTRIM

*(expr [,set])*

The RTRIM function returns the text string operand *expr* with all the rightmost characters that appear in *set* removed. The default for *set* is a blank. If none of the rightmost 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')
```

## SUBSTRING

SUBSTR(*expr*, *m* [, *n*])

SUBSTRING(*expr*, *m* [, *n*])

The SUBSTRING function returns a substring of the text string operand *expr* of length *n* characters beginning at the *m*th character. If you omit the third operand, the substring starts from *m* and finishes at the end of *expr*.

**Note:** The first operand is a text operand. The second and third operands are numeric operands. The resulting data type of this function is text.



**Suggestion:** Always check string length before you start to substring. For 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 ...
```

Or, to check if *Tax\_code* is a string of at least two characters starting with 'GG':

```
IF Tax_code LIKE '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*)

The TRANSLATE function 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 are not in *from* are not 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 are removed from the return value. Oracle FastFormula interprets the empty string as null, and if this function has a null argument, it returns null.

## **TRIM**

TRIM(*trim\_character* FROM *trim\_source*)

The TRIM function allows you to trim heading or trailing characters (or both) from a character string. If *trim\_character* or *trim\_source* is a character literal, you must enclose it in single quotes. You can specify LEADING or TRAILING to remove leading or trailing characters. If you specify none of these, both leading and trailing characters are removed equal to *trim\_character*.

## **UPPER**

UPPER(*expr*)

The UPPER function converts a text string to upper case.

## **Numeric Functions**

### **ABS**

ABS(*n*)

The ABS function 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) returns 17

### **FLOOR**

FLOOR(*n*)

The FLOOR function returns the integer part of a numeric operand *n*.

If the value of the operand contains information after the decimal point, Oracle FastFormula discards that information and returns a whole number.



Example:

```
FLOOR(35.455) returns 35
```

## **GREATEST**

```
GREATEST(n, n [, n] . . .)
```

```
GREATEST_OF(n, n [, n] . . .)
```

The GREATEST function compares all the operands and returns the largest value.

## **LEAST**

```
LEAST(n, n [, n] . . .)
```

```
LEAST_OF(n, n [, n] . . .)
```

The LEAST function compares all the operands and returns the smallest value.

## **POWER**

```
POWER(m,n)
```

Returns *m* raised to the *n*th power. The base *m* and the exponent *n* can be any numbers, but if *m* is negative, *n* must be an integer.

## **ROUND**

```
ROUND(n [, m])
```

The ROUND function rounds off a numeric value *n* to *m* decimal places and a date depending on the format of *m*. For numeric values, the first operand is the value Oracle FastFormula rounds off, the second the number of places Oracle FastFormula rounds off to. For dates, ROUND returns *n* rounded to the unit specified by the format model of *m* such as Year or Day. Refer to the *SQL Language Reference Manual* for details of the valid formats you can specify.

Examples:

```
ROUND(2.3401, 2) returns 2.34
```

```
ROUND (2.3461, 2) returns 2.35
```

```
ROUND (TO_DATE('27-OCT-1992', 'DD-MON-YYYY'), 'YEAR')  
returns 01-JAN-1993
```

## ROUNDUP

ROUNDUP(*n* [, *m*])

ROUND\_UP(*n* [, *m*])

The ROUNDUP function rounds a numeric value *n* up to *m* decimal places. The first operand is the value to be rounded up, the second the number of places to round to. If the digits after the rounding point are zero, the value is unchanged. If the digits are not zero, the value is incremented at the rounding point.

Examples:

```
ROUND_UP(2.3401, 2) returns 2.35
```

```
ROUND_UP(2.3400, 2) returns 2.34.
```

## TRUNC

TRUNC(*n* [, *m*])

TRUNCATE(*n* [, *m*])

The TRUNC function rounds a numeric value *n* down to *m* decimal places. The first operand is the value to be rounded down, the second the number of places to round to. TRUNC also returns *n* with the time portion of the day truncated to the unit specified by the format model of *m*. If you omit *m*, *d* is truncated to the nearest day. The default model, 'DD', returns the date rounded or truncated to the day with a time of midnight.

Oracle FastFormula drops all digits (if any) after the specified truncation point.

Examples:

```
TRUNC(2.3401, 2) returns 2.34.
```

```
TRUNC(TO_DATE('27-OCT-1992', 'DD-MON-YYYY'), 'YEAR') returns  
01-JAN-1992
```

## Date Functions

### ADD\_DAYS

ADD\_DAYS(*date*, *n*)

The ADD\_DAYS function adds a number of days to a date. The resulting date accords with the calendar.

**Note:** Oracle FastFormula ignores any fractional part of the number *n*.

Example:

```
ADD_DAYS ('30-DEC-1990' (date), 6) returns 5 JAN 1991
```

## **ADD\_MONTHS**

**ADD\_MONTHS**(*date*, *n*)

The **ADD\_MONTHS** function adds a number of months to a date. The resulting date accords with the calendar.

**Note:** Oracle FastFormula ignores any fractional part of the number *n*.

## **ADD\_YEARS**

**ADD\_YEARS**(*date*, *n*)

The **ADD\_YEARS** function adds a number of years to a date. The resulting date accords with the calendar.

**Note:** Oracle FastFormula ignores any fractional part of the number *n*.

## **GREATEST**

**GREATEST**(*date1*, *date2* [, *date3*] . . .)

The **GREATEST** function compares all the operands and returns the latest date.

## **LAST\_DAY**

**LAST\_DAY**(*d*)

The **LAST\_DAY** function returns the date of the last day of the month that contains *d*. You might use this function to determine how many days are left in the current month.

## **LEAST**

**LEAST**(*date1*, *date2* [, *date3*] . . .)

The **LEAST** function compares all the operands and returns the earliest date.

## **DAYS\_BETWEEN**

**DAYS\_BETWEEN**(*date1*, *date2*)

The **DAYS\_BETWEEN** function returns the number of days between two dates. If the later date is first, the result is a positive number. If the earlier date is first, the result is a negative number. The number returned is also based on the real calendar.

**Note:** The result is a numeric data type.

Example:

```
DAYS_BETWEEN('1995/06/27 00:00:00' (date), '1995/07/03  
00:00:00' (date)) returns -5
```

## MONTHS\_BETWEEN

MONTHS\_BETWEEN(*date1*, *date2*)

The MONTHS\_BETWEEN function returns the number of months between two dates. If the later date is first, the result is a positive number. If the earlier date is first, the result is a negative number. The number returned is also based on the real calendar.

If the result is not a whole number of months (that is, there are some days as well), the days part is shown as a decimal.

**Note:** The result is a numeric data type.

## NEW\_TIME

NEW\_TIME(*d*, *z1*, *z2*)

Returns the date and time in zone *z2* when the date and time in zone *z1* are *d*. The arguments *z1* and *z2* can be any one of these text strings:

AST or ADT	Atlantic Standard or Daylight Time
BST or BDT	Bering Standard or Daylight 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 or YDT	Yukon Standard or Daylight Time

## NEXT\_DAY

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

The NEXT\_DAY function returns the date of the first weekday named by *expr* that is later than the date *d*. The argument *expr* must be a day of the week in your session's date language. The return value has the same hours, minutes, and seconds component as the argument *d*.

## Data Conversion Functions

Use data conversion functions to convert from one data type to another data type. For example, you could have an expression returning a number value for salary, which you want to include in a printed message (that is, a character value). To print the number as part of the message, you need to convert the value of salary from a number to a character value, using the TO\_TEXT function.

## CONVERT

(*expr*, *dest\_char\_set* [, *source\_char\_set*])

The CONVERT function converts a character string from one character set to another. The *expr* argument is the value to be converted. The *dest\_char\_set* argument is the name of the character set to which *expr* is converted. The *source\_char\_set* argument is the name of the character set in which *expr* is stored in the database. The default value is the database character set.

## INSTR

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

The INSTR function searches *expr1* beginning with its *n*th character for the *m*th occurrence of *expr2* and returns the position of the character in *expr1* that is the first character of this occurrence. If *n* is negative, Oracle FastFormula counts and searches backwards.

## NUM\_TO\_CHAR

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

Converts the number *n* from number data type to text data type using the specified format. This function is equivalent to the SQL TO\_CHAR function. For example:

```
NUM_TO_CHAR(amount, '$9,990.99')
```

This returns the amount with a leading dollar sign, commas every three digits, and two decimal places. Refer to the *SQL Language Reference Manual* for a full list of the valid number formats you can specify.

## TO\_DATE

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

Converts the expression *expr* of text data type to a date data type. The text expression must be of the form 'YYYY/MM/DD HH24:MI:SS' if no format is provided. The day and year must be in numeric form. For example:

```
/* legal */
date_1 = TO_DATE ('12 January 89', 'DD Month YY')
/* illegal */
date_1 = TO_DATE ('12 January Nineteen-Eighty-Nine',
                  'DD Month Year')
```

**Note:** When assigning date variables from constants it is much more efficient to say:

```
date_1 = '1989/01/12 00:00:00'(date)
```

**Note:** The text expression must be in the format of either YYYY/MM/DD HH24:MI:SS or DD-MON-YYYY if no format is provided.

## TO\_NUMBER

TO\_NUM(*expr*)

TO\_NUMBER(*expr*)

Converts the expression *expr* of text data type to a number data type. The expression must represent a valid number. So for example, you cannot convert an expression such as 'Type 24' but you can convert the text expression '1234'. For decimal values, you must always use a period as a decimal point, for example '4.5'.

## TO\_TEXT

TO\_TEXT(*n*)    TO\_TEXT (*date1* [, *format*])

TO\_CHAR(*n*)    TO\_CHAR(*date1* [, *format*])

DATE\_TO\_TEXT(*n*) (*date1* [, *format*])

The TO\_TEXT function converts:

- the number *n* from number data type to text data type. The default number format has the decimal point as a period, for example '4.5'.
- the date *date1* from date data type to text data type. The optional *format* should be a text string like 'DD/MM/YYYY'. The default format is 'YYYY/MM/DD HH24:MI:SS'.

For example:

```
birthdate = '21-JAN-1960' (date)
  msg = 'Birthdate is: ' + TO_CHAR (birthdate)
/* sets msg to 'Birthdate is: 1960/01/21 00:00:00' */
  msg = 'Birthdate is: ' + TO_CHAR (birthdate,
    'DD-MON-YY')
/* sets msg to 'Birthdate is: 21-JAN-60' */
  msg = 'Birthdate is: ' + TO_CHAR (birthdate,
    'DD Month Year')
/* sets msg to 'Birthdate is: 21 January Nineteen-Sixty' */
```

## Functions to Get Values From Tables

### GET\_LOOKUP\_MEANING

GET\_LOOKUP\_MEANING(*lookup\_type*, *lookup\_code*)

The GET\_LOOKUP\_MEANING function enables Oracle FastFormula to translate a lookup code into a meaning. This can be used for any descriptive flexfield items or developer flexfield items that are based on lookups.

Example:

```
GET_LOOKUP_MEANING ('ETH_TYPE', PEOPLE_GB_ETHNIC_ORIGIN)
```

### GET\_FTE\_VALUE

GET\_FTE\_VALUE(*number*)

The GET\_FTE\_VALUE identifies the FTE value for a given assignment on a specified date in the past from the PER\_ASSIGNMENT\_BUDGET\_VALUES table.

This function uses the following Input Parameters:

- P\_ASSIGNMENT\_ID (number) – this is a required input and identifies which assignment to retrieve FTE for..
- P\_QUERY\_DATE (date) – this is a required input and specifies the date at which to retrieve FTE.

### GET\_TABLE\_VALUE

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

The GET\_TABLE\_VALUE function returns the value of a cell in a user-defined table. The three text operands, which identify the cell (*table\_name*, *column\_name*, and *row\_value*), are mandatory. The date operand is optional. If it is not supplied, the function returns the cell value as of the effective date.

You cannot use this function in formulas for user table validation or QuickPaint reports.

Example:

```
GET_TABLE_VALUE('WAGE RATES', 'Wage Rate', Rate_Code)
```

## **RAISE\_ERROR**

**RAISE\_ERROR**(*application\_ID, message name*)

This function allows you to raise a functional error message from within a formula. It accepts an Application ID and the message\_name of an Oracle Applications error message to raise.

Example:

```
ERROR = RAISE_ERROR(800, 'error_name')
```

## **RATES\_HISTORY**

**RATES\_HISTORY**(*element or rate type name, date, element or rate type indicator, time dimension*)

This function uses information stored in the UK Element Attribution Information EIT and information about the assignment's contract type to calculate a payment rate as of the given date and expressed for the selected time dimension (such as hourly or annual). If the element or rate type indicator is R, the function sums the rates for all elements classified with the given rate type (which is stored against the element in the Rate Type Information EIT).

The time dimension parameter must be A (annual), D (daily), H (hourly), or P (periodic). The element or rate type parameter must be R (rate type) or E (element).

The function can also adjust the returned rate for FTE and length of service, if these factors are set to Yes in the Element Attribution Information.

## **Functions for Accrual Type Formulas**

In addition to the standard FastFormula functions, you may find the following functions useful for your Accrual and Carry Over formulas.

### **CALCULATE\_PAYROLL\_PERIODS**

This function takes no parameters; it uses the payroll id context. It calculates the number of payroll periods in one year for that payroll, and sets the global variable PAYROLL\_YEAR\_NUMBER\_OF\_PERIODS to that value. For example, the function would set the global variable to 12 for a calendar month payroll.



Example:

```
E = CALCULATE_PAYROLL_PERIODS
```

## **GET\_ABSENCE**

**GET\_ABSENCE**(*calculation date, start date*)

This function calculates the total amount of absence contributing to an accrual plan between two dates. It counts the whole of any absence that:

- has a start date and an end date, and
- starts on or between the two dates given as inputs

Example:

```
TOTAL_ABSENCE = GET_ABSENCE('01-JAN-1999'(date),  
'31-DEC-1999'(date))
```

## **GET\_CARRY\_OVER**

**GET\_CARRY\_OVER**(*calculation date, start date*)

This function returns the number of days or hours recorded on the Carry Over element entry with an effective date on or between the two input dates. If more than one element entry is effective between these dates, the function sums the hours or days.

Carry Over element entries may also have an expiry date, after which any unused carry over is lost. If the calculation date is after the expiry date, the function checks the absences taken between start and calculation date. If the person took absences totaling the carry over, the function returns total carry over because it was all used before it expired. If absences total less than the carry over, the function returns total absence time; the rest of the carryover is lost.

For example, if the carry over is 10 days and 6 days absence were taken up to the expiry date, the function returns 6. The other four days of carry over have expired and been lost.

## **GET\_NET\_ACCRUAL**

**GET\_NET\_ACCRUAL**(*calculation date, plan id, accrual start date, accrual latest balance*)

This function calls the accrual formula defined in the accrual plan to return the net accrual at the calculation date. The following contexts must be available to a formula calling this function: assignment id, payroll id, business group id, and assignment action id.

## **GET\_OTHER\_NET\_CONTRIBUTION**

**GET\_OTHER\_NET\_CONTRIBUTION**(*calculation date, start date*)

This function calculates the total amount of net contribution other than absences or carry over between two dates. It looks for element entries for all elements that have been added in the Net Calculation Rules window. It sums the days or hours from all entries with an effective date on or between the two input dates.

### **GET\_PAYROLL\_PERIOD**

**GET\_PAYROLL\_PERIOD**(*date*)

This function determines the payroll period spanning the input date and sets global variables containing the start and end date and the period number. It returns 0 if successful, and otherwise error.

This example shows how to use this function then use the GET\_DATE and GET\_NUMBER functions to retrieve the values it sets in the global variables:

```
E = GET_PAYROLL_PERIOD(Calculation_date)
Calculation_Period_SD = GET_DATE('PAYROLL_PERIOD_START_DATE')
Calculation_Period_ED = GET_DATE('PAYROLL_PERIOD_END_DATE')
Calculation_Period_PNUM = GET_NUMBER('PAYROLL_PERIOD_NUMBER')
```

### **GET\_ACCRUAL\_BAND**

**GET\_ACCRUAL\_BAND**(*number*)

This function determines the appropriate accrual band for the specified length of service. It sets global variables containing the ANNUAL\_RATE, UPPER\_LIMIT and CEILING values for the band. ANNUAL\_RATE is the amount that should accrue this accrual term. UPPER\_LIMIT is the length of service that must be completed for the employee to go to the next accrual band. CEILING is the maximum number of hours or days the employee can accrue at any time. The function returns 0 if successful, and otherwise error.

This example shows how to use this function then use the GET\_NUMBER function to retrieve the values it sets in the global variables:

```
IF (GET_ACCRUAL_BAND(Years_Service) = 0 THEN
(
Annual_Rate = GET_NUMBER('ANNUAL_RATE')
Upper_Limit = GET_NUMBER('UPPER_LIMIT')
Ceiling = GET_NUMBER('CEILING')
ELSE
( ... [processing for error] ....)
)
```

### **GET\_ASSIGNMENT\_STATUS**

**GET\_ASSIGNMENT\_STATUS**(*date*)

This function determine the assignment status at a given date. It populates the globals ASSIGNMENT\_EFFECTIVE\_SD, ASSIGNMENT\_EFFECTIVE\_ED and ASSIGNMENT\_SYSTEM\_STATUS. It returns 0 if successful, and otherwise error.

Example:

```
ERROR = GET_ASSIGNMENT_STATUS('01-JAN-1999'(date))
```

### **GET\_ASG\_INACTIVE\_DAYS**

**GET\_ASG\_INACTIVE\_DAYS**(*period start date, period end date*)

This function checks the assignment status on each day from period start date to period end date. It calls the function GET\_WORKING\_DAYS to calculate the total number of working days in the period (Mondays to Fridays) and subtracts the number of working days when the assignment was inactive. It returns the number of inactive working days.

### **GET\_PERIOD\_DATES**

**GET\_PERIOD\_DATES**(*date1, unit, date2, number*)

This function determines the start and end dates of a period of time with the duration specified by the unit input and the number (such as 2 months). Valid units are D (days), M, (months), and W (weeks). The period spans date1 and starts on a date that is a multiple of the unit duration on from date2 (or backwards from date2).

The function populates the globals PERIOD\_START\_DATE and PERIOD\_END\_DATE. It returns 0 if successful, and otherwise error.

Example:

```
Error = GET_PERIOD_DATES('1-FEB-1999'(date), 'M',  
'15-DEC-1998'(date), 1)
```

This example populates PERIOD\_START\_DATE with 15-JAN-1999 and PERIOD\_END\_DATE with 14-FEB-1999.

An example where the period starts before date2:

```
Error = GET_PERIOD_DATES('1-FEB-1999'(date), 'M',  
'15-APR-1999'(date), 2)
```

This example populates PERIOD\_START\_DATE with 15-JAN-1999 and PERIOD\_END\_DATE with 14-MAR-1999.

### **GET\_START\_DATE**

**GET\_START\_DATE**(*accrual start date, start of accrual term*)

This function returns the date at which the accrual formula should start calculating accruals.

- If there is no payroll balance holding gross accruals, the date is always the start of the accrual term.
- If there is a payroll balance and there are retrospective absence entries that have not already been used in an accrual calculation, the function returns the start date of the earliest of these entries.
- If there is a payroll balance and there are no unprocessed retrospective absence entries, the function returns accrual start date.

This date, which is passed into the accrual formula, is the day after either the Date Earned or the Date Paid of the last payroll period in which the assignment was processed—depending on the PTO Balance Type set for the business group.

**Note:** Although GET\_START\_DATE returns the start date of the earliest of any unprocessed retrospective element entries, this date is not currently used in the seeded accrual formulas. If GET\_START\_DATE finds any unprocessed retrospective element entries, the formula always calculates accruals from the beginning of the accrual term.

### **GET\_WORKING\_DAYS**

GET\_WORKING\_DAYS(*start date, end date*)

This function returns the number of working days (Mondays to Fridays) in the period from start date to end date.

### **PUT\_MESSAGE**

PUT\_MESSAGE(*expr*)

This function adds a message to the stack to be output at the end of the formula by the Accruals form.

Example:

```
E = PUT_MESSAGE('The assignment is not yet eligible for accrual')
```

## **Functions to Call a Formula**

These functions allow you to call another formula, either once or in a loop. They require all the contexts available to the Accruals formula type.

### **CALL\_FORMULA**

CALL\_FORMULA(*formula name*)

This function runs a named formula with no inputs and no outputs.

## **LOOP\_CONTROL**

**LOOP\_CONTROL**(*formula name*)

This function repeatedly calls another formula, which must have the return parameter 'CONTINUE\_LOOP'. The loop continues until the function detects a value other than 'Y' in CONTINUE\_LOOP. If it detects 'N', the function returns 0 (success); if it detects another value, the function returns 1 (error).

## **Functions to Set and Get Globals**

Using the following functions, you can set and use globals in SQL\*Plus from within your formulas.

### **SET\_TEXT, SET\_NUMBER, SET\_DATE**

**SET\_TEXT**(*variable name, value*)

**SET\_NUMBER**(*variable name, value*)

**SET\_DATE**(*variable name, value*)

These functions accept the name of a global variable and the value to be set. They determine whether the global exists and, if not, create a new global. They return 0 if successful and 1 if not successful.

Examples:

```
E = SET_NUMBER('UPPER_LIMT', 0)
E = SET_DATE('CONTINUOUS_SERVICE_DATE', service_start_date)
```

### **GET\_TEXT, GET\_NUMBER, GET\_DATE**

**GET\_TEXT**(*variable name*)

**GET\_NUMBER**(*variable name*)

**GET\_DATE**(*variable name*)

These functions accept the name of a global variable and return its value. If they cannot find the global, they return NULL.

Example:

```
Calculation_Period_SD = GET_DATE('PAYROLL_PERIOD_START_DATE')
```

### **CLEAR\_GLOBALS**

This function sets to NULL the value of all global variables that were set using SET\_TEXT, SET\_NUMBER, and SET\_DATE. There are no inputs. It returns 0 if successful and 1 if not successful.

## **REMOVE\_GLOBALS**

This function removes all global variables. There are no inputs. It returns 0 if successful and 1 if not successful.

## **ISNULL**

**ISNULL**(*variable name*)

A set of three functions that test whether a text, numeric, or date value is NULL. Returns Y if the value is NULL and N otherwise.

Example:

```
IF IS_NULL(VARIABLE_NAME) = 'Y' THEN  
ERROR = SET_NUMBER(VARIABLE_NAME, 0)
```

---

## Comments

A formula may contain any number of comments, which can be placed anywhere in the formula.

Comments start with the sequence `/*` (slash asterisk), and finish with `*/` (asterisk slash). Oracle FastFormula ignores all text within these comment delimiters.



**Warning:** Do not put a comment within a comment. This causes Oracle FastFormula to return a syntax error.

---

## Statements

Statements are instructions that Oracle FastFormula carries out. There are six types of statement you can use:

- Alias statement
- Assignment statement
- Default statement
- If statement
- Inputs statement
- Return statement

An If statement can have Assignment, Return, and other If statements nested within it, enabling you to build up powerful formulas.

### Order of Statements

1. Alias statements (if any)
2. Default statements (if any)
3. Input statement (if any)
4. Other statements

## Alias Statement

The format of the Alias statement is:

```
ALIAS varname1 AS varname2
```

where `varname1` is the name of an existing database item or global value, and `varname2` is a unique name not currently known to the system nor used previously in your formula.

Use the Alias statement to define another name, or alias, for existing variables in the system. You can declare aliases for database items and global values.

Alias statements must appear before any other statements in a formula.

## Default Statement

The format of the Default statement is:

```
DEFAULT FOR <varname> IS <constant>
```

where `varname` is an input value or database item, and `constant` is a constant value matching the data type of `varname`.



Use the Default statement to set a default value for an input value or database item. The formula uses the default value if the database item is empty or the input value is not provided when you run the formula.

You can use the Default statement with the 'WAS DEFAULTED' test to detect when a default value has been used. For example:

```
DEFAULT FOR hourly_rate IS 3.00
X = hours_worked * hourly_rate
IF hourly_rate WAS DEFAULTED
THEN
    MSG = 'Warning: hourly rate defaulted'
```

This example sets a default of 3.00 for the database item `hourly_rate`. If `hourly_rate` is empty (NULL) in the database, the formula uses the default value of 3.00 and issues a warning message.



**Attention:** You must use the Default statement for database items that can be empty. The Database Items window includes a check box labelled Default Required. This check box is checked for database items that can be empty. The Database Items window appears when you click the Show Items button on the Formulas window.

## Inputs Statement

The format of the Inputs statement is:

```
INPUTS ARE varname1(data type)[, varname2 (data type)] ...
```

Use the Inputs statement to pass input values from an element into a formula.

For example,

```
INPUTS ARE bonus      (number),
           start_date (date)
```

You do not need to declare the type of number variables because this is the default data type. You can define up to 15 input values for an element.

The Inputs statement must appear before the other formula statements except:

- any Alias statements, which must always be at the top of the formula
- any Default statements that provide default values for input values

## Input Variables or Database Items

Always use the Inputs statement to retrieve the input values of the element associated with the formula. Using a database item forces the

formula to execute the code and work out the path to retrieve the database item.

For example, the formula below:

```
INPUTS ARE wage_rate,  
           hours_worked  
wage = wage_rate * hours_worked  
RETURN wage
```

is more efficient than the second formula:

```
wage = wage_wage_rate * wage_hours_worked  
RETURN wage
```

## Assignment Statement

Use the Assignment statement to set a value for a local variable. The format of the Assignment statement is:

```
varname = expression
```

For example:

```
rate = hourly_rate + 14  
wage = hours_worked * rate
```

Oracle FastFormula evaluates the expression on the right hand side of the statement, and 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.

## IF Statement

Use the If statement to check a condition that controls whether a sequence of statements is executed.

There are two *clauses* in the If statement: the THEN and ELSE clauses.

- The THEN clause lets you define what to do if the conditions are true.
- The ELSE clause lets you define what to do if the conditions are not true.

The If statement is the only statement that can have other statements *nested* within it, including other IF statements.

## Format of Statement

The format of the If statement is:

```
IF [NOT] condition
[logical operator] [NOT] condition
THEN
    statement [statement ..]
ELSE
    statement [statement ..]
```

The If statement can consist of a single condition, or two or more conditions combined with logical operators. The logical operators are AND, OR and NOT. The first two combine the conditions logically, and the third negates a condition:

- The AND operator means that if both conditions are true, then their combination is true.
- The OR operator means that if either condition is true, then their combination is true.
- If the NOT operator precedes a condition, this inverts the truth of the condition. That is, if condition X is true, then NOT X is false.

## Format of Conditions

A condition itself has a valid format. This is:

```
expression comparator expression
```

The values represented by each expression are compared together in the way described by the comparator. The two expressions must both return the same data type. There are eight comparators, and the way they work depends upon the data type of the values they are comparing.

Comparator	Symbols	Data Types	Meaning
Equals	=	All	The condition is true if both expressions have exactly the same value. For text, the case of the expression must be the same. So, for example, 'Smith' is not equal to 'SMITH'.
Not Equal to	!= <> ><	All	The condition is true if the result of the first expression does NOT have the same value as the result of the second expression.
Greater than	>	All	The condition is true if the first expression is alphabetically after, or has a numerically greater value, or a later date than the second expression.

Comparator	Symbols	Data Types	Meaning
Less than	<	All	The condition is true if the first expression is alphabetically before, or has a numerically smaller value, or an earlier date than the second expression.
Greater than or equal to	>= =>	All	The condition is true if either the greater than OR the equal to comparator returns a true result.
Less than or equal to	<= =<	All	The condition is true if either the less than OR the equal to comparator returns a true result.
Like	<b>LIKE</b>	Text	The condition is true if the two text expressions match according to the rules of the LIKE syntax. You can include Wildcards in the text to allow searching for text that matches a pattern, or words that begin with a certain sequence of letters. <ul style="list-style-type: none"> <li>- percent sign (%) matches any number of characters in that position</li> <li>- underscore ( ) matches a single character occurrence in that position.</li> </ul>
Not Like	<b>NOT LIKE</b>	Text	The condition is true if the two text expressions do NOT match according to the rules of the LIKE syntax.

There is a special comparator called WAS DEFAULTED that you can use to test database items and input values. If there is no value available for an input value or database item, the formula uses a default value. The condition containing the WAS DEFAULTED comparator is True if a default value was used. For example:

```

DEFAULT FOR employee_middle_name IS ' '
IF employee_middle_name WAS DEFAULTED
THEN
/* special processing */

```

### Correct Use of Brackets

If you group more than one statement under the THEN or ELSE clauses, enclose the group of statements within brackets, that is ( and ). In the absence of brackets, Oracle FastFormula conditionally executes *only* the statement that immediately follows the THEN or ELSE clause. Any other statements are executed unconditionally. For example, when the following formula runs, High\_salary is always set to Y:

```
High_salary = 'N'  
IF Salary > 20000  
  THEN Tax = Salary * .25  
       High_salary = 'Y'
```

To prevent this, use brackets as follows:

```
High_salary = 'N'  
IF Salary > 20000  
  THEN  
  (  
    Tax = Salary * .25  
    High_salary = 'Y'  
  )
```

## Return Statement

Use the Return statement to return values in local variables to the application. Oracle FastFormula can pass back any number of variables. The variable does not need to contain a value.

Example:

```
/* Formula: LONDON ALLOWANCE FORMULA */  
INPUTS ARE this_months_extra (number)  
London_allowance = (grade_pay/20 + this_months_extra)  
RETURN London_allowance
```

Notice that you do not have to declare the data type of local variables in the Return statement (as the formula already knows the data type).

Oracle FastFormula stops executing the formula when it reaches the Return statement. Any statements after the Return statement are ignored.

---

## Formula Compilation

When you have created or edited a formula in the Formula window, you choose the Verify button to compile it.

If you need to compile many formulas at the same time, you can run the Bulk Compile Formulas process in the Submit Requests window. For example, you run this process when you upgrade your legislative information, which includes formulas.

**Note:** If you make any changes to a function after you have compiled a formula that uses it, you need to recompile the formula for the changes to take effect.

---

## Formula Errors

There are two types of error that can occur when using Oracle FastFormula:

- **Verify-time errors** occur in the Formulas window when you run the formula verification. An error message explains the nature of the error.

Common verify-time errors are syntax errors resulting from typing mistakes.

- **Run-time errors** occur when a problem arises while a formula is running. The usual cause is a data problem, either in the formula or in the application database.

The basic Oracle FastFormula errors that can occur at run-time are:

- **Uninitialized Variables:** An uninitialized local variable is one that has no value when the formula runs. The term 'uninitialized' means you have not assigned any value to the variable before you try to use it. This causes an error in all statements except the Return statement. For example:

```
IF (tax_band < 2000)
  THEN tax = salary / 8
IF (tax_band > 2000)
  THEN tax = salary / 10
IF tax > 1000
  THEN ...
```

This formula fails with an 'Uninitialized variable' message (for the variable tax) if the tax band is set to 2000.

- **Divide by Zero:** Dividing a number by zero is an operation that provides no logical result. If this situation ever arises, Oracle FastFormula passes a code back to the application indicating an error (the application then takes the appropriate action).

Always check for the possibility of a divide by zero error if there is any chance it could occur. For example, the formula:

```
x = salary/contribution_proportion
```

produces an error if the contribution proportion is set to zero. In this formula, check for the divide by zero condition as follows:

```
IF contribution_proportion = 0
  THEN (message = 'The contribution proportion is not
    valid.' RETURN message)
ELSE x = salary/contribution_proportion
```

- **No Data Found:** A database item supposed to be in the database was not found. This represents an error in the application data.

- **Too Many Rows:** The database item definition within the application caused more than one value to be fetched from the database.
- **Value Exceeded Allowable Range:** This can occur for a variety of reasons such as:
  - exceeding the maximum allowable length of a string (which is 240 characters)
  - rounding up a number to an excessive number of places, for example, round (1,100)
  - using an invalid date, for example, 39-DEC-1990.
- **Invalid Number:** This occurs only when a database item contains an item that does not make sense as a number.
- **Null Data Found:** A database item was found to have a null value when it should have had a non-null value. Use the Default statement for database items marked as Default Required in the Database Items window.



---

## Database Items

This topic lists the database items available to you in Oracle HRMS for writing formulas and defining QuickPaint reports. The database items are grouped into two listings:

- Static Database Items
- Dynamic Database Items

Static database items are shipped with the system and you cannot modify them. Dynamic database items are created by Oracle HRMS processes whenever you define new elements or other related entities.

---

## Static Database Items

Static database items are shipped with the system and you cannot modify them.

### Accrual Plan Information

Database item	Description
ACP_CARRIED_OVER_DATE	The effective date stored in the latest Carry Over element entry for the assignment and accrual plan
ACP_CARRIED_OVER_PTO	The amount of PTO stored for an assignment in the latest Carry Over element entry
ACP_CATEGORY	The category of accrual plan
ACP_CONTINUOUS_SERVICE_DATE	An employee's adjusted service date
ACP_ENROLLMENT_END_DATE	The end date of an employee's enrollment in the accrual plan
ACP_ENROLLMENT_START_DATE	The start date of an employee's enrollment in the accrual plan
ACP_INELIGIBLE_PERIOD_LENGTH	The length of the plan's ineligibility period (a number)
ACP_INELIGIBLE_PERIOD_TYPE	The units (e.g. months) for measuring the length of the plan's ineligibility period
ACP_NAME	The name of the accrual plan
ACP_SERVICE_START_DATE	The start date of an employee's period of service
ACP_START	The rule for determining the start date for new hires in the plan
ACP_TERMINATION_DATE	The end date of an employee's period of service
ACP_U_O_M	The units (hours or days) for accumulating PTO
PTO_ACCRUAL_PLAN_ID	The id of the accrual plan.

Database item	Description
PTO_DATE_PAID_CALCULATION_DATE	The last day of the period for calculating accruals in the payroll run (when the PTO Balance Type for the business group is Date Paid)
PTO_DATE_EARNED_CALCULATION_DATE	The last day of the period for calculating accruals in the payroll run (when the PTO Balance Type for the business group is Date Earned)
PTO_DATE_PAID_START_DATE	The first day of the period for calculating accruals in the payroll run (when the PTO Balance Type for the business group is Date Paid)
PTO_DATE_EARNED_START_DATE	The first day of the period for calculating accruals in the payroll run (when the PTO Balance Type for the business group is Date Earned)

## Applicant Information

Database item	Description
APL_DATE_END	The date the application ended
APL_DATE_RECEIVED	The date the application was received

## Employee Assignment Information

Database item	Description
ASG_ASSIGNMENT_SEQUENCE	This is used as a default for assignment number
ASG_BARGAINING_UNIT_CODE	The employee's bargaining unit code

<b>Database item</b>	<b>Description</b>
<b>ASG_DATE_FROM</b>	The date from which this assignment information is effective
<b>ASG_DATE_TO</b>	The date to which this assignment information is effective
<b>ASG_EMPLOYMENT_CATEGORY</b>	The employment category for the assignment
<b>ASG_EMPLOYMENT_CATEGORY_CODE</b>	The employment category code for the assignment
<b>ASG_END_TIME</b>	The standard end time for the assignment
<b>ASG_FREQ</b>	The frequency for which the assignment working hours are measured
<b>ASG_FREQ_CODE</b>	The working hours frequency code for the assignment
<b>ASG_FTE_VALUE</b>	The full-time equivalent budget actual value for the assignment
<b>ASG_FULL_TIME_FREQ</b>	The full-time frequency for the assignment
<b>ASG_FULL_TIME_HOURS</b>	The full-time working hours for the assignment
<b>ASG_GRADE</b>	The employee's grade
<b>ASG_GRADE_DATE_FROM</b>	The date from which this assignment grade information is effective
<b>ASG_GRADE_DATE_TO</b>	The date to which this assignment grade information is effective
<b>ASG_GROUP</b>	The employee's group
<b>ASG_HEAD_VALUE</b>	The head budget value for the assignment
<b>ASG_HOURS</b>	The standard number of working hours for the assignment
<b>ASG_INT_ADDR_LINE</b>	The internal address of the assignment
<b>ASG_JOB</b>	The employee's job

<b>Database item</b>	<b>Description</b>
<b>ASG_JOB_DATE_FROM</b>	The date from which this assignment job information is effective
<b>ASG_JOB_DATE_TO</b>	The date to which this assignment job information is effective
<b>ASG_LABOUR_UNION_MEMBER_FLAG</b>	Whether the assignment is a union member
<b>ASG_LAST_CHANGE_REASON</b>	The reason the salary was changed
<b>ASG_LAST_PERFORMANCE_DATE</b>	Last performance review date
<b>ASG_LAST_PERFORMANCE_LOCATION</b>	Last performance review location
<b>ASG_LAST_PERFORMANCE_RATING</b>	Last performance review rating
<b>ASG_LAST_PERFORMANCE_TYPE</b>	Last performance review type
<b>ASG_LAST_PROC_PAYROLL_NAME</b>	The payroll name the assignment was last processed
<b>ASG_LAST_PROC_PERIOD_ID</b>	The time period ID the assignment was last processed
<b>ASG_LAST_PROC_PERIOD_NAME</b>	The period name the assignment was last processed
<b>ASG_LAST_PROC_PERIOD_NUMBER</b>	The period number the assignment was last processed
<b>ASG_LAST_PROPOSED_SALARY_CHANGE</b>	The proposed salary change
<b>ASG_LAST_PROPOSED_SALARY_PERCENT</b>	The proposed salary change as a percentage
<b>ASG_LAST_SALARY_CHANGE_APPROVED</b>	Whether the last proposed salary change has been approved
<b>ASG_LAST_SALARY_DATE</b>	The last salary change date
<b>ASG_LOCATION</b>	The employee's location
<b>ASG_LOC_INACTIVE_DATE</b>	The date to which the location information is effective
<b>ASG_MANAGER</b>	Whether the assignment is a managerial assignment (yes/no)
<b>ASG_MONEY_VALUE</b>	The assignment's money budget actual value
<b>ASG_NEXT_PERFORMANCE_DATE</b>	Next performance review date

<b>Database item</b>	<b>Description</b>
<b>ASG_NEXT_SALARY_DATE</b>	The date of the next salary change
<b>ASG_NUMBER</b>	The assignment number
<b>ASG_ORG</b>	The employee's organization
<b>ASG_ORG_DATE_FROM</b>	The date from which assignment organization information is effective
<b>ASG_ORG_DATE_TO</b>	The date to which assignment organization information is effective
<b>ASG_PAYROLL</b>	The employee's payroll
<b>ASG_PERFORMANCE_REVIEW_FREQUENCY</b>	The performance review period for the assignment
<b>ASG_PERFORMANCE_REVIEW_PERIOD</b>	The performance review frequency for the assignment
<b>ASG_PER_STATUS_DP</b>	Personal status for the assignment (as of Date Paid)
<b>ASG_PFT_VALUE</b>	The PFT budget value for the assignment
<b>ASG_POSITION</b>	The employee's position
<b>ASG_POS_DATE_FROM</b>	The date from which this assignment position information is effective
<b>ASG_POS_DATE_TO</b>	The date to which this assignment position information is effective
<b>ASG_POS_END_TIME</b>	The standard end time for the assignment position
<b>ASG_POS_FREQ</b>	The frequency for which the assignment position's hours are measured
<b>ASG_POS_HOURS</b>	The standard number of working hours for the position
<b>ASG_POS_PROB_PERIOD</b>	The probation period for the assignment position
<b>ASG_POS_START_TIME</b>	The standard start time for the assignment position

<b>Database item</b>	<b>Description</b>
<b>ASG_PRIMARY</b>	Whether this is the employee's primary assignment (yes/no)
<b>ASG_PRIMARY_CODE</b>	The assignment's primary code
<b>ASG_PROB_END_DATE</b>	The probation period end date
<b>ASG_PROB_PERIOD</b>	The assignment's probation period
<b>ASG_PROB_UNITS</b>	The units of the assignment's probation period
<b>ASG_REC_FULL_NAME</b>	The full name for the recruiter
<b>ASG_RELIEF</b>	The relief position if the current position holder is absent
<b>ASG_SALARY</b>	The current salary for an employee
<b>ASG_SALARY_BASIS</b>	The payment basis (i.e. frequency) for the assignment, e.g. monthly
<b>ASG_SALARY_BASIS_ANNUALIZATION_FACTOR</b>	The payment basis pay annualization factor for the assignment
<b>ASG_SALARY_BASIS_CODE</b>	The payment basis lookup code for the assignment
<b>ASG_SALARY_BASIS_GRADE_ANNUALIZATION_FACTOR</b>	The payment grade basis pay annualization factor for the assignment
<b>ASG_SALARY_BASIS_NAME</b>	The salary basis name for the assignment
<b>ASG_SALARY_ELEMENT</b>	The display element name
<b>ASG_SALARY_ELEMENT_VALUE_NAME</b>	The display input value name
<b>ASG_SALARY_GRADE_RATE</b>	The display rate name
<b>ASG_SALARY_RATE_BASIS</b>	The salary rate basis
<b>ASG_SALARY_REVIEW_FREQUENCY</b>	The salary review frequency for the assignment
<b>ASG_SALARY_REVIEW_PERIOD</b>	The salary review period for the assignment
<b>ASG_START_DATE</b>	The start date of the assignment
<b>ASG_START_TIME</b>	The standard start time for the assignment

Database item	Description
ASG_STATUS	The primary status for the assignment
ASG_SUCESSOR	The position name that will succeed into this position
ASG_SUP_FULL_NAME	The full name for the supervisor
ASG_TYPE	Whether this is an employee or applicant assignment
ASG_VACANCY	The name of the vacancy applied for

## Contact Addresses

Database item	Description
CON_ADR_CITY	The name of the contact's town or city
CON_ADR_COUNTRY	The name of the contact's country
CON_ADR_DATE_FROM	The first date on which the contact can be contacted
CON_ADR_DATE_TO	The last date on which the contact can be contacted
CON_ADR_LINE_1	The first line of the contact's address
CON_ADR_LINE_2	The second line of the contact's address
CON_ADR_LINE_3	The third line of the contact's address
CON_ADR_PHONE_1	The contact's first telephone number
CON_ADR_PHONE_2	The contact's second telephone number
CON_ADR_PHONE_3	The contact's third telephone number
CON_ADR_POSTAL_CODE	The contact's postal code



Database item	Description
CON_ADR_REGION_1	The first line of the contact's region
CON_ADR_REGION_2	The second line of the contact's region
CON_ADR_REGION_3	The third line of the contact's region

## Contact Information

Database item	Description
CON_AGE	The contact's age
CON_APP_NUMBER	The contact's applicant number
CON_BENEFICIARY_FLAG	The contact's beneficiary flag
CON_CURRENT_APP	Whether the contact is a current applicant (yes/no)
CON_CURRENT_EMP	Whether the contact is a current employee (yes/no)
CON_DATE_END	The end date of the contact's relationship
CON_DATE_START	The start date of the contact's relationship
CON_DATE_OF_BIRTH	The contact's date of birth
CON_DEPENDENT_FLAG	The contact's dependent flag
CON_DISABLED	Whether the contact is disabled (yes/no)
CON_EMP_NUMBER	The contact's employee number
CON_END_DATE	The date to which this contact information is effective
CON_END_LIFE_REASON_ID	ID for the reason for the end of the relationship
CON_FIRST_NAME	The contact's first name
CON_FULL_NAME	The contact's full name

Database item	Description
CON_KNOWN_AS	The contact's preferred name
CON_LAST_NAME	The contact's last name
CON_MARITAL_STATUS	The contact's marital status
CON_MIDDLE_NAMES	The contact's middle names
CON_NATIONALITY	The contact's nationality
CON_PERSONAL_FLAG	Personal relationship flag
CON_PERSON_TYPE	The contact's person type – employee or applicant, for example
CON_RLTD_PER_RSIDS_W_DSGNTR	Whether the contact shares the same residence as the employee.
CON_RELATIONSHIP	The relationship of the contact to the employee
CON_SEQUENCE_NUMBER	Contact's sequence number
CON_SEX	The contact's sex
CON_START_DATE	The date from which this contact information is effective
CON_START_LIFE_REASON_ID	ID for reason for the start of the relationship
CON_THIRD_PARTY_PAY_FLAG	Third party payments relationship flag
CON_TITLE	The contact's title
CON_WORK_PHONE	The contact's work telephone number

## Contracts Information

Database item	Description
CTR_STATUS_MEANING	Contract status meaning
CTR_TYPE_MEANING	Contract type meaning

Database item	Description
CTR_STATUS	Contract status code
CTR_TYPE	Contract type code

## Employee Hire Information

Database item	Description
EMP_HIRE_DATE	The employee's hire date
EMP_LAST_PROCESS_DATE	The date the employee was last processed
EMP_LEAVING_REASON	The reason the employee left
EMP_PROJ_TERM_DATE	The employee's projected termination date
EMP_TERM_ACCEPTED_BY	The person who accepted the employee's notice
EMP_TERM_DATE	The employee's termination date

## Location Details

Database item	Description
LOC_ADR_LINE_1	The first line of the assignment's work address
LOC_ADR_LINE_2	The second line of the assignment's work address
LOC_ADR_LINE_3	The third line of the assignment's work address
LOC_ADR_POSTAL_CODE	The postal code for the assignment's work address
LOC_ADR_REGION_1	The first line of the assignment's region
LOC_ADR_REGION_2	The second line of the assignment's region

Database item	Description
LOC_ADR_REGION_3	The third line of the assignment's region
LOC_ADR_PHONE_1	The assignment's first work telephone number
LOC_ADR_PHONE_2	The assignment's second work telephone number
LOC_ADR_PHONE_3	The assignment's third work telephone number
LOC_ADR_CITY	The town or city where the assignment works
LOC_ADR_COUNTRY	The country where the assignment works

### Work Address Details (US only)

Database item	Description
LOC_ADR_US_COUNTY	The assignment's work county (US only)
LOC_ADR_US_STATE	The assignment's work state (US only)
LOC_ADR_US_STATE_CODE	The assignment's work state code (US only)

### Work Address Details (UK only)

Database item	Description
LOC_ADR_UK_COUNTY	The assignment's work county (UK only)

## Payroll Details

Database item	Description
PAY_ARREARS_FLAG	Value of arrears flag for payrolls
PAY_EARNED_CUT_OFF_DATE	The cut-off date of the earned period
PAY_EARNED_DIRECT_DEPOSIT_DATE	The direct deposit date of the earned period
PAY_EARNED_END_DATE	The end date of the earned period
PAY_EARNED_PAY_ADVICE_DATE	The pay advice date of the earned period
PAY_EARNED_PERIOD_ID	The ID of the time period of the earned period
PAY_EARNED_PERIOD_NAME	The period name for the earned period
PAY_EARNED_PERIOD_NUMBER	The period number for the earned period
PAY_EARNED_START_DATE	The start date of the earned period
PAY_PERIODS_PER_YEAR	The number of pay periods in the year
PAY_PROC_PERIOD_CUT_OFF_DATE	The cut off date for the payroll period
PAY_PROC_PERIOD_DATE_PAID	The date the payroll was paid
PAY_PROC_PERIOD_DIRECT_DEPOSIT_DATE	The direct deposit date for the payroll period
PAY_PROC_PERIOD_END_DATE	The end date of the payroll period
PAY_PROC_PERIOD_ID	The ID of the time period for the payroll
PAY_PROC_PERIOD_NAME	The period name for the payroll
PAY_PROC_PERIOD_NUMBER	The current period number for the payroll
PAY_PROC_PERIOD_PAY_ADVICE_DATE	The pay advice date for the payroll period
PAY_PROC_PERIOD_START_DATE	The start date of the payroll period

## People Addresses

Database item	Description
PER_ADR_CITY	The name of the person's town or city
PER_ADR_COUNTRY	The name of the person's country
PER_ADR_COUNTRY_CODE	The person's country code
PER_ADR_DATE_FROM	The first date on which the person can be contacted at this address
PER_ADR_DATE_TO	The last date on which the person can be contacted at this address
PER_ADR_LINE_1	The first line of the person's address
PER_ADR_LINE_2	The second line of the person's address
PER_ADR_LINE_3	The third line of the person's address
PER_ADR_PHONE_1	The person's first contact number
PER_ADR_PHONE_2	The person's second contact number
PER_ADR_PHONE_3	The person's third contact number
PER_ADR_POSTAL_CODE	The person's postal code
PER_ADR_REGION_1	The first line of the person's region
PER_ADR_REGION_2	The second line of the person's region
PER_ADR_REGION_3	The third line of the person's region

## Home Address Details (US only)

Database item	Description
PER_ADR_US_COUNTY	The person's county (US only)
PER_ADR_US_STATE	The person's state (US only)
PER_ADR_US_STATE_CODE	The person's state code (US only)

## Home Address Details (UK only)

Database item	Description
PER_ADR_UK_COUNTY	The person's home county (UK only)

## People Information

Database item	Description
PER_AGE	The person's age
PER_APPLICANT_NUMBER	The person's applicant number
PER_BENEFIT_GROUP_ID	The ID of the person's benefit group
PER_COORD_BEN_MED_PLN_NO	The benefits medical plan number for the person
PER_COORD_BEN_NO_CVG_FLAG	Whether there is any other benefits coverage
PER_CURRENT_APP	Whether the person is a current applicant (yes/no)
PER_CURRENT_EMP	Whether the person is a current employee (yes/no)
PER_DATE_OF_BIRTH	The person's date of birth
PER_DATE_OF_DEATH	The person's date of death
PER_DATE_VERIFIED	The date the employee last verified his or her personal data

Database item	Description
PER_DISABLED	Whether the person is disabled (yes/no)
PER_DPNT_ADOPTION_DATE	The person's dependent's adoption date
PER_DPNT_VLNTRY_SVCE_FLAG	Whether the dependent is on voluntary service
PER_EMP_NUMBER	The person's employee number
PER_FIRST_NAME	The person's first name
PER_FULL_NAME	The person's full name
PER_KNOWN_AS	The person's preferred name
PER_LAST_NAME	The person's last name
PER_MAIL_DESTINATION	The person's mail destination
PER_MARITAL_STATUS	The person's marital status
PER_MIDDLE_NAMES	The person's middle names
PER_NATIONALITY	The person's nationality
PER_NATIONAL_IDENTIFIER	The person's national identifier
PER_ORIGINAL_DATE_OF_HIRE	Date the person was first hired
PER_PERSON_TYPE	Type of person (employee or applicant, for example)
PER_PREFIX	The person's name prefix
PER_PREV_LAST_NAME	The person's previous last name
PER_RECEIPT_OF_DEATH_CERT_DATE	Date of receipt of the person's death certificate
PER_SEND_EXPENSES	Where to send the person's expenses (home/office)
PER_SEX	The person's sex
PER_SUFFIX	The person's name suffix
PER_TITLE	The person's title



Database item	Description
PER_USES_TOBACCO_FLAG	Whether the person uses tobacco
PER_WORK_PHONE	The person's work telephone number

## Recruiter Information

Database item	Description
REC_CURRENT_APP	Whether the recruiter is a current applicant (yes/no)
REC_CURRENT_EMP	Whether the recruiter is a current employee (yes/no)
REC_EMP_NUMBER	The recruiter's employee number
REC_GRADE	The recruiter's grade
REC_INT_ADDR_LINE	The recruiter's internal address
REC_JOB	The recruiter's job
REC_LOCATION	The recruiter's work location
REC_MANAGER	Whether the assignment is a managerial assignment (yes/no)
REC_ORG	The name of the recruiter's organization
REC_PERSON_TYPE	The recruiter's person type (employee or applicant, for example)
REC_POSITION	The recruiter's position
REC_WORK_PHONE	The recruiter's work telephone number

## Supervisor Information

Database item	Description
SUP_CURRENT_EMP	Whether the supervisor is a current employee (yes/no)
SUP_DATE_FROM	The date from which this supervisor information is effective
SUP_DATE_TO	The date to which this supervisor information is effective
SUP_EMAIL_ADDRESS	The supervisor's email address
SUP_EMP_NUMBER	The supervisor's employee number
SUP_GRADE	The supervisor's grade
SUP_INT_ADDR_LINE	The supervisor's internal address
SUP_JOB	The supervisor's job
SUP_LOCATION	The supervisor's work location
SUP_MANAGER	Whether the assignment is a managerial assignment (yes/no)
SUP_ORG	The supervisor's organization
SUP_PERSON_TYPE	The supervisor's person type
SUP_POSITION	The supervisor's position
SUP_WORK_PHONE	The supervisor's work telephone number

## Date Information

Database item	Description
SESSION_DATE	The effective date from FND_SESSIONS
SYSDATE	The system date

---

## Dynamic Database Items

Dynamic database items are created by Oracle HRMS processes whenever you define new elements or other related entities.

## Element Database Items

When you define a new element, Oracle HRMS runs a process to create a number of related database items for it. To ensure easy recognition of these items, the process adds the element name <ENAME> to each one. It also creates further database items for each pay and input value you use <INAME>.

Here is a list of database items created each time you define an element using the Element window:

Database item	Description
<ENAME>_BEN_CLASS	The element's benefit classification
<ENAME>_CLASSIFICATION	The element's classification
<ENAME>_CLOSED_FOR_ENTRY	Yes/no flag: translated into local language. If Yes, new element entries cannot be created but existing element entries can still be modified.
<ENAME>_CLOSED_FOR_ENTRY_CODE	Yes/no flag: If Yes, new element entries cannot be created but existing element entries can still be modified.
<ENAME>_COSTABLE_TYPE	The element's costable type (from lookup table)
<ENAME>_COSTABLE_TYPE_CODE	The element's costable type (code values)
<ENAME>_COUNT	The element entry count
<ENAME>_END_DATE	The date to which this element is effective
<ENAME>_INPUT_CURRENCY_CODE	The element's input currency code
<ENAME>_LENGTH_OF_SERVICE	The element's qualifying length of service
<ENAME>_OUTPUT_CURRENCY_CODE	The element's output currency code

Database item	Description
<ENAME>_PROCESSING_PRIORITY	The element's processing priority
<ENAME>_QUALIFYING_AGE	The element's qualifying age
<ENAME>_QUALIFYING_UNITS_CODE	The qualifying length of service units (code values)
<ENAME>_QUALIFYING_UNITS	The qualifying length of service units (from lookup table)
<ENAME>_REPORTING_NAME	The element's reporting name
<ENAME>_STANDARD_LINK	Yes/no flag: yes = standard, no = discretionary
<ENAME>_STANDARD_LINK_CODE	Yes/no flag: yes = standard, no = discretionary
<ENAME>_<INAME>_UNIT_OF_MEASURE	The element's unit of measure (from lookup table)
<ENAME>_<INAME>_UNIT_OF_MEASURE_CODE	The element's unit of measure (code values)
<ENAME>_<INAME>_DEFAULT	The element's default input value
<ENAME>_<INAME>_MIN	The element's minimum input value
<ENAME>_<INAME>_MAX	The element's maximum input value

In addition to the items above, Oracle HRMS creates the following four items for elements defined with multiple entries *not* allowed:

Database item	Description
<ENAME>_<INAME>_ENTRY_VALUE	The element value
<ENAME>_<INAME>_USER_ENTERED_CODE	Whether a value exists at the element entry level (yes/no)
<ENAME>_<INAME>_START_DATE	The element's start date
<ENAME>_<INAME>_END_DATE	The element's end date

In addition to the common list above, Oracle HRMS creates the following item for elements defined with multiple entries allowed whose input values are numeric (that is, hours, integer, money or number).

Database item	Description
<ENAME>_<INAME>_ENTRY_VALUE	The summed element values for the multiple entries

The units for '**<ENAME> <INAME> ENTRY VALUE**' are generated for both recurring and nonrecurring elements and are user-definable. Oracle HRMS modifies the definition text to retrieve the entry value in the unit of measure as specified in the **PAY\_INPUT\_VALUES\_F** table.

### Grade Rate Database Items

When you define a grade rate, Oracle HRMS runs a process to create a number of related database items for it. To ensure easy recognition of these items, the process adds the grade rate name <NAME> to each one.

Here is a list of database items created each time you define a grade rate using the Grade Rate window:

Database item	Description
GRADE_<NAME>_VALUE	The grade rate's value
GRADE_<NAME>_MINIMUM	The grade rate's minimum value
GRADE_<NAME>_MAXIMUM	The grade rate's maximum value

### Pay Scale Rate Database Items

When you define a pay scale rate, Oracle HRMS runs a process to create the following database item for it. To ensure easy recognition of this item, the process adds the rate name <NAME> to it.

Database item	Description
SPINE_<NAME>_VALUE	The pay scale rates value

## Descriptive Flexfield Database Items

When you define descriptive flexfield segments you make them available for use in QuickPaint by running the Create Descriptive Flexfield DB Items process from the Submit Requests window. This process creates database items for each of the descriptive flexfields listed below.

To ensure easy recognition of these items, the process adds the descriptive flexfield segment name <SEGMENT\_NAME> to each one.

Database item	Description
PEOPLE_<SEGMENT_NAME>	People descriptive flexfield database items
PAYROLLS_<SEGMENT_NAME>	Payroll descriptive flexfield database items
ASSIGNMENTS_<SEGMENT_NAME>	Assignment descriptive flexfield database items
GRADES_<SEGMENT_NAME>	Grade descriptive flexfield database items
ABSENCES_<SEGMENT_NAME>	Absence descriptive flexfield database items
ABSENCE_TYPES_<SEGMENT_NAME>	Absence Type descriptive flexfield database items
PERSON_ADDRESSES_<SEGMENT_NAME>	Person Address descriptive flexfield database items
EVENTS_<SEGMENT_NAME>	Events descriptive flexfield database items
JOBS_<SEGMENT_NAME>	Jobs descriptive flexfield database items
CONTACTS_<SEGMENT_NAME>	Contacts descriptive flexfield database items
PERIODS_OF_SERVICE_<SEGMENT_NAME>	Periods of Service descriptive flexfield database items
RECRUITMENT_ACTIVITIES_<SEGMENT_NAME>	Recruitment Activities descriptive flexfield database items
POSITION_<SEGMENT_NAME>	Position descriptive flexfield database items

Database item	Description
APPLICATIONS_<SEGMENT_NAME>	Applications descriptive flexfield database items
ORGANIZATION_<SEGMENT_NAME>	Organization descriptive flexfield database items

## Key Flexfield Database Items

When you define key flexfield segments you make them available for use in QuickPaint by running the Create Key Flexfield DB Items process from the Submit Requests window. This process creates database items for each of the key flexfields listed below.

To ensure easy recognition of these items, the process adds the key flexfield segment name <SEGMENT\_NAME> to each one.

Run this process for each of your Business Groups. If you define context-dependent key flexfield structures using BUSINESS\_GROUP\_ID as the reference field, the process creates database items for those flexfield segments as well. BUSINESS\_GROUP\_ID is the only reference field that the Create Key Flexfield DB Items process supports.

Database item	Description
GRADE_KF_<SEGMENT_NAME>	Grade key flexfield database items
JOB_KF_<SEGMENT_NAME>	Job key flexfield database items
POS_KF_<SEGMENT_NAME>	Position key flexfield database items
GROUP_KF_<SEGMENT_NAME>	Group key flexfield database items

## Absence Database Items

When you define an absence type, Oracle HRMS runs a process to create the following database item for it. To ensure easy recognition of this item, the process adds the absence type name <ABSENCE\_NAME> to it.

Database item	Description
<ABSENCE_NAME>_CUM_BALANCE	The cumulative balance for an absence type



---

## Formulas for Payroll Legislative Checks

Oracle FastFormula contains a formula type called Legislative Check that may have been set up by your localization team. This formula type can be set up to perform certain checks during a payroll run. For example, you can catch errors such as negative gross pay and cause the payroll run to fail appropriately.

Sometimes you may not want to run the legislation check in the payroll run so it is possible that this functionality can be enabled or disabled using the HR:Execute Legislative Check Formula within Run user profile.

---

## Formulas for Benefits Administration

Oracle FastFormula contains many formula types that you can use for benefits administration. For example, grandfathered clauses and other special case scenarios may require you to write a FastFormula rule that defines special treatment for a subset of your benefits participants.

You can use Oracle FastFormula to calculate:

- The number of Hours Worked by a person in a given period
- A person's length of service
- The maximum coverage amount for a life insurance plan
- Participation Eligibility
- Other benefits related functions.

After you write a formula for use with Standard or Advanced Benefits, you link that formula to your plan design by selecting the formula in the Rule field of the appropriate window.

## Total Compensation Formula Types

The following table lists the formula types you can use in administering benefits with Oracle HRMS.

Formula Type	Description	Contexts	Input Values	Return Value	Sample Rule
Hours Worked Calculation	Used to determine an Hours Worked amount to be used for Eligibility, Coverage or Benefit, Premium, and Rate calculations	Business_Group_Id, Effective_Date, Jurisdiction_Code, Organization_Id, Assignment Id, Pgm_Id, Pl_Typ_Id, Pl_Id, Opt_Id, Ler_Id	Hrs_Wkd_Fctr_Id; Value of the Hrs_Wkd_Fctr.Val, Min Val, Max Val	Amount	Determine total number of hours worked during the person's most recent pay period.
Age Calculation	Used to determine an Age value to be used for Eligibility, Coverage or Benefit, Premium, and Rate calculations	''	Value of the Age_Fctr.Val, Min Val, Max Val	Amount	
Length of Service Calculation	Used to determine a Service value to be used for Eligibility, Coverage or Benefit, Premium, and Rate calculations	''	LOS_Fctr_Id; Value of the LOS_Fctr.Val, Min Val, Max Val	Amount	
Compensation Calculation	Used to determine a Total Compensation amount to be used for Eligibility, Coverage or Benefit, Premium and Rate calculations	''	Comp_Lvl_Fctr_Id; Value of the Comp_Lvl_Fctr.Val, Min Val, Max Val	Amount	Sum the amounts for person's current Regular Salary defined balance as of the beginning of the year, Prior Year Total Commission benefits balance type, and Prior Year Total Bonus benefits balance type, and return the total amount.

Rate Value Calculation	Calculates a rate amount for a person. May be used to calculate a base or variable rate.	”	Acty_Base_Rt_Id or Vrbl_Rt_Id; Value of the Acty_Base_Rt.Val, Min Val, Max Val; Value of the Vrbl_Rt_Prfl.Val, Min Val, Max Val,	Amount	If a person is currently enrolled in this plan, or has been enrolled in this plan within the past 2 years, then the rate is equal to the rate person is currently paying. Otherwise, rate is equal to X.
Premium Value Calculation	Calculates a premium amount for a person.	”	Acty_Base_Rt_Id or Vrbl_Rt_Id or Actl_Prem_Id or Cvg_Amt_Calc_Mthd_Id; Value of the Acty_Base_Rt.Val, Min Val, Max Val	Amount	Premium amount is equal to .05 times Coverage amount less 50,000
Matching Amount Calculation	Calculates the matching amount.	”	”	Amount	People that work at Division A get matching amount of 5%. All others get 4% matching amount.
Minimum Coverage Amount Calculation	Calculates a minimum coverage or benefit amount for a person.	”	”	Amount	If in Benefits Group A, minimum amount is \$100. If in Benefits Group B, minimum amount is \$200.

Maximum Coverage Amount Calculation	Calculates a maximum coverage or benefit amount for a person.	”	”	Amount	Coverage maximum amount is equal to coverage amount for current enrollment; if no current enrollment, then maximum is \$100,000.
Period to Date Amount Calculation	Determines the maximum period to date amount a person may have for a particular activity rate.	”		Amount	
Coverage Amount Calculation	Calculates a coverage or benefit amount for a person.	”		Amount	
Age Determination Date	Determines the date from which age will be calculated (e.g., the first day of the next plan year).	”		Date	If person is in an Annual Enrollment event then return 10/1/99. If person is becoming eligible as a result of a new hire, rehire, return from leave or a change from part-time to full-time then return the event occurred on date.

Hours Worked Determination Date	Determines the date from which hours worked will be calculated (e.g., the first day of the next plan year).	”		Date	If person is in an Annual Enrollment event then return 10/1/99. If person is becoming eligible as a result of a new hire, rehire, return from leave or a change from part-time to full-time then return the event occurred on date.
Length of Service Date to Use	Determines the low date from which LOS will be calculated (e.g., original hire date or rehire date).	”		Date	Return Start Date from the person’s Absence Attendance row in order to determine the elapsed time that a person has been on a Leave of Absence.
Length of Service Determination Date	Determines the high date for which LOS will be calculated (e.g., the first day of the following month).	”		Date	Return Start Date from the person’s Absence Attendance row.
Compensation Determination Date	Determines the high date from which hours compensation will be calculated (e.g., the first day of the next plan year).	”		Date	
Action Type Due Date	Determines the date on which the Action Type must be completed (e.g. 90 days from the Life Event Creation Date).	”		Date	If person is currently enrolled, 60 days from life event creation date. If person is not currently enrolled, 30 days from life event creation date.

Participation Eligibility Start Date	Determines when eligibility for a person should start.	”		Date	Add 6 months to the event date and return.
Participation Eligibility End Date	Determines when eligibility for a person should end.	”		Date	If Organization on current assignment = A, return date equal to Event Date; otherwise, return date equal to Start of Following Month after event date.
Enrollment Coverage Start Date	Determines when enrollment coverage for a participant should start.	”		Date	Coverage start date based on how long you have been absent from a plan.
Enrollment Coverage End Date	Determines when enrollment coverage for a participant should end.	”		Date	If Calculated LOS is less than 5 years, then end of current month. If Calculated LOS is greater than or equal to 5 years, then end of 6 months after event date.
Dependent Coverage Start Date	Determines when coverage for a dependent should start.	”		Date	If notified of birth within 31 days of event, Coverage Start Date is Date of Birth. Otherwise, coverage start date is date of notification.

Dependent Coverage End Date	Determines when coverage for a dependent should end.	”		Date	If Contact Relationship Type = Spouse, coverage ends on date of event; if Contact Relationship Type = Child, coverage ends on end of the month following event.
Rate Start Date	Determines when a rate for an enrollment result for a participant should start.	”		Date	Rate starts on the date after the rate end date of the current enrollment.
Rate End Date	Determines when a rate for an enrollment result for a participant should end.	”		Date	Rate ends on the day before the person’s next pay period.
Participation Eligibility	Determines whether a person is eligible for the associated compensation object. (This is the rule used in the eligibility profile rule entities.)	Business_Group_Id, Effective_Date, Jurisdiction_Code, Organization_Id, Assignment Id, Pgm_Id, Pl_Typ_Id, Pl_Id, Opt_Id, Ler_Id, Balance_Date		Y/N	
Dependent Eligibility	Determines whether a person is eligible to be covered by a participant in a compensation object.	”		Y/N	Dependent is eligible if Employee Work Location is 001 and Dependent is under age 21 or if Employee Work Location is not 001 and Dependent is under age 25.



Enrollment Opportunity	Determines whether the compensation object should be an electable choice for the person.	Business_Group_Id, Effective_Date, Jurisdiction_Code, Organization_Id, Assignment Id, Pgm_Id, Pl_Typ_Id, Pl_Id, Opt_Id, Ler_Id		One of the BEN_ENRT_MTHD values: A or E; or N	As a result of a transfer event, the participant can only enroll in an HMO if they were previously enrolled in an HMO and that HMO is no longer available to them in their new location.
To Be Sent Date	Determines the date on which the communication should be sent to the person.	Business_Group_Id, Effective_Date, Jurisdiction_Code, Organization_Id, Assignment Id, Pgm_Id, Pl_Typ_Id, Pl_Id, Opt_Id, Ler_Id, Cm_Typ_Id		Date	If participant's organization is equal to {org1} or {org2} then return start of enrollment period – 14 days. If participant's organization is not equal to {org1} or {org2} then return start of enrollment period – 7 days.
Rounding	Rounds a number to the specified place or decimal.	None		Amount	Amount to be rounded – \$250 rounded to the next \$500
Percent Rounding	Rounds a percent to the specified place or decimal.			Amount	
Automatic Enrollment Method	Determines the conditions under which a person should be automatically enrolled in a compensation object.	Business_Group_Id, Effective_Date, Jurisdiction_Code, Organization_Id, Assignment Id, Pgm_Id, Pl_Typ_Id, Pl_Id, Opt_Id, Ler_Id		One of the BEN_ENRT_MTHD values: A or E; or N	Reinstate active benefits if rehired in the same plan year.

Deduction Schedule	Determines the deduction schedule to be used for this person.	''		One of the BEN_DE D_SCHE D values	If bargaining unit code is not null return ''Second Period In Month'' else return ''Every Pay Period''.
Payment Schedule	Determines the payment schedule to be used for this person.	''		One of the BEN_PY MT_SCH ED values	If bargaining unit code is not null return ''Second Period In Month'' else return ''Every Pay Period''.
Default to Assign Pending Action	Determines the applicable default option or benefit to assign to the person when his or her choice has been suspended.	''		One of the BEN_DF LT_TO_A SN_PND G_CTFN values	Reinstate the active benefits that a person had at the time that the person was previously active. (NOTE: cannot reinstate most recent benefits, as person could have some other elections, such as COBRA).
Enrollment Certification Required	Determines the conditions under which a person must provide certification in order to enroll or elect a particular plan or option in plan, or benefit.	''		Y/N	Participant does not need to provide certification if they were enrolled in the compensation object and had provided certification within the past 12 months.
Dependent Certification Required	Determines the conditions under which a person must provide certification for his or her designated dependents.	''		Y/N	

Beneficiary Certification Required	Determines the conditions under which a person must provide certification for his or her designated beneficiaries.	''		Y/N	
Waive Certification Required	Determines the conditions under which a person must provide certification when he or she waives participation.	''		Y/N	
Inspection Required	Determines whether inspection of the communication is required. If so, the "Inspection Flag" is set to 'yes' for this person's communication.	Business_Group_Id, Effective_Date, Jurisdiction_Code, Organization_Id, Assignment Id, Pgm_Id, Pl_Typ_Id, Pl_Id, Opt_Id, Ler_Id, Cm_Typ_Id		Y/N	If participant's division = "Corporate" or if participant's HCE indicator = "Y" return "YES".
Communication Appropriate	Determines for this communication and trigger, whether the communication should be sent; restricts to whom to send.	Business_Group_Id, Effective_Date, Jurisdiction_Code, Organization_Id, Assignment Id, Pgm_Id, Pl_Typ_Id, Pl_Id, Ler_Id, Cm_Typ_Id		Y/N	If person is in an Annual Enrollment event and today's date is within seven days of the enrollment period end date return "No".
Communication Type	Determines whether the communication should be sent.	Business_Group_Id, Effective_Date, Jurisdiction_Code, Organization_Id, Assignment Id, Pgm_Id, Pl_Typ_Id, Pl_Id, Ler_Id, Cm_Typ_Id		Y/N	

Mandatory Determination	Determines whether this option in plan should be assigned to a person (and not be optional) as part of the enrollment process.	Business_Group_Id, Effective_Date, Jurisdiction_Code, Organization_Id, Assignment Id, Pgm_Id, Pl_Typ_Id, Pl_Id, Opt_Id, Ler_Id		Y/N	
Postelection Edit	Performs edits on an enrollment result, e.g. is the spouse of the participant also enrolled; does the spouse of the participant work at the same company.	”		Y/N and Text	If the participant elects a non-waive plan under the spouse life plan type, the participant must elect an equal or greater amount of participant life insurance.
Partial Month Proration Method	Determines which value to use: Date Earned, Pay Period End Date, or Payment Date. This tells the proration process which date to use when determining how many pay periods remain, and when the element entry should start.	”		One of the Values of : BEN_PR TL_MO_DET_MT HD	If the participant is paid monthly and enrolls in a medical plan between the 8th and 15th of the month, then return 75% of the normal monthly price tag.

Partial Year Coverage Restriction	Determines the maximum coverage amount for partial years of coverage. Allows determination of values other than delivered values of: 1) Standard maximum as defined in the plan. 2) Prorate maximum based on the number of months remaining in the plan year.	”		Amount	
Partial Month Effective Date Determination	Determines which value to use: Date Earned, Pay Period End Date, or Payment Date. This tells the proration process which date to use when determining how many pay periods remain, and when the element entry should start.	”		One of the Values of : BEN_PR TL_MO_ EFF_DT_ DET	
Lack Certification Allow Reimbursement	Identifies cases where certification is waived.	”		Y/N	
Compensation Object Selection	Determines which compensation objects are to be included for processing in a concurrent manager process.	”		Y/N	Run the default process for the flex and nonflex programs only. Participation process selection mode: determine eligibility for all compensation objects that use derivable factors.

Person Selection	Determines which people are to be included for processing in a concurrent manager process.	”		Y/N	Select all ex-participants who were working at a specific organization.
Verify Family Member	Determines whether the person has one or more contacts of a specific type or types, e.g. spouse, spouse and one child, more than one child.	”		Y/N	
Five Percent Owner	Determines for this plan and regulation whether the person is a five percent owner as defined in the regulation.	”		Y/N	
Highly Compensated	Determines for this plan and regulation whether the person is considered to be ”highly compensated” as defined in the regulation.	”		Y/N	
Key Employee	Determines for this plan and regulation whether the person is considered to be a ”key employee” as defined in the regulation.	”		Y/N	
Break in Service Value	Break in Service Value	”		Amount	
Break in Service Determination	Determines whether a break in service has occurred and if this break should not be ignored.	”		Y/N	

Contribution Nondiscriminatory	Compares individual employee pretax contribution amounts to total pretax contributions for highly and non-highly compensated persons.	”		Y/N	
Coverage Nondiscriminatory	Compares the total number of persons eligible to participate in a plan minus the persons who are not eligible due to legislated allowable factors to the number actually participating.	”		Y/N	
Extract Person Data Element	Specifies person or assignment information to be included as a data element item. This rule type can also return the results of a calculation performed on person or assignment information.	Business_Group_Id, Effective_Date, Jurisdiction_Code, Organization_Id, Assignment_Id		Text	
Extract Person Inclusion	Specifies person or assignment level inclusion or exclusion for system extract.	”		Y/N	
Communication Usage	Determines whether a Communication Usage requirement has been satisfied. If so, then a communication should be triggered for this usage.	Business_Group_Id, Effective_Date, Jurisdiction_Code, Organization_Id, Assignment Id, Pgm_Id, Pl_Typ_Id, Pl_Id, Ler_Id, Cm_Typ_Id		Y/N	

Default Enrollment	Determines whether this option in plan should be assigned to a person as part of the default enrollment process.	Business_Group_Id, Effective_Date, Jurisdiction_Code, Organization_Id, Assignment Id, Pgm_Id, Pl_Typ_Id, Pl_Id, Opt_Id, Ler_Id			
Enrollment Period Start Date	Determines the date on which the enrollment period starts.			Date	
Enrollment Period End Date	Determines the date on which the enrollment period ends.	”		Date	
Pop-Up Message	Determines whether a pop-up message displays and it what form.	”		Y/N	For a participant who selects life insurance at 10x salary, display a warning message on the enrollment form if the participant’s salary is less than 50k.
Collapse Life Event Resulting Occurred On Date	When life events are collapsed this rule determines the date to use for the resulting life event.	”		Date	
Collapse Life Event Evaluation	Determines whether the life event should be collapsed and deleted, collapsed and voided, or neither.	”		One of the values of: BEN_EV AL_DT: Collapse or Void; or leave as is	



Vested Value	Determine the vested percent for a person.	Business_Group_Id, Effective_Date, Jurisdiction_Code, Organization_Id, Assignment Id, Pgm_Id, Pl_Typ_Id, Pl_Id, Opt_Id, Ler_Id, Balance_Date		Percent	
Change Dependent Coverage	Used to determine if a dependent can be 'Added Only', 'Removed Only', 'Added and Removed', or 'Neither'.	Business_Group_Id, Effective_Date, Jurisdiction_Code, Organization_Id, Assignment Id, Pgm_Id, Pl_Typ_Id, Pl_Id, Opt_Id, Ler_Id		Y/N	
Evaluate Life Event	Determines if this life event is valid, or if the life event information needs to be changed, e.g. status, voided date, unprocessed date.	Business_Group_Id, Effective_Date, Jurisdiction_Code, Organization_Id, Assignment Id, Ler Id		Y/N, status code, unprocessed date, processed date.	
Maximum Waiting Period Date to Use	Determines the low date from which the maximum waiting period will be calculated (e.g., original hire date or rehire date).	''		Date	
Waiting Period Value and UOM	Determines the waiting period and unit of measure to be applied to a person.	''		Amount and UOM	
Maximum Period of Enrollment Value and UOM	Determines the maximum enrollment period and unit of measure for a compensation object.	''		Amount and UOM	

Person Change Causes Life Event	Determines whether this life event is valid for a person based on the data that changed.	”		Y/N	
Related Person Change Causes Life Event	Determines whether this life event is valid for a related person based on the data that changed.	”		Y/N	
Participant Eligible to Rollover	Determines whether this person may roll over flex credits into a particular compensation object.	Business_Group_Id, Effective_Date, Jurisdiction_Code, Organization_Id, Assignment Id, Pgm_Id, Pl_Typ_Id, Pl_Id, Opt_Id, Ler_Id, Balance_Date		Y/N	
Payment Must Be Received	Determines whether a payment is missing or late.	”		Y/N	
Life Event Reason Timeliness	Determines whether a life event has been reported in a timely manner.	Business_Group_Id, Effective_Date, Jurisdiction_Code, Organization_Id, Assignment Id, Pgm_Id, Pl_Typ_Id, Pl_Id, Opt_Id, Ler_Id		Y/N	
Required Period of Enrollment	Determines the earliest deenrollment date for a person’s electable choice for a compensation object.	”		Date	

Rate Lower Limit	Calculates an amount used to determine the lower limit value to which an activity rate or variable rate value is compared.	Business_Group_Id, Effective_Date, Jurisdiction_Code, Organization_Id, Assignment Id, Pgm_Id, Pl_Typ_Id, Pl_Id, Opt_Id, Ler_Id, Balance_Date		Amount	
Rate Upper Limit	Calculates an amount used to determine the upper limit value to which an activity rate or variable rate value is compared.	”		Amount	
Coverage Lower Limit	Calculates an amount used to determine the lower limit for an activity or variable coverage or benefit.	”		Amount	
Coverage Upper Limit	Calculates an amount used to determine the upper limit for an activity or variable coverage or benefit.	”		Amount	
Premium Lower Limit	Calculates an amount used to determine the lower limit for an activity or variable premium.	”		Amount	
Premium Upper Limit	Calculates an amount used to determine the upper limit for an activity or variable premium.	”		Amount	

Maximum Period of Enrollment	Period of Enrollment Rule determines whether a person has been enrolled for the maximum length of time allowed for a plan or option in a plan.	Business_Group_Id, Effective_Date, Jurisdiction_Code, Organization_Id, Assignment Id, Pgm_Id, Pl_Typ_Id, Pl_Id, Opt_Id, Ler_Id		Y/N	
Maximum Period of Enrollment Determination Date	Determines the low date to be used when determining whether the person has been enrolled in a plan or option in plan for the maximum period of time.	Business_Group_Id, Effective_Date, Jurisdiction_Code, Organization_Id, Assignment Id, Pgm_Id, Pl_Typ_Id, Pl_Id, Opt_Id, Ler_Id		Date	
Partial Month Proration Value Calculation	Calculates a value for a partial month enrollment.	”		Percent	
Variable Rate Add On Calculation	Calculates a new value when a variable rate result is used.	”	Result of the Variable Rate Calculation	Amount	Multiply the result by 102%.
Segment Costing Method	Determines how an amount is to be costed based on the segment in the COST KEY ALLOCATION KEY FLEXFIELD.	”		Key Cost Allocation Flexfield Segment	
Extract Enrollment Data Element	Specifies enrollment information to be included as a data element item. This formula type can also return the results of a calculation performed on enrollment information.	Business_Group_Id, Effective_Date, Assignment Id, Pl_Id, Opt_Id, Ler_Id		Text	

Maximum Credit Rollover Value	Determines the maximum amount a person may rollover to another plan or option in plan.	Business_Group_Id, Effective_Date, Jurisdiction_Code, Organization_Id, Assignment Id, Pgm_Id, Pl_Typ_Id, Pl_Id, Opt_Id, Ler_Id		Amount	
Default Excess Credit Treatment	Determines how any excess credits are to be allocated.	Business_Group_Id, Effective_Date, Jurisdiction_Code, Organization_Id, Assignment Id, Pgm_Id, Pl_Typ_Id, Pl_Id, Opt_Id, Ler_Id		One of the values in BEN_DEFAULT_EXCESS_TRTMT_CD	
Prorate Annual Election Value	Determines how a minimum or maximum annual election value is prorated.	Business_Group_Id, Effective_Date, Jurisdiction_Code, Organization_Id, Assignment Id, Pgm_Id, Pl_Typ_Id, Pl_Id, Opt_Id, Ler_Id		Amount	
Extract Post Process	This formula type provides additional system extract processing and is called after all extracted records are written.	Business_Group_Id, Effective_Date	EXT_RSLT_ID (this is found on ben_ext_rslt and ben_ext_rslt_dtl tables).	Nothing. All processing should be via formula function. Commit will occur in calling program.	Allows you to insert intermittent totals, delete records, change sorting, format fields, update values, etc.

---

## User Defined Functions

Functions enable you to manipulate data in different ways. Some functions work on only one type of data, some can work on two, others work on three data types.

Many functions are supplied with Oracle FastFormula. See: Functions: page 2 – 13. In addition, you can create other functions as you require for use by Oracle FastFormula. Doing this involves two steps: creating a new PL/SQL function, then registering it in the system using the Define Function window.

You need to be aware of two restrictions when creating user defined functions. These are:

- The current release of Oracle Payroll enables you to define functions with IN type parameters only. This means that a user-defined function may only return one value. IN OUT and OUT type parameters are not supported.
- User defined functions should not have any commit, rollback, or savepoint statements as these could lead to data integrity problems with the payroll run. In addition, functions should not save data changes to the database because FastFormula is intended as a read-only tool.

---

## UK Only Functions

The following functions have been registered for use in the UK only formulas.

### **CALCULATE\_TIME\_WORKED**

Calculates the time worked between a start date and end date for an assignment.

### **CLASS1A\_YTD**

Calculate car benefit, year to date

### **COUNT\_ASSIGNMENTS**

Count of assignments an employee has

### **DIRECTOR\_WEEKS**

Number of weeks an employee has been a director

### **GET\_BACS\_PROCESS\_DATE**

Return the BACS processing date

### **NI\_ABLE\_DIR\_YTD**

NIable pay for a director, year to date

### **NI\_ABLE\_PER\_PTD**

NIable pay for a person with multiple assignments

### **NI\_CO\_RATE\_FROM\_CI\_RATE**

Find the NI contracted out rate from the CI rate

### **PAYMENT\_YTD**

Calculate car payment, year to date

### **PERIOD\_TYPE\_CHECK**

Test whether the period type is valid

### **SESSION\_DATE**

Return the session date

### **UK\_TAX\_YR\_END**

Find the end of the tax year

**UK\_TAX\_YR\_START**

Find the start of the tax year

**USER\_RANGE\_BY\_LABEL**

Lower bound of range from user table using row label

**USER\_VALUE\_BY\_LABEL**

Value from user table using row label

**VALIDATE\_BACS\_DATE**

Return the previous BACS process date to a given date

**VALIDATE\_USER\_VALUE**

Verify that a given value is in a user table.



---

## JP Functions

The following functions have been registered for use in Japanese legislative formulas:

### CHECK\_FORMAT

(expr, fmt)

The CHECK\_FORMAT function returns TRUE or FALSE to verify that the text string operand expr matches the fmt type as follows:

0:0-9	Numbers only, non-omissible
9:0-9	Numbers only, omissible
A:A-Z	Capital alphabet only, non-omissible
P:A-Z	Capital alphabet only, omissible
a:a-z	Small alphabet only, non-omissible
p:a-z	Small alphabet only, omissible
L:0-9, A-Z	Numbers and capital alphabet only, non-omissible
C:0-9, A-Z	Numbers and capital alphabet only, omissible
l:0-9, a-z	Numbers and small alphabet only, non-omissible
c:0-9, a-z	Numbers and small alphabet only, omissible

Example:

```
CHECK_FORMAT ('123456ABC', '999999PPL') returns TRUE
```

### CHECK\_DATE\_FORMAT

(date, date fmt)

The CHECK\_DATE\_FORMAT function returns TRUE or FALSE to verify that the date operand matches a given date fmt.

Example:

```
CHECK_DATE_FORMAT ('19990623', 'YYYYMMDD') returns TRUE
```

### GET\_LAST\_ASSACT

(date1, date2)

The GET\_LAST\_ASSACT function returns latest assignment\_action\_id with 'SEQUENCED' classification\_name between date1 and date2. Action\_types with 'SEQUENCED' classification\_name are as follows:

ACTION_TYPE	MEANING	CLASSIFICATION_NAME
B	Balance Adjustment	SEQUENCED
F	Advance Pay	SEQUENCED
I	Balance Initializa- tion	SEQUENCED
O	RetroPay	SEQUENCED
Q	QuickPay Run	SEQUENCED
R	Run	SEQUENCED
V	Reversal	SEQUENCED
Z	Purge	SEQUENCED

### **ORG\_EXISTS**

(organization\_id, org\_class)

The ORG\_EXISTS function returns TRUE or FALSE to verify that the organization ID is in the organization class..

Example:

```
ORG_EXISTS(12345, 'JP_KENPO')
```

---

## Sample Accrual Formula

This topic suggests how you can implement a whole range of accrual plan rules in your Accrual formula. The suggestions are based on a simple formula, which is similar to the seeded PTO\_SIMPLE\_MULTIPLIER formula. The sample formula is for a plan with the following rules:

- An accrual term of one calendar year starting 01 January.
- Monthly accrual periods and a fixed accrual of 2 days per month.
- An accrual ceiling of 20 days, fixed within the formula.
- Accrual for new hires begins on whichever of these dates is the latest: hire date, plan enrollment date, or continuous service date (which can be entered as an input value when you enroll an employee in a plan).

The top level formula repeatedly calls another formula in a loop to calculate the accrual for each period. Both the top level formula (PTO\_ONE\_YEAR\_MULTIPLIER) and the looping formula (PTO\_PERIOD\_ACCRUAL) are given below.

```
1.  /*-----
    NAME : PTO_ONE_YEAR_MULTIPLIER
    This formula calculates the dates between which
    an assignment is to accrue time
    -----*/

2.  DEFAULT FOR ACP_CONTINUOUS_SERVICE_DATE IS
    '31-DEC-4712' (date)
3.  DEFAULT FOR ACP_TERMINATION_DATE IS
    '31-DEC-4712' (date)
4.  DEFAULT FOR ACP_ENROLLMENT_START_DATE IS
    '31-DEC-4712' (date)
5.  DEFAULT FOR ACP_SERVICE_START_DATE IS
    '31-DEC-4712' (date)

6.  INPUTS ARE
    Calculation_Date (date)

7.  E = SET_NUMBER('CEILING', 20)
8.  E = SET_NUMBER('ACCRUAL_RATE', 2)

9.  Accruing_Frequency = 'M' /* Month */
10. Accruing_Multiplier = 1

11. E = SET_TEXT('ACCRUING_FREQUENCY',
                Accruing_Frequency)
```

```

12. E = SET_NUMBER('ACCRUING_MULTIPLIER',
                  Accruing_multiplier)

13. Beginning_Of_Calculation_Year = to_date
    ('0101' || to_char(Calculation_Date, 'YYYY'), 'DDMMYYYY')

14. E = SET_DATE('BEGINNING_OF_CALCULATION_YEAR',
                Beginning_Of_Calculation_Year)

15. /*-----
    Set the start and end dates of the first accrual
    period in the calculation year.
    -----*/

16. E = GET_PERIOD_DATES
    (Beginning_of_Calculation_Year,
     Accruing_Frequency,
     Beginning_Of_Calculation_Year,
     Accruing_Multiplier)

17. First_Period_SD = get_date('PERIOD_START_DATE')
18. First_Period_ED = get_date('PERIOD_END_DATE')

19. /*-----
    Set the Calculation_Date to the Termination Date
    if not null
    ----- */

20. IF NOT (ACP_TERMINATION_DATE WAS DEFAULTED) THEN
    (
21. IF (ACP_TERMINATION_DATE < Calculation_Date) THEN
    (
22. Calculation_Date = ACP_TERMINATION_DATE
    )
    )

23. /* -----
    Get the last whole period prior to the
    Calculation Date and ensure that it is within the
    Year (if the Calculation Date is the End of a
    Period then use that period)
    ----- */

24. E = GET_PERIOD_DATES(Calculation_Date,
                        Accruing_Frequency,
                        Beginning_of_Calculation_Year,
                        Accruing_Multiplier)

25. Calculation_Period_SD = get_date('PERIOD_START_DATE')

```

```

26. Calculation_Period_ED = get_date('PERIOD_END_DATE')

27. IF (Calculation_Date <> Calculation_Period_ED)
    THEN
    (
28.     E = GET_PERIOD_DATES
        (ADD_DAYS(Calculation_Period_SD,-1),
         Accruing_Frequency,
         Beginning_of_Calculation_Year,
         Accruing_Multiplier)
29.     Calculation_Period_SD = get_date('PERIOD_START_DATE')
30.     Calculation_Period_ED = get_date('PERIOD_END_DATE')
    )

31. /* -----
    Set the Continuous Service Global Variable using
    the Continuous Service Date (if it was entered
    when the employee enrolled in the plan) and
    otherwise using hire date, whilst also ensuring
    that the continuous service date is before the
    Calculation Period.
    ----- */

32. IF (ACP_CONTINUOUS_SERVICE_DATE WAS DEFAULTTED)
    THEN
33. (
        E = set_date('CONTINUOUS_SERVICE_DATE',
                    ACP_SERVICE_START_DATE)
    )
34. ELSE
    (
35.     E = set_date('CONTINUOUS_SERVICE_DATE',
                    ACP_CONTINUOUS_SERVICE_DATE)
    )

36. Continuous_Service_Date = get_date('CONTINUOUS_SERVICE_DATE')

37. /* -----
    Determine the date on which PTO actually starts
    accruing based on Continuous Service Date, the
    Start Date of the Calculation Year, and plan
    Enrollment Start Date. Remember, we have already
    determined whether to use hire date or CSD at
    lines 32 to 35 above.
    ----- */

38. Actual_Accrual_Start_Date =
        greatest(ACP_ENROLLMENT_START_DATE,
                Continuous_Service_Date,

```

First\_Period\_SD)

```
39.  /* -----
      Determine the start and end date of the first
      accrual period to use in the accrual calculation.
      Get the start and end dates of the accrual period
      in which the Actual Accrual Start Date falls. If
      the Actual Accrual Start Date does not fall on
      the first day of this period, start accruing from
      the next period.
      -----*/

40.  IF Actual_Accrual_Start_Date > First_Period_SD
      THEN
      (
41.    E = GET_PERIOD_DATES(Actual_Accrual_Start_Date,
                          Accruing_Frequency,
                          Beginning_Of_Calculation_Year,
                          Accruing_Multiplier)
42.    Accrual_Start_Period_SD = get_date('PERIOD_START_DATE')
43.    Accrual_Start_Period_ED = get_date('PERIOD_END_DATE')
44.    IF Actual_Accrual_Start_Date > Accrual_Start_Period_SD THEN
      (
45.      E = GET_PERIOD_DATES
          (add_days(Accrual_Start_Period_ED,1),
          Accruing_Frequency,
          Beginning_of_Calculation_Year,
          Accruing_Multiplier)
46.      Accrual_Start_Period_SD = get_date('PERIOD_START_DATE')
47.      Accrual_Start_Period_ED = get_date('PERIOD_END_DATE')
      )
      )
48.  ELSE
      (
49.    Accrual_Start_Period_SD = First_Period_SD
50.    Accrual_Start_Period_ED = First_Period_ED
      )

51.  /* -----
      Now set up the information that will be used
      when looping through the periods
      ----- */

52.  IF Calculation_Period_ED >= Accrual_Start_Period_ED THEN
      (
53.    E = set_date('PERIOD_SD',Accrual_Start_Period_SD)
54.    E = set_date('PERIOD_ED',Accrual_Start_Period_ED)
55.    E = set_date('LAST_PERIOD_SD',Calculation_Period_SD)
56.    E = set_date('LAST_PERIOD_ED',Calculation_Period_ED)
```

```

57.     E = set_number('TOTAL_ACCRUED_PTO',0)
58.     E = LOOP_CONTROL('PTO_PERIOD_ACCRUAL')
59.     Total_Accrued_PTO = get_number('TOTAL_ACCRUED_PTO')
      )

60. IF Accrual_Start_Period_SD > Calculation_Period_ED THEN
      (
61.     Accrual_Start_Period_SD = First_Period_SD
      )

62. Effective_start_date = Accrual_Start_Period_SD
63. Effective_end_date = Calculation_Date
64. Accrual_end_date = Calculation_Period_ED

65. RETURN Total_Accrued_PTO,
66.         Effective_start_date,
67.         Effective_end_date,
68.         Accrual_end_date

```

## Looping Formula

```

1.  /* -----
      NAME : PTO_PERIOD_ACCRUAL
      This formula calculates the dates between which an
      assignment is to accrue time
      -----*/

2.  /*-----
      Get the global variable to be used in this formula
      -----*/

3.  Continuous_Service_Date = get_date('CONTINUOUS_SERVICE_DATE')
4.  Total_Accrued_PTO = get_number('TOTAL_ACCRUED_PTO')
5.  Period_SD = get_date('PERIOD_SD')
6.  Period_ED = get_date('PERIOD_ED')
7.  Last_Period_SD = get_date('LAST_PERIOD_SD')
8.  Last_Period_ED = get_date('LAST_PERIOD_ED')
9.  Accrual_Rate = get_number('ACCRUAL_RATE')
10. Accruing_Frequency = get_text('ACCRUING_FREQUENCY')
11. Accruing_Multiplier = get_number('ACCRUING_MULTIPLIER')
12. Beginning_of_Calculation_Year =

get_date('BEGINNING_OF_CALCULATION_YEAR')
13. Ceiling = get_number('CEILING')

14. /* -----
      Calculate the Amount Accrued this Period
      -----*/

15. Period_Accrued_PTO = Accrual_Rate

```

```

16.  /*-----
      Calculate any absence or bought/sold time etc. to
      be accounted for in this period.
      -----*/

17.  Absence = GET_ABSENCE(Period_ED,
                          Beginning_of_Calculation_Year)
18.  CarryOver = GET_CARRY_OVER(Period_ED,
                                Beginning_of_Calculation_Year)
19.  Other = GET_OTHER_NET_CONTRIBUTION(Period_ED,
                                       Beginning_of_Calculation_Year)

20.  Period_Others = CarryOver + Other - Absence

21.  /* -----
      Now establish whether the Accrual this period has
      gone over the ceiling if one exists
      -----*/

22.  IF (Ceiling > 0) THEN
    (
23.    IF (Total_Accrued_PTO + Period_Accrued_PTO +
          Period_Others > Ceiling) THEN
      (
24.        Amount_Over_Ceiling = Total_Accrued_PTO +
                                Period_Accrued_PTO + Period_Others - Ceiling
25.        IF (Amount_Over_Ceiling > Period_Accrued_PTO)
           THEN
          (
26.            Period_Accrued_PTO = 0
          )
27.        ELSE
          (
28.            Period_Accrued_PTO = Period_Accrued_PTO -
                                    Amount_Over_Ceiling
          )
        )
      )
    )
29.  /*-----
      Set the Running Total
      -----*/

30.  E = set_number
      ('TOTAL_ACCRUED_PTO',Total_Accrued_PTO +
      Period_Accrued_PTO)

31.  /* -----

```



```

Establish whether the current period is the last
one, if so end the processing, otherwise get the
next period
-----*/

32. IF Period_SD = Last_Period_SD THEN
   (
33.   Continue_Processing_Flag = 'N'
   )
34. ELSE
   (
35.   E = GET_PERIOD_DATES(ADD_DAYS(Period_ED,1),
                        Accruing_Frequency,
                        Beginning_of_Calculation_Year,
                        Accruing_Multiplier)
36.   E = set_date('PERIOD_SD', get_date('PERIOD_START_DATE'))
37.   E = set_date('PERIOD_ED', get_date('PERIOD_END_DATE'))
38.   Continue_Processing_Flag = 'Y'
   )

39. Return Continue_Processing_Flag

```

## Changing the Length of the Accrual Periods

The accrual period is determined by the variables `Accruing_Frequency` and `Accruing_Multiplier` (lines 9 and 10 of the top level formula). You can set `Accruing_Frequency` to M (month), D (day) or W (week). For example, if the frequency is set to W and the multiplier set to 2, time is accrued every two weeks.

These examples use the calendar to measure the length of accrual periods. You can also use payroll periods. In this case, you do not need to set the variables `Accruing_Frequency` and `Accruing_Multiplier`. At lines 16, 24, 28, 41, and 45, replace the call to `get_period_dates` with a call to `get_payroll_period`. For example, replace lines 24 to 26 with:

```

E = GET_PAYROLL_PERIOD (ADD_DAYS(Calculation_Period_SD, -1))
CALCULATION_PERIOD_SD = get_date('PAYROLL_PERIOD_START_DATE')
CALCULATION_PERIOD_ED = get_date('PAYROLL_PERIOD_END_DATE')

```

## Changing the Accrual Term Start Date

The accrual term start date is set to 01 January at line 13 of the sample top level formula. To use another fixed date (such as 01 June) replace this line with the following:

```

Beginning_Of_Calculation_Year = to_date
                              ('0106' || to_char(Calculation_Date, 'YYYY'), 'DDMMYYYY')

```

```

If Beginning_Of_Calculation_Year > Calculation_Date then
(
  Beginning_Of_Calculation_Year =
    add_months(Beginning_Of_Calculation_Year, -12)
)

```

To start an employee's accrual term on his or her hire date anniversary, replace line 13 with:

```

Beginning_Of_Calculation_Year = to_date(
    to_char(ACP_SERVICE_START_DATE, 'DD-MM') ||
    to_char(Calculation_Date, 'YYYY'), 'DDMMYYYY')

```

```

If Beginning_Of_Calculation_Year > Calculation_Date then
(
  Beginning_Of_Calculation_Year =
    add_months(Beginning_Of_Calculation_Year, -12)
)

```

This example uses the ACP\_SERVICE\_START\_DATE database item, although any substitute may be used.

## Adding Start Rules for New Hires

In this sample formula, accrual for new hires begins on whichever of these dates is the latest: hire date, plan enrollment date, or continuous service date. However, you may want your top level accrual formula to check whether there is a start rule defined for the plan. The seeded PTO\_PAYROLL\_CALCULATION formula shows you how to do this.

In summary, the formula has to:

- Check which start rule was entered for the plan, and calculate the First\_Eligible\_To\_Accrue\_Date accordingly.
- Get the first full accrual period following the First\_Eligible\_To\_Accrue\_Date.
- Check whether the Actual\_Accrual\_Start\_Date is after the First\_Eligible\_To\_Accrue\_Date and, if not, set the Accrual\_Start\_Period start and end dates to the First\_Eligible\_To\_Accrue\_Period start and end dates. This affects lines 40 to 50 of the sample formula.

There are three seeded start rules: hire date, beginning of calendar year after hire date, and six months after hire date. If you need a different start rule, define it as a value for the Lookup Type US\_ACCRUAL\_START\_TYPE.

Then add a line to your formula, of this form:

```

IF (ACP_START = '<your new Lookup Value>') THEN
(First_Eligible_To_Accrue_Date = <calculation for start date>)

```

## Basing the Accrual Amount on Time or Pay Elements

You need to create database items for the element input values that determine the accrual amount. Suppose plan participants accrue one hour for every 10 hours worked. You could include this rule in your formula as follows:

```
Accrual = ACP_HOURS_WORKED / 10
```

where ACP\_HOURS\_WORKED is a database item. This kind of calculation would be found typically in the looping formula.

## Using Up Front Accruals

The sample formula assumes that plan participants accrue a certain amount of time each month. But in some plans, participants accrue their full entitlement at the start of the accrual term. In this case the formula does not need to loop through each accrual period.

Here is a very basic sample formula, assuming an accrual of 20 days for every calendar year. There are no ineligibility rules or start rules in this example. It also does not calculate the accrual for part years (for example, for employees joining the plan midway through a year).

```
INPUTS ARE
Calculation_Date (date)

Total_Accrued_PTO = 20
Effective_start_date =
    to_date('0101' || to_char(calculation_date, 'YYYY'))
Effective_end_date =
    to_date('3112' || to_char(calculation_date, 'YYYY'))
Accrual_end_date =
    to_date('0101' || to_char(calculation_date, 'YYYY'))

RETURN Total_Accrued_PTO,
        Effective_start_date,
        Effective_end_date,
        Accrual_end_date
```

## Changing the Ceiling

In the sample top level formula, the ceiling is set at line 7. You can change the ceiling within the formula, or set it to zero to remove the ceiling:

```
E = SET_NUMBER('CEILING', 0)
```

You can also set the ceiling outside the formula, using the Accrual Bands window or a user table.

## Changing the Date Used for Continuous Service

The continuous service date is used in the formula to determine when a new hire begins to accrue time. If your plan has accrual bands based on length of service, this date also determines the amount the employee is eligible to accrue.

In the sample, length of service is calculated from hire date (using the database item `ACP_SERVICE_START_DATE`) or the continuous service date input value, if it was entered when the employee was enrolled in the plan (database item `ACP_CONTINUOUS_SERVICE_DATE`). Alternatively, you can define and use another database item. For example, replace lines 32 to 36 of the top level formula with:

```
E = set_date('CONTINUOUS_SERVICE_DATE', <NEW_DATABASE_ITEM>)

Continuous_Service_Date = get_date('CONTINUOUS_SERVICE_DATE')
```

## Adding Rules for Suspended Assignments

Some accrual plans may not allow employees to accrue PTO while on certain types of leave, such as maternity leave or study leave. In this case, your formula needs to check the status of the assignment.

For example, suppose that assignments with the status “Suspended” cannot accrue time. You can use the `ASG_STATUS` database item to get the value of the assignment status. Replace line 8 with:

```
IF ASG_STATUS = 'Suspended' THEN
(
  E = SET_NUMBER('ACCRUAL_RATE', 0)
)
ELSE
(
  E = SET_NUMBER('ACCRUAL_RATE', 2)
)
```

## Adding Rules for Part Time Employees

You can use the `ASG_EMPLOYMENT_CATEGORY` database item to check whether the assignment is part time or full time.

Suppose part timers accrue at only half the rate of full timers, and have a lower ceiling, then replace lines 7 and 8 with:

```
IF ASG_EMPLOYMENT_CATEGORY = 'FULL TIME' then
(
  E = SET_NUMBER('CEILING', 20)
  E = SET_NUMBER('ACCRUAL_RATE', 2)
)
```

```

ELSE
(
  E = SET_NUMBER('CEILING', 10)
  E = SET_NUMBER('ACCRUAL_RATE', 1)
)

```

If part time employees accrue at different rates depending on their hours as a percentage of full time, you could set up HR budgets to record the value of each assignment. Then you need to define a database item for the budget value.

## Adding a Long Service Leave Entitlement

If you want to see how much long service leave has been awarded to an employee in the Accruals window, you need to set up a separate accrual plan. Otherwise, you can include the entitlement in the standard accrual plan. The following approach adds the long service accrual in a single accrual period, ignoring the usual accrual ceiling.

Suppose employees are entitled to a one time bonus of 10 extra days after 15 years of service. First, in the top level formula, add the following after line 57 to set up a variable that can be accessed in the looping formula:

```
e = set_number('LONG_SERVICE_ACCRUAL', 0)
```

Then, in the looping formula, calculate years service to ascertain whether the employee is entitled to long service leave. Add the following after line 13:

```
years_service = floor(months_between(period_ed,
                                     continuous_service_date)/12)
```

Next, set up a variable that can be used later to detect whether long service accrual has already been added. After line 29, add:

```

long_service_accrual = get_number('LONG_SERVICE_ACCRUAL')

IF (years_service > 15) and long_service_accrual = 0 THEN
(
  long_service_accrual = 10
  e = set_number('LONG_SERVICE_ACCRUAL', long_service_accrual)
)
ELSE
(
  long_service_accrual = 0
)

```

Finally, add any long service accrual to the total accrual, ignoring the ceiling. Replace line 30 with:

```
E = set_number('TOTAL_ACCRUED_PTO', Total_Accrued_PTO +
```

```
Period_Accrued_PTO + long_service_accrual)
```

## Using Accrual Bands Based on Length of Service

The seeded looping formulas demonstrate how to use accrual bands based on length of service criteria (entered in the Accrual Bands window). To change the sample formulas to use accrual bands:

### In the top level formula:

- Remove lines 7 and 8
- Add the following after line 57:

```
E = set_number('ANNUAL_RATE',0)
E = set_number('UPPER_LIMIT',0)
E = set_number('CEILING',0)
```

### In the looping formula

- Remove lines 9, 10, 11, and 13
- Add the following after line 12:

```
Annual_Rate = get_number('ANNUAL_RATE')
Upper_Limit = get_number('UPPER_LIMIT')
Ceiling = get_number('CEILING')

Years_Service = Floor(Months_Between(Period_ED,
                                   Continuous_Service_Date)/12)

/*-----
If the Upper Limit was defaulted or years service
is greater than or equal to the upper limit of
the current band reset the globals to the
appropriate band.
-----*/

IF (Upper_Limit = 0 OR Years_Service >= Upper_Limit) THEN
(
  If (GET_ACCRUAL_BAND(Years_Service) = 0) THEN
  (
    Annual_Rate = get_number('ANNUAL_RATE')
    Upper_Limit = get_number('UPPER_LIMIT')
    Ceiling = get_number('CEILING')
  )
  ELSE /*function returned an error */
  (
    Continue_Processing_Flag = 'N'
    Return Continue_Processing_Flag
  )
)
```

)

```
Accrual_Rate = Annual_Rate / 12
```

### Changing the Length of Service Units

Normally the bands entered in the Accrual Bands window refer to years. However, you can change the formula to interpret the bands as another unit, such as six months. In the sample given above to change the looping formula to use accrual bands, you would simply replace '12' with '6' in the line:

```
Years_Service = Floor(Months_Between(Period_ED,  
Continuous_Service_Date)/12)
```

### Using Accrual Bands Based on Other Criteria

You can set up a user table to hold the values you require. For example:

	<b>Accrual</b>	<b>Ceiling</b>	<b>Max Carry Over</b>
Grade A	20	25	5
Grade B	24	25	8

You can also base your accrual bands on a combination of criteria (such as grade and length of service). In this case, you need to set up separate user tables for each value you want to hold (such as accrual amount, ceiling and maximum carry over). The table for accrual amount might look like this:

	<b>Grade A</b>	<b>Grade B</b>
0 to 5 years service	20	25
5 to 50 years service	24	28

To use data from a user table in a formula, use the GET\_TABLE\_VALUE function:

```
yearly_accrual = get_table_value(<table_name>, <column_name>,  
row_value>, <effective_date>)
```

The effective date parameter is optional. Example:

```
yearly_accrual = get_table_value ('MY_TABLE', 'GRADE A', 5)
```

---

## Sample Proration Formulas (UK Only)

The following sample formulas show how you can create payroll formulas to be used in different situations requiring proration.

Oracle Corporation may change or upgrade this formula in future releases of HRMS programs. This formula is strictly for example or prototype purposes only, and is not intended to provide a ready-made solution to be used in your environment. You can make a copy of this formula, subject to the terms of the license agreement for the programs, but you should not change your copied version for prototyping purposes. You should always write a new formula of your own to meet your particular requirements. This formula may contain certain hard-coded values to simplify the use of formulas for proration.

A typical example of proration would be when a new employee starts work in the middle of a monthly payroll period and your payroll department makes a pro-rata payment to reflect the proportion of monthly pay to which the employee is entitled.

The prototype formula supplied is only designed to work for payroll calculations involving proration and should not be used for any other purpose. Any use of the formula is subject to the terms of the Oracle license agreement for the HRMS programs and documentation.

### FastFormula to use if Employee is Paid using Salary Administration

```

/*****
Formula name : FF_PRORATION_SAL_MANAGEMENT
Formula to calculate salary in proration

DISCLAIMER: Oracle Corporation may change or upgrade this
FastFormula in the future releases. This FastFormula is strictly
for example or prototype purposes only. This FastFormula is not
intended to provide a ready-made solution to the users. This
formula may contain certain hard-coded values added to simplify
the concept of usage of FastFormulas in proration. Users should
make a copy of this formula and not change this formula. Users
need to write their own new formula(s) to meet requirements.
*****/
default for ASG_SALARY_BASIS_GRADE_ANNUALIZATION_FACTOR is 0
default for prorate_start is '01-JAN-1990' (date)
default for prorate_end is '01-JAN-1900' (date)
default for PAY_PROC_PERIOD_START_DATE IS '01-JAN-1950' (date)
default for PAY_PROC_PERIOD_END_DATE IS '01-JAN-1950' (date)
default for NI_NEW_TAX_YEAR IS '01-JAN-1950' (date)

default for annual_salary is 0.0
inputs are annual_salary (number),
```



```

        prorate_start (date),
        prorate_end (date)

/**
Prorate_start and prorate_end dates are passed from the payroll
engine. These dates basically represent the dates on which the
changes occur in the salary amount during the pay period.
annual_salary is an input value created in the element to which
this formula will be tied. Here the assumption is that the user
enters the annual salary amount through salary management.
**/

        l_amount=annual_salary
        l_string = ''

if(ASG_SALARY_BASIS_GRADE_ANNUALIZATION_FACTOR was not defaulted)
then
(
/**
This calculation makes sure that l_amount has the annual salary.
**/
        l_amount =
            amount*ASG_SALARY_BASIS_GRADE_ANNUALIZATION_FACTOR
    )

/**
l_tax_year_start_date derives the value from the DBI
NI_NEW_TAX_YEAR.
The following calculation sets the start date of the tax year to
01-APR-YYYY and end date of the tax year to 31-MAR-(YYYY+1)
**/

        l_tax_year_start_date = NI_NEW_TAX_YEAR
        l_tax_year_start_date = TRUNC(l_tax_year_start_date,'month')

        l_tax_year =
            TO_NUMBER(TO_CHAR(l_tax_year_start_date,'YYYY')) + 1
        l_tax_year_end_date =
            TO_DATE('31/03/'||TO_CHAR(l_tax_year),'DD/MM/YYYY')

/**
The following calculation is for the case when no proration occurs
during the pay period and element entry start date and element
entry end date is not equal to the pay period start and end dates
respectively.

**/
IF(prorate_start was defaulted) then
(

```

```

IF(PAY_PROC_PERIOD_END_DATE<>ENTRY_END_DATE) THEN
(
  IF(PAY_PROC_PERIOD_START_DATE<>ENTRY_START_DATE) THEN
  (
    /**
    Since prorate_start date is defaulted, it means no proration
    occurred, therefore we simply return the annual salary amount
    divided by 12.
    **/
    l_amount = ROUND(amount/12,2)
    result1 = l_amount
    return result1
  )
)
)

l_post_calc = 'N'

/**
l_post_calc is a flag used to do the calculation differently
depending on the values of this flag.
**/
/**
The following if condition takes care of the case when
Either
The element entry starts on the same day as pay period start date
OR
Prorate start date IS EQUAL TO element entry start date AND
Prorate start date IS NOT EQUAL TO one day prior to pay period
start date.

For example, in a monthly payroll running for the month of June
2000.

Either
Element entry start date is )01-JUN-2000 (employee starts on the
first day of the payroll period)

OR
Prorate start date is 14-JUN-2000 (employee starts in the middle
of the month or gets a salary change in the middle of the month)
Prorate start date IS NOT 31-MAY-2000
**/
if((prorate_start = ENRTY_START_DATE and
prorate_start <> ADD_DAYS(PAY_PROC_PERIOD_START_DATE,-1
)) or
ENTRY_START_DATE = PAY_PROC_PERIOD_START_DATE) then
(

```

```

/**
This code is executed when it is a new entry. Thus the post calc
must be performed.
**/
    l_post_calc = 'Y'
)
else
(
    if(prorate_end<>ENTRY_END_DATE) then
    (
        if(prorate_end = PAY_PROC_PERIOD_END_DATE) then
        )
    )
/**
This code is executed when it is a normal last proration entry.
Thus the post calc must be performed.
**/
    l_post_calc = 'Y'
)
)

/**
The following code sets the right proration start and end dates*/
**/
if(prorate_start was defaulted) then
(
/**
Prorate start is defaulted when no proration occurs
**/

    if(ENTRY_START_DATE=PAY_PROC_PERIOD_START_DATE) then
    (
/**
Element entry start date is the same as the payroll period start
date
**/
        prorate_start=ENTRY_START_DATE
        if(ENTRY_END_DATE=PAY_PROC_PERIOD_END_DATE) then
        (
            prorate_end=ENTRY_END_DATE
        )
        else
        (
            prorate_end=PAY_PROC_PERIOD_END_DATE
        )
    r)
    else if(ENTRY_END_DATE=PAY_PROC_PERIOD_END_DATE) then
    (
        prorate_end=ENTRY_END_DATE

```

```

        prorate_start=PAY_PROC_PERIOD_START_DATE
    )
)

/*Do we have to do daily calc*/
if((prorate_start>PAY_PROC_PERIOD_START_DATE
and prorate_end<PAY_PROC_PERIOD_END_DATE)
or
(prorate_start=PAY_PROC_PERIOD_START_DATE
and prorate_end<PAY_PROC_PERIOD_END_DATE
and prorate_start=ENTRY_START_DATE)) then
(
    l_post_calc='D'
)

/*Now perform the calculations*/
if(l_post_calc='N') then
(
/*Perform in the pre mode, i.e.start of year to current date*/

    l_days=days_between(prorate_end,
        l_tax_year_start_date) + 1
    l_amount1=l_amount * l_days/365
    l_months=TRUNC(months_between(PAY_PROC_PERIOD_START_DATE,
l_tax_year_start_date
        ))
    l_amount2=l_amount * l_months/12
    result1=l_amount1-l_amount2
)
else
)
    if(l_postcalc='Y') then
)
/*perform in post mode, i.e.current date to end of year*/

    l_days=days_between(l_tax_year_end_date,
        prorate_start) + 1
    l_amount1=l_amount*l_days/365
    l_months=TRUNC(months_between(l_tax_year_end_date,
        PAY_PROC_PERIOD_END_DATE
    ))
    l_amount2=l_amount*l_months/12
    result1=l_amount1-l_amount2
)
else
(
/**
This code is executed when l_post_calc=D. Perform in daily mode.
**/

```

```

    l_days=days_between(prorate_end,prorate-start) + 1
    l_amount1=0
    l_months=0
    l_amount2=0
    result1=l_amount*l_days/365
  )
)

/**
The following code is just to convert all the non-character
variables into the character variables, so that the values of the
variables could be displayed in the messages available either in
SOE form or messages.
**/
prorate_start_res=to_char(prorate_start,'DD-MON-YYYY')
prorate_end_res=to_char(prorate_end,'DD-MON-YYYY')
l_days_res=TO_CHAR(l_days)
l_amount1_res=TO_CHAR(l_amount1)
l_months_res=TO_CHAR(l_months)
l_amount2_res=TO_CHAR(l_amount2)

return result1,
       l_post_calc,
       l_days_res,
       l_amount1_res,
       l_months_res,
       l_amount2_res,
       prorate_start_res,
       prorate_end_res

```

## FastFormula to use if Employee is Paid using a Spinal Point/Pay Scale

```

/*****
Formula Name = UK_PRORATION_SPINAL_POINT
Formula Type = Oracle Payroll

```

Description: This formula is executed from within the payroll run by processing the element UK Salary Spinal Point. It calculates the rate of pay and returns this value. It uses the version of the function RATES\_HISTORY which requires a date input to be passed. This formula is just a prototype.

DISCLAIMER: Oracle Corporation may change or upgrade this fast formul in the future releases. This FastFormula is strictly for example or prototype purposes. This FastFormula is not intended to provide a ready-made solution to the users. This formula may contain certain hard-coded values added to simplify the concept of usage of FastFormulas in proration. Users should make a copy of

```

this formula and not change this formula. Users need to write
their own new formula(s) to meet requirements.
*****/
DEFAULT FOR date_worked IS '01-JAN-1950'(date)
DEFAULT FOR PAY_PROC_PERIOD_END_DATE IS '01-JAN-1950'(date)
DEFAULT FOR PAY_PROC_PERIOD_START_DATE IS '01-JAN-1950'(date)
DEFAULT FOR prorate_start IS '01-JAN-1950'(date)
DEFAULT FOR prorate_end IS '01-JAN-1950'(date)

inputs are prorate_start(date),
          prorate_end(date)

/**
Prorate_start and prorate_end dates are passed from the payroll
engine. These dates basically represent the dates on which the
changes occur in the pay scale/grade rate during the pay period.
**/
amount=0
message=''
IF(prorate_start WAS DEFAULTTED)then
(
/**
prorate_start date is defaulted when no proration occurs
**/
  l_date_worked=PAY_PROC_PERIOD_END_DATE
  prorate_start=PAY_PROC_PERIOD_START_DATE
  prorate_end=PAY_PROC_PERIOD_END_DATE
)
else
)
  l_date_worked=prorate_end
)
message1='Date defaulted to'||TO_CHAR(date_worked,'DD-MON-YYYY')

/**
The following function returns the value of the pay scale on the
l_date_worked which is last date of hte payroll period if no
proration occurs, otherwise it is the proration end date.
**/

if rates_history(l_date_worked,
  'UK Spinal Point',
  'E',
  'P',
  amount,
  message)= -1 then
(
  return message, message1
)

```

```

else
(
    l_days = days_between(prorate_end,prorate_start)+ 1

/**
l_days stores the number of days between prorate start and prorate
end dates
**/

message1 = 'The value
is' || TO_CHAR(ROUND(amount,2)) || for=' || TO_CHAR(l_days) || 'days from
' || TO_CHAR(prorate_start,'DD-MON-YYYY') || 'to' || TO_CHAR(prorate_end
,'DD-MON-YYYY')

/**
The message is just for informational purposes.
**/
    l_amount=amount*(12*l_days/365)

/**
The above calculation is hard-coded to simplify the calculation.
Please write your own calculation logic. The assumption was that
the value in the pay scale is monthly. Therefore to get annual
amount it is multiplied by 12. Then it is divided by 365 days to
get the amount for a single day. Once it is multiplied by l_days,
we get the amount for the days we want.
**/
/**
It is another assumption that a year contains 365 days. Please
change this assumption to take account of the leap year. This
formula will not work properly in a leap year.
**/
return l_amount, message1
)

```

## FastFormula to Use if Employee is Paid Using a Grade Rate

```

/*****
Formula Name = UK_PRORATION_GRADE_RATE
Formula Type = Oracle Payroll
Description: This formula is executed from within the payroll run
by processing the element UK Salary Grade Rate. It calculates the
rate of pay and returns this value. It uses the version of the
function RATES_HISTORY which requires a date input to be passed.
This formula is just a prototype.

```

DISCLAIMER: Oracle Corporation may change or upgrade this FastFormula in the future releases. This FastFormula is strictly for example or prototype purposes only. This FastFormula is not

intended to provide a ready-made solution to the users. This formula may contain certain hard-coded values added to simplify the concept of usage of FastFormulas in proration. Users should make a copy of this formula and not change this formula. Users need to write their own new formula(s) to meet requirements.

```

*****/
DEFAULT FOR date_worked IS '01-JAN-1950'(date)
DEFAULT FOR PAY_PROC_PERIOD-END_DATE IS '01-JAN-1950'(date)
DEFAULT FOR PAY_PROC_PERIOD-START-DATE IS '01-JAN-1950'(date)
DEFAULT FOR prorate_start IS '01-JAN-1950'(date)
DEFAULT FOR prorate_end IS '01-JAN-1950'(date)

```

```

inputs are prorate_start(date),
         prorate_end(date)

```

```

/**
Prorate_start and prorate_end dates are passed from the payroll
engine. These dates basically represent the dates on which the
changes occur in the pay scale/grade rate during the pay period.
**/

```

```

amount = 0
message = ''

```

```

IF(prorate_start WAS DEFAULTED)then
(
/**
prorate_start date is defaulted when no proration occurs
**/
  l_date_worked=PAY_PROC_PERIOD_END_DATE
  prorate_start=PAY_PROC_PERIOD_START_DATE
  prorate_end=PAY_PROC_PERIOD_END_DATE
)
else
(
  l_date_worked=prorate_end
)
message1='Date defaulted to'||TO_CHAR(date_worked,'DD-MON-YYYY')

```

```

/**
The following function returns the value of the pay scale on the
l_date_worked which is last date of the payroll period if no
proration occurs, otherwise it is the proration end date.
**/

```

```

if rates_history(l_date_worked,
  'UK Grade Rate',
  'E',

```



```

        'H',
        amount,
        message) = -1 then
    (
        return message, message1
    )
else
    (
        l_days = days_between(prorate_end,prorate_start) + 1
    /**
    l_days stores the number f days between prorate start and prorate
    end dates
    **/
    message1 = 'The value
    is' || TO_CHAR(ROUND(amount,2)) || for=' ' || TO_CHAR(l_days) || 'days from
    ' || TO_CHAR(prorate_start, 'DD-MON-YYYY') || 'to' || TO_CHAR(prorate_end
    , 'DD-MON-YYYY')
    /**
    The message is just for informational purposes.
    **/

        l_amount=amount * ((l_days*1600)/365)
    /**
    The above calculation is hard-coded to simplify the calculation.
    Please write your own calculatioin logic. The assumption was that
    the value in the grade rate is hourly. Therefore to get the annual
    amount it is multiplied by 16000 hours (assuming that there are
    1600 hours in a year). Then it is divided by 365 days to get the
    amount for a single day. Once it is multiplied by l_days, we get
    the amount for the days we want.
    **/
    /**
    It is another assumption that a year contains 365 days. Please
    change this asumption to take account of leap year. This formula
    will not work properly in a leap year.
    **/
        return l-amount,message1
    )

```

## FastFormula to use for a Deduction

```

/*****
Formula name: UK_PRORATION_DEDUCTIONS
Purpose: Formula prototype to calculate prorated deductions.

DISCLAIMER: Oracle Corporation may change or upgrade this
FastFormula in future releases. This FastFormula is strictly for
example or prototype purposes only. This FastFormula is ot
intended to provide a ready-made solution for the user. This

```

```

formula may contain certain hard-coded values added to simplify
the concept of usage of FastFormulas in proration. Users should
make a copy of this formula and not change this formula. Users
need to write their own new formula(s) to meet requirements.
*****/
default for prorate_start is '01-JAN-1900'(date)
default for prorate_end is '01-JAN-1900'(date)

default for amount is 0.0
inputs are
    prorate_start(date),
    prorate_end(date),
    annual_deduction(number)
/**
Prorate-start and prorate_end dates are passed from the payroll
engine. These dates basically represent the dates on which the
changes occur in the deduction amount during the pay period.
annual_deduction is an input value created in the element to which
this formula will be tied. Here the assumption is that the user
enters the annual deduction amount in the input value.
**/

l_amount=annual_deduction
message='Proration Start Date' || TO_CHAR(prorate_start,
'DD-MON-YYYY')
IF(prorate_start was defaulted)then
(
/**
prorate_start date is defaulted when no proration occurs.
Therefore we should just return the annual deduction amount
divided by 12.
**/
    l_amount=l_amount/12
    return l_amount
)
else
(
    l_days=days_between(prorate_end, prorate_start) + 1
/**
l_days stores the number of days between prorate start and prorate
end dates
**/
    l_days_in_fiscal_year=365

/**
It is another assumption that a year contains 365 days. Please
change this assumption to take care of leap year. This formula
will not work properly in a leap year.
**/

```

```
    l_amount=(l_amount*l_days)/l_days_in_fiscal_year
/**
In the above calculation, since l_amount contains the annual
deduction amount, it is divided by 365 days of the year to get the
deduction amount per day. This amount is then multiplied by the
number of days in question to get the proper deduction amount
**/

return l_amount, message
)
```

---

## Sample Rates History Formulas (UK Only)

The following sample formulas show how you can create payroll formulas to be used in different situations requiring historic rates.

Oracle Corporation may change this formula in future releases. This formula is strictly for example or prototype uses only, and is not intended to provide a ready-made solution. You can make a copy of this formula but you should not change your copied version. Always write a formula of your own to meet your own requirements. This formula may contain certain hard-coded values to simplify the use of formulas for proration.

### FastFormula to Calculate the Salary when Attached to a Grade Rate

```

/*****
Formula name : Grade_Rate_Formula
Description : Formula to calculate the salary when attached to a
Grade Rate
*****/

inputs are date_worked(date)

amount=0
message=''

if rates_history (date_worked,
                  'Salary Element GR',
                  'E',
                  'P',
                  amount,
                  message)=-1 then
(
return message
)
else
(
return amount
)
.

```

### FastFormula to Calculate the Salary when Attached to a Spinal Point

```

/*****
Formula name : Spinal_Point_Formula
Description : Formula to calculate the salary when attached to a
Spinal Point
*****/

```

```

inputs are date_worked(date)

amount=0
message=''

if rates_history (date_worked,
                  'Salary Element SP',
                  'E',
                  'P',
                  amount,
                  message)=-1 then
(
return message
)
else
(
return amount
)
.

```

## FastFormula to Calculate the Salary when Attached to a Global Value

```

/*****
Formula name : Global_Value_Formula
Description : Formula to calculate the salary when attached to a
Global Value
*****/

inputs are date_worked(date)

amount=0
message=''

if rates_history (date_worked,
                  'Salary Element GV',
                  'E',
                  'P',
                  amount,
                  message)=-1 then
(
return message
)
else
(
return amount
)
.

```

## FastFormula to Calculate the Salary when there is an Input Value

```

/*****
Formula name : Input_Value_Formula
Description : Formula to calculate the salary when there is an
input value
*****/

inputs are date_worked(date)

amount=0
message=''

if rates_history (date_worked,
                  'Salary Element IV',
                  'E',
                  'P',
                  amount,
                  message)=-1 then
(
return message
)
else
(
return amount
)
.

```

## FastFormula to Calculate the Overtime when there is a Rate Type

```

/*****
Formula name : Overtime_Formula
Description : Formula to calculate the overtime when there is an
rate type
*****/

inputs are date_worked(date)

amount=0
message=''

if rates_history (date_worked,
                  'Overtime Element',
                  'R',
                  'P',
                  amount,
                  message)=-1 then
(
return message
)
.

```

```
)  
else  
(  
return amount  
)  
.
```





# Glossary

**360 Degree Appraisal** Part of the SSHR Appraisal function and also known as a Group Appraisal. This is an employee appraisal undertaken by managers with participation by reviewers.

**360 Degree Self Appraisal** Part of the SSHR Appraisal function and also known as a Group Appraisal. This is a 360 Degree appraisal initiated by an employee. The employee (initiator) can add managers and reviewers to the appraisal.

## A

**Absence Types** Categories of absence, such as medical leave or vacation leave, that you define for use in absence windows.

**Accrual** The recognized amount of leave credited to an employee which is accumulated for a particular period.

**Accrual Band** A range of values that determines how much paid time off an employee accrues. The values may be years of service, grades, hours worked, or any other factor.

**Accrual Plan** See: *PTO Accrual Plan*

**Accrual Period** The unit of time, within an accrual term, in which PTO is accrued. In many plans, the same amount of time is accrued in each accrual period, such as two days per month. In other plans, the amount accrued varies from period to period, or the entitlement for the full accrual term is given as an up front amount at the beginning of the accrual term.

**Accrual Term** The period, such as one year, for which accruals are calculated. In most accrual plans, unused PTO accruals must be carried over or lost at the end of the accrual term. Other plans have a rolling accrual term which is of a certain duration but has no fixed start and end dates.

**Activity Rate** The monetary amount or percentage associated with an activity, such as \$12.35 per pay period as an employee payroll contribution for medical coverage. Activity rates can apply to participation, eligibility, coverages, contributions, and distributions.

**Actual Premium** The per-participant premium an insurance carrier charges the plan sponsor for a given benefit.

**Administrative Enrollment** A type of scheduled enrollment caused by a change in plan terms or conditions and resulting in a re-enrollment.

**Applicant** A candidate for employment in a Business Group.

**Appraisee** A person being appraised by an appraiser..

**Appraiser** A person, usually a manager, who appraises an employee.

**Appraisal** An appraisal is a process where an employee's work performance is rated and future objectives set. See also: *Assessment*.

**Appraising Manager** The person who initiates and performs an Employee-Manager or 360 Degree Appraisal. An appraising manager can create appraisal objectives.

**Apply for a Job** An SSHR function that enables an employee to, apply, search and prepare applications for an internally advertised vacancy.

**Arrestment** Scottish court order made out for unpaid debts or maintenance payments. See also: *Court Order*

**Assessment** An information gathering exercise, from one or many sources, to evaluate a person's ability to do a job. See also: *Appraisal*.

**Assignment** An employee's assignment identifies his or her role and payroll within a Business Group. The assignment is made up of a number of assignment components. Of these, organization is mandatory, and payroll is a required component for payment purposes.

**Assignment Number** A number that uniquely identifies an employee's assignment. An employee with multiple assignments has multiple assignment numbers.

**Assignment Set** A grouping of employees and/or applicants that you define for running QuickPaint reports and processing payrolls. See also: *QuickPaint Report*

**Assignment Status** For employees, used to track their permanent or temporary departures from your enterprise, and to control the remuneration they receive. For applicants, used to track the progress of their applications.

## B

**BACS** Banks Automated Clearing System. This is the UK system for making direct deposit payments to employees.

**Balances** Positive or negative accumulations of values over periods of time normally generated by payroll runs. A balance can sum pay values, time periods or numbers. See also: *Predefined Components*

**Balance Adjustment** A correction you make to a balance. You can adjust user balances and assignment level predefined balances only.

**Balance Dimension** The period for which a balance sums its balance feeds, or the set of assignments/transactions for which it sums them. There are five time dimensions: Run, Period, Quarter, Year and User. You can choose any reset point for user balances.

**Balance Feeds** These are the input values of matching units of measure of any elements defined to feed the balance.

**Bargaining Unit** A bargaining unit is a legally organized group of people which have the right to negotiate on all aspects of terms and conditions with employers or employer federations. A bargaining unit is generally a trade union or a branch of a trade union.

**Base Currency** The currency in which Oracle Payroll performs all payroll calculations for your Business Group. If you pay employees in different currencies to this, Oracle Payroll calculates the amounts based on exchange rates defined in the system.

**Behavioral Indicators** Characteristics that identify how a competence is exhibited in the work context. See also: *Proficiency Level*

**Benefit** Any part of an employee's remuneration package that is not pay. Vacation time, employer-paid medical insurance and stock options are all examples of benefits. See also: *Elements*

**Beneficiary** A person or organization designated to receive the benefits from a benefit plan upon the death of the insured.

**Block** The largest subordinate unit of a window, containing information for a specific business function or entity. Every window consists of at least one block. Blocks contain fields and, optionally, regions. They are delineated by a bevelled edge. You must save your entries in one block before navigating to the next. See also: *Region, Field*

**Budget Value** In Oracle Human Resources you can enter staffing budget values and actual values for each assignment to measure variances between actual and planned staffing levels in an organization or hierarchy.

**Business Group** The highest level organization in the Oracle HRMS system. A Business Group may correspond to the whole of your enterprise or to a major grouping such as a subsidiary or operating division. Each Business Group must correspond to a separate implementation of Oracle HRMS.

**Business Number (BN)** In Canada, this is the employer's account number with Revenue Canada. Consisting of 15 digits, the first 9 identify the employer, the next 2 identify the type of tax account involved (payroll vs. corporate tax), and the last 4 identify the particular account for that tax.

## C

**Cafeteria Benefits Plan** See: Flexible Benefits Program

**Calendars** In Oracle Human Resources you define calendars that determine the start and end dates for budgetary years, quarters and periods. For each calendar you select a basic period type. In Oracle SSP/SMP you define calendars to determine the start date and time for SSP qualifying patterns.

**Calendar Exceptions** In Oracle SSP/SMP you define calendar exceptions for an SSP qualifying pattern, to override the pattern on given days. Each calendar exception is another pattern which overrides the usual pattern.

**Canada/Quebec Pension Plan (CPP/QPP)**

**Contributions** Contributions paid by employers and employees to each of these plans provide income benefits upon retirement.

**Candidate Offers** An SSHR function used by a line manager to offer a job to a candidate. This function is supplied with its own responsibility.

**Career Path** This shows a possible progression from one job or position from any number of other jobs or positions within the Business Group. A career path must be based on either job progression or position progression; you cannot mix the two.

**Carry Over** The amount of unused paid time off entitlement an employee brings forward from one accrual term to the next. It may be subject to an expiry date i.e. a date by which it must be used or lost. See also: *Residual*

**Cash Analysis** A specification of the different currency denominations required for paying your employees in cash. Union contracts may require you to follow certain cash analysis rules.

**Certification** Documentation required to enroll or change elections in a benefits plan as the result of a life event, to waive participation in a plan, to designate dependents for coverage, or to receive reimbursement for goods or services under an FSA.

**Ceiling** The maximum amount of unused paid time off an employee can have in an accrual plan. When an employee reaches this maximum, he or she must use some accrued time before any more time will accrue.

**Child/Family Support payments** In Canada, these are payments withheld from an employee's compensation to satisfy a child or family support order from a Provincial Court. The employer is responsible for withholding and remitting the payments to the court named in the order.

**Collective Agreement** A collective agreement is a form of contract between an employer or employer representative, for example, an employer federation, and a bargaining unit for example, a union or a union branch.

**Collective Agreement Grade** Combination of information that allows you to determine how an employee is ranked or graded in a collective agreement.

**Communications** Benefits plan information that is presented in some form to participants. Examples include a pre-enrollment package, an enrollment confirmation statement, or a notice of default enrollment.

**Compensation** The pay you give to employees, including wages or salary, and bonuses. See also: *Elements*

**Compensation Object** For Standard and Advanced Benefits, compensation objects define, categorize, and help to manage the benefit plans that are offered to eligible participants. Compensation objects include programs, plan types, plans, options, and combinations of these entities.

**Competence** Any measurable behavior required by an organization, job or position that a person may demonstrate in the work context. A competence can be a piece of knowledge, a skill, an attitude or an attribute.

**Competence Evaluation** A method used to measure an employees ability to do a defined job.

**Competence Profile** Where you record applicant and employee accomplishments, for example, proficiency in a competence.

**Competence Requirements** Competencies required by an organization, job or position. See also: *Competence, Core Competencies*

**Competence Type** A group of related competencies.

**Configurable Forms** Forms that your system administrator can modify for ease of use or security purposes by means of Custom Form restrictions. The Form Customization window lists the forms and their methods of configuration.

**Consolidation Set** A grouping of payroll runs within the same time period for which you can schedule reporting, costing, and post-run processing.

**Contact** A person who has a relationship to an employee that you want to record. Contacts can be dependents, relatives, partners or persons to contact in an emergency.

**Contract** A contract of employment is an agreement between an employer and employee or potential employee that defines the fundamental legal relationship between an employing organization and a person who offers his or her services for hire. The employment contract defines the terms and conditions to which both parties agree and those that are covered by local laws.

**Contribution** An employer's or employee's monetary or other contribution to a benefits plan.

**Core Competencies** Also known as *Leadership Competencies* or *Management Competencies*. The competencies required by every person to enable the enterprise to meet its goals. See also: *Competence*

**Costable Type** A feature that determines the processing an element receives for accounting and costing purposes. There are four costable types in Oracle HRMS: costed, distributed costing, fixed costing, and not costed.

**Costing** Recording the costs of an assignment for accounting or reporting purposes. Using Oracle Payroll, you can calculate and transfer costing information to your general ledger and into systems for project management or labor distribution.

**Court Order** A ruling from a court that requires an employer to make deductions from an employee's salary for maintenance payments or debts, and to pay the sums deducted to a court or local authority. See also: *Arrestment*

## D

**Database Item** An item of information in Oracle HRMS that has special programming attached, enabling Oracle FastFormula to locate and retrieve it for use in formulas.

**Date To and Date From** These fields are used in windows not subject to DateTrack. The period you enter in these fields remains fixed until you change the values in either field. See also: *DateTrack*, *Effective Date*

**DateTrack** When you change your effective date (either to past or future), DateTrack enables you to enter information that takes effect on your new effective date, and to review information as of the new date. See also: *Effective Date*

**Dependent** In a benefit plan, a person with a proven relationship to the primary participant whom the participant designates to receive coverage based on the terms of the plan.

**Deployment Factors** See: *Work Choices*

**Derived Factor** A factor (such as age, percent of fulltime employment, length of service, compensation level, or the number of hours worked per period) that is used in calculations to determine Participation Eligibility or Activity Rates for one or more benefits.

**Descriptive Flexfield** A field that your organization can configure to capture additional information required by your business but not otherwise tracked by Oracle Applications. See also: *Key Flexfield*

**Developer Descriptive Flexfield** A flexfield defined by your localization team to meet the specific legislative and reporting needs of your country. See also: *Extra Information Types*

**Direct Deposit** The electronic transfer of an employee's net pay directly into the account(s) designated by the employee.

**Distribution** Monetary payments made from, or hours off from work as allowed by, a compensation or benefits plan.

## E

**Effective Date** The date for which you are entering and viewing information. You set your effective date in the Alter Effective Date window. See also: *DateTrack*

**EIT** See: *Extra Information Type*

**Electability** The process which determines whether a potential benefits participant, who has satisfied the eligibility rules governing a program, plan, or option in a plan, is able to elect benefits. Participants who are *eligible* for benefits do not always have *electable* benefit choices based on the rules established in a benefit plan design.

**Elements** Components in the calculation of employee pay. Each element represents a compensation or benefit type, such as salary, wages, stock purchase plans, and pension contributions.

**Element Classifications** These control the order in which elements are processed and the balances they feed. Primary element classifications and some secondary classifications are predefined by Oracle Payroll. Other secondary classifications can be created by users.

**Element Entry** The record controlling an employee's receipt of an element, including the period of time for which the employee receives the element and its value. See also: *Recurring Elements, Nonrecurring Elements*

**Element Link** The association of an element to one or more components of an employee assignment. The link establishes employee eligibility for that element. Employees whose assignment components match the components of the link are eligible for the element. See also: *Standard Link*

**Element Set** A group of elements that you define to process in a payroll run, or to control access to compensation information from a configured form, or for distributing costs.

**Eligibility** The process by which a potential benefits participant satisfies the rules governing whether a person can ever enroll in a program, plan, or option in a plan. A participant who is *eligible* for benefits must also satisfy *electability* requirements.

**Employee Histories** An SSHR function for an employee to view their, Training History, Job Application History, Employment History, Absence History, or Salary History. A manager can also use this function to view information on their direct reports.

**Employment Category** A component of the employee assignment. Four categories are defined: Full Time – Regular, Full Time – Temporary, Part Time – Regular, and Part Time – Temporary.

**Employment Insurance (EI)** Benefit plan run by the federal government to which the majority of Canadian employers and employees must contribute.

**Employment Insurance Rate** In Canada, this is the rate at which the employer contributes to the EI fund. The rate is expressed as a percentage of the employee's contribution. If the employer maintains an approved wage loss replacement program, they can reduce their share of EI premiums by obtaining a reduced contribution rate. Employers would remit payroll deductions under a different employer account number for employees covered by the plan.

**Employment Equity Occupational Groups (EEOG)** In Canada, the Employment Equity Occupational Groups (EEOG) consist of 14 classifications of work used in the Employment Equity Report. The EEOGs were derived from the National Occupational Classification system.

**Enroll in a Class** An SSHR function which enables an employee to search and enroll in an internally published class. An employee can also use this function to maintain their competencies.

**Enrollment Action Type** Any action required to complete enrollment or de-enrollment in a benefit.

**Entitlement** In Australia, this is all unused leave from the previous year that remains to the credit of the employee.

**ESS** Employee Self Service. A predefined SSHR responsibility.

**Event** An activity such as a training day, review, or meeting, for employees or applicants.

**Expected Week of Confinement (EWC)** In the UK, this is the week in which an employee's baby is due. The Sunday of the expected week of confinement is used in the calculations for Statutory Maternity Pay (SMP).

**Extra Information Type (EIT)** A type of developer descriptive flexfield that enables you to create an unlimited number of information types for six key areas in Oracle HRMS. Localization teams may also predefine some EITs to meet the specific legislative requirements of your country. See also: *Developer Descriptive Flexfield*

## F

**Field** A view or entry area in a window where you enter, view, update, or delete information. See also: *Block, Region*

**Flex Credit** A unit of "purchasing power" in a flexible benefits program. An employee uses flex credits, typically expressed in monetary terms, to "purchase" benefits plans and/or levels of coverage within these plans.

**Flexible Benefits Program** A benefits program that offers employees choices among benefits plans and/or levels of coverage. Typically, employees are given a certain amount of flex credits or moneys with which to "purchase" these benefits plans and/or coverage levels.

**Flexible Spending Account (FSA)** Under US Internal Revenue Code Section 125, employees can set aside money on a pretax basis to pay for eligible unreimbursed health and dependent care expenses. Annual monetary limits and use-it-or-lose-it provisions exist. Accounts are subject to annual maximums and forfeiture rules.

**Form** A predefined grouping of functions, called from a menu and displayed, if necessary, on several windows. Forms have blocks, regions and fields as their components. See also: *Block, Region, Field*

## G

**Global Value** A value you define for any formula to use. Global values can be dates, numbers or text.

**Goods or Service Type** A list of goods or services a benefit plan sponsor has approved for reimbursement.

**Grade** A component of an employee's assignment that defines their level and can be used to control the value of their salary and other compensation elements.

**Grade Comparatio** A comparison of the amount of compensation an employee receives with the mid-point of the valid values defined for his or her grade.

**Grade Rate** A value or range of values defined as valid for a given grade. Used for validating employee compensation entries.

**Grade Scale** A sequence of steps valid for a grade, where each step corresponds to one point on a pay scale. You can place each employee on a point of their grade scale and automatically increment all placements each year, or as required. See also: *Pay Scale*

**Grade Step** An increment on a grade scale. Each grade step corresponds to one point on a pay scale. See also: *Grade Scale*

**Grandfathered** A term used in Benefits Administration. A person's benefits are said to be grandfathered when a plan changes but they retain the benefits accrued.

**Group** A component that you define, using the People Group key flexfield, to assign employees to special groups such as pension plans or unions. You can use groups to determine employees' eligibility for certain elements, and to regulate access to payrolls.

**Group Certificate** In Australia, this is a statement from a legal employer showing employment income of an employee for the financial year..

## H

**Hierarchy** An organization or position structure showing reporting lines or other relationships. You can use hierarchies for reporting and for controlling access to Oracle HRMS information.

## I



**Imputed Income** Certain forms of indirect compensation that US Internal Revenue Service Section 79 defines as fringe benefits and taxes the recipient accordingly. Examples include employer payment of group term life insurance premiums over a certain monetary amount, personal use of a company car, and other non-cash awards.

**Initiator** In SSHR a person who starts a 360 Degree appraisal (Employee or Self) on an individual. An initiator and the appraisee are the only people who can see all appraisal information.

**Input Values** Values you define to hold information about elements. In Oracle Payroll, input values are processed by formulas to calculate the element's run result. You can define up to fifteen input values for an element.

**Instructions** An SSHR user assistance component displayed on a web page to describe page functionality.

## K

**Key Flexfield** A flexible data field made up of segments. Each segment has a name you define and a set of valid values you specify. Used as the key to uniquely identify an entity, such as jobs, positions, grades, cost codes, and employee groups. See also: *Descriptive Flexfield*

## L

**Leave Loading** In Australia, an additional percentage amount of the annual leave paid that is paid to the employee.

**Leaver's Statement** In the UK, this Records details of Statutory Sick Pay (SSP) paid during a previous employment (issued as form SSP1L) which is used to calculate a new employee's entitlement to SSP. If a new employee falls sick, and the last date that SSP was paid for under the previous employment is less than eight calendar weeks before the first day of the PIW for the current sickness, the maximum liability for SSP is reduced by the number of weeks of SSP shown on the statement.

**Legal Employer** A business in Australia that employs people and has registered with the Australian Tax Office as a Group Employer.

**Life Event** A significant change in a person's life that results in a change in eligibility or ineligibility for a benefit.

**Life Event Collision** A situation in which the impacts from multiple life events on participation eligibility, enrollability, level of coverage or activity rates conflict with each other.

**Life Event Enrollment** A benefits plan enrollment that is prompted by a life event occurring at any time during the plan year.

**Linking Interval** In the UK, this is the number of days that separate two periods of incapacity for work. If a period of incapacity for work (PIW) is separated from a previous PIW by less than the linking interval, they are treated as one PIW according to the legislation for entitlement to Statutory Sick Pay (SSP). An employee can only receive SSP for the maximum number of weeks defined in the legislation for one PIW.

**Linked PIWs** In the UK, these are linked periods of incapacity for work that are treated as one to calculate an employee's entitlement to Statutory Sick Pay (SSP). A period of incapacity for work (PIW) links to an earlier PIW if it is separated by less than the linking interval. A linked PIW can be up to three years long.

**LMSS** Line Manager Self Service. A predefined SSHR responsibility.

**Long Service Leave** Leave with pay granted to employees of a particular employer after a prescribed period of service or employment with that employer.

**Lookup Types** Categories of information, such as nationality, address type and tax type, that have a limited list of valid values. You can define your own Lookup Types, and you can add values to some predefined Lookup Types.

**Lower Earnings Limit (LEL)** In the UK, this is the minimum average weekly amount an employee must earn to pay National Insurance contributions. Employees who do not earn enough to pay National Insurance cannot receive Statutory Sick Pay (SSP) or Statutory Maternity Pay (SMP).

## M

**Manager–Employee Appraisal** Part of the SSHR Appraisal function. A manager appraisal of an employee. However, an appraising manager does not have to be a manager.

**Maternity Pay Period** In the UK, this is the period for which Statutory Maternity Pay (SMP) is paid. It may start at any time from the start of the 11th week before the expected week of confinement and can continue for up to 18 weeks. The start date is usually agreed with the employee, but can start at any time up to the birth. An employee is not eligible to SMP for any week in which she works or for any other reason for ineligibility, defined by the legislation for SMP.

**Menus** You set up your own navigation menus, to suit the needs of different users.

**Medicare Levy** An amount payable by most taxpayers in Australia to cover some of the cost of the public health system.

## N

**NACHA** National Automated Clearing House Association. This is the US system for making direct deposit payments to employees.

**National Identifier** This is the alphanumeric code that is used to uniquely identify a person within their country. It is often used for taxation purposes. For example, in the US it is the Social Security Number, in Italy it is the Fiscal Code, and in New Zealand it is the IRD Number.

**National Occupational Classification (NOC) code** In Canada, the National Occupational Classification (NOC) System was developed to best reflect the type of work performed by employees. Occupations are grouped in terms of particular tasks, duties and responsibilities. The use of this standardized system ensures consistency of data from year to year within the same company as well as between companies. These codes are used in the Employment Equity Report.

**Net Accrual Calculation** The rule that defines which element entries add to or subtract from a plan's accrual amount to give net entitlement.

**Net Entitlement** The amount of unused paid time off an employee has available in an accrual plan at any given point in time.

**Nonrecurring Elements** Elements that process for one payroll period only unless you make a new entry for an employee. See also: *Recurring Elements*

**North American Industrial Classification (NAIC) code** The North American Industrial Classification system (NAICs) was developed jointly by the US, Canada and Mexico to provide comparability in statistics regarding business activity across North America. The NAIC replaces the US Standard Industrial Classification (SIC) system, and is used in the Employment Equity Report.

**Not in Program Plan** A benefit plan that you define outside of a program.

## O

**Open Enrollment** A type of scheduled enrollment in which participants can enroll in or alter elections in one or more benefits plans.

**Oracle FastFormula** An Oracle tool that allows you to write Oracle HRMS formulas without using a programming language.

**Organization** A required component of employee assignments. You can define as many organizations as you want within your Business Group. Organizations can be internal, such as departments, or external, such as recruitment agencies. You can structure your organizations into organizational hierarchies for reporting purposes and for system access control.

**OSSWA** Oracle Self Service Web Applications.

**OTM** Oracle Training Management.

## P

**Pattern** A pattern comprises a sequence of time units that are repeated at a specified frequency. Oracle SSP/SMP uses SSP qualifying patterns to determine employees entitlement to Statutory Sick Pay (SSP).

**Pattern Time Units** A sequence of time units specifies a repeating pattern. Each time unit specifies a time period of hours, days or weeks.

**Pay Scale** A set of progression points that can be related to one or more rates of pay. Employee's are placed on a particular point on the scale according to their grade and, usually, work experience. See also: *Grade Scale*

**Payment Type** There are three standard payment types for paying employees: check, cash and direct deposit. You can define your own payment methods corresponding to these types.

**Payroll** A group of employees that Oracle Payroll processes together with the same processing frequency, for example, weekly, monthly or bimonthly. Within a Business Group, you can set up as many payrolls as you need.

**People List** An SSHR line manager utility used to locate an employee.

**Performance (within Assessment)** An expectation of "normal" performance of a competence over a given period. For example, a person may exceed performance expectation in the communication competence. See also: *Proficiency (within Assessment)*, *Competence*, *Assessment*

**Period of Incapacity for Work (PIW)** In the UK, this is a period of sickness that lasts four or more days in a row, and is the minimum amount of sickness for which Statutory Sick Pay can be paid. If a PIW is separated by less than the linking interval, a linked PIW is formed and the two PIWs are treated as one.

**Period Type** A time division in a budgetary calendar, such as week, month, or quarter.

**Person Search** An SSHR function which enables a manager to search for a person. There are two types of search, Simple and Advanced.

**Person Type** There are eight system person types in Oracle HRMS. Seven of these are combinations of employees, ex-employees, applicants, and ex-applicants. The eighth category is 'External'. You can create your own user person types based on the eight system types.

**Personal Public Service Number (PPS)** The Irish equivalent to National Insurance number in the UK, or the Social Security number in the US.

**Personal Tax Credits Return (TD1)** A Revenue Canada form which each employee must complete. Used by the employee to reduce his or her taxable income at source by claiming eligible credits and also provides payroll with such important information as current address, birth date, and SIN. These credits determine the amount to withhold from the employee's wages for federal/provincial taxes.

**Plan Design** The functional area that allows you to set up your benefits programs and plans. This process involves defining the rules which govern eligibility, available options, pricing, plan years, third party administrators, tax impacts, plan assets, distribution options, required reporting, and communications.

**Plan Sponsor** The legal entity or business responsible for funding and administering a benefits plan. Generally synonymous with employer.

**Position** A specific role within the Business Group derived from an organization and a job. For example, you may have a position of Shipping Clerk associated with the organization Shipping and the job Clerk.

**Predefined Components** Some elements and balances, all primary element classifications and some secondary classifications are defined by Oracle Payroll to meet legislative requirements, and are supplied to users with the product. You cannot delete these predefined components.

**Professional Information** An SSHR function which allows an employee to maintain their own professional details or a line manager to maintain their direct reports professional details.

**Proficiency (within Assessment)** The perceived level of expertise of a person in a competence, in the opinion of the assessor, over a given period. For example, a person may demonstrate the communication competence at Expert level. See also: *Performance (within Assessment)*, *Competence*, *Assessment*

**Proficiency Level** A system for expressing and measuring how a competence is exhibited in the work context. See also: *Behavioral Indicators*.

**Progression Point** A pay scale is calibrated in progression points, which form a sequence for the progression of employees up the pay scale. See also: *Pay Scale*

**Provincial/Territorial Employment Standards Acts** In Canada, these are laws covering minimum wages, hours of work, overtime, child labour, maternity, vacation, public/general holidays, parental and adoption leave, etc., for employees regulated by provincial/territorial legislation.

**Provincial Health Number** In Canada, this is the account number of the provincially administered health care plan that the employer would use to make remittances. There would be a unique number for each of the provincially controlled plans i.e. EHT, Quebec HSF, etc.

**PTO Accrual Plan** A benefit in which employees enroll to entitle them to accrue and take paid time off. The purpose of absences allowed under the plan, who can enroll, how much time accrues, when the time must be used, and other rules are defined for the plan.

**Qualification Type** An identified qualification method of achieving proficiency in a competence, such as an award, educational qualification, a license or a test. See also: *Competence*

**Qualifying Days** In the UK, these are days on which Statutory Sick Pay (SSP) can be paid, and the only days that count as waiting days. Qualifying days are normally work days, but other days may be agreed.

**Qualifying Pattern** See: *SSP Qualifying Pattern*

**Qualifying Week** In the UK, this is the week during pregnancy that is used as the basis for the qualifying rules for Statutory Maternity Pay (SMP). The date of the qualifying week is fifteen weeks before the expected week of confinement and an employee must have been continuously employed for at least 26 weeks continuing into the qualifying week to be entitled to SMP.

**Quebec Business Number** In Canada, this is the employer's account number with the Ministère du Revenu du Québec, also known as the Quebec Identification number. It consists of 15 digits, the first 9 identify the employer, the next 2 identify the type of tax account involved (payroll vs. corporate tax), and the last 4 identify the particular account for that tax.

**Questionnaire** An SSHR function which records the results of an appraisal.

**QuickPaint Report** A method of reporting on employee and applicant assignment information. You can select items of information, paint them on a report layout, add explanatory text, and save the report definition to run whenever you want. See also: *Assignment Set*

## Q

**QPP** (See Canada/Quebec Pension Plan)

## R

**Rates** A set of values for employee grades or progression points. For example, you can define salary rates and overtime rates.

**Rating Scale** Used to describe an enterprise's competencies in a general way. You do not hold the proficiency level at the competence level. See also: *Proficiency Level*

**Record of Employment (ROE)** A Human Resources Development Canada form that must be completed by an employer whenever an interruption of earnings occurs for any employee. This form is necessary to claim Employment Insurance benefits.

**Recruitment Activity** An event or program to attract applications for employment. Newspaper advertisements, career fairs and recruitment evenings are all examples of recruitment activities. You can group several recruitment activities together within an overall activity.

**Recurring Elements** Elements that process regularly at a predefined frequency. Recurring element entries exist from the time you create them until you delete them, or the employee ceases to be eligible for the element. Recurring elements can have standard links. See also: *Nonrecurring Elements, Standard Link*

**Region** A collection of logically related fields in a window, set apart from other fields by a rectangular box or a horizontal line across the window. See also: *Block, Field*

**Registered Pension Plan (RPP)** This is a pension plan that has been registered with Revenue Canada. It is a plan where funds are set aside by an employer, an employee, or both to provide a pension to employees when they retire. Employee contributions are generally exempt from tax.

**Registered Retirement Savings Plan (RRSP)** This is an individual retirement savings plan that has been registered with Revenue Canada. Usually, contributions to the RRSP, and any income earned within the RRSP, is exempt from tax.

**Report Parameters** Inputs you make when submitting a report to control the sorting, formatting, selection, and summarizing of information in the report.

**Report Set** A group of reports and concurrent processes that you specify to run together.

**Requisition** The statement of a requirement for a vacancy or group of vacancies.

**Request Groups** A list of reports and processes that can be submitted by holders of a particular responsibility. See also: *Responsibility*

**Residual** The amount of unused paid time off entitlement an employee loses at the end of an accrual term. Typically employees can carry over unused time, up to a maximum, but they lose any residual time that exceeds this limit. See also: *Carry Over*

**Responsibility** A level of authority in an application. Each responsibility lets you access a specific set of Oracle Applications forms, menus, reports, and data to fulfill your business role. Several users can share a responsibility, and a single user can have multiple responsibilities. See also: *Security Profile, User Profile Options, Request Groups, Security Groups*

**Retry** Method of correcting a payroll run or other process *before* any post-run processing takes place. The original run results are deleted and the process is run again.

**Revenue Canada** Department of the Government of Canada which, amongst other responsibilities, administers, adjudicates, and receives remittances for all taxation in Canada including income tax, Employment Insurance premiums, Canada Pension Plan contributions, and the Goods and Services Tax (legislation is currently proposed to revise the name to the Canada Customs and Revenue Agency). In the province of Quebec the equivalent is the Ministère du Revenu du Québec.

**Reviewer (SSHR)** A person invited by an appraising manager to add review comments to an appraisal.

**Reversal** Method of correcting payroll runs or QuickPay runs *after* post-run processing has taken place. The system replaces positive run result values with negative ones, and negative run result values with positive ones. Both old and new values remain on the database.

**Rollback** Method of removing a payroll run or other process *before* any post-run processing takes place. All assignments and run results are deleted.

**Search by Date** An SSHR sub-function used to search for a Person by Hire date, Application date, Job posting date or search by a Training event date.

**Salary Basis** The period of time for which an employee's salary is quoted, such as hourly or annually. Defines a group of employees assigned to the same salary basis and receiving the same salary element.

**Scheduled Enrollment** A benefits plan enrollment that takes place during a predefined enrollment period, such as an open enrollment. Scheduled enrollments can be administrative, open, or unrestricted.

**Security Group** Security groups enable HRMS users to partition data by Business Group. Only used for Security Groups Enabled security. See also: *Responsibility, Security Profile, User Profile Options*

**Security Groups Enabled** Formerly known as Cross Business Group Responsibility security. This security model uses security groups and enables you to link one responsibility to many Business Groups.

**Security Profile** Security profiles control access to organizations, positions and employee and applicant records within the Business Group. System administrators use them in defining users' responsibilities. See also: *Responsibility*

**Self Appraisal** Part of the SSHR Appraisal function. This is an appraisal undertaken by an employee to rate their own performance and competencies.

**SMP** See: *Statutory Maternity Pay*

**Social Insurance Number (SIN)** A unique number provided by Human Resources Development Canada (HRDC) to each person commencing employment in Canada. The number consists of 9 digits in the following format (###-###-###).

## S

**Source Deductions Return (TP 1015.3)** A Ministere du Revenu du Quebec form which each employee must complete. This form is used by the employee to reduce his or her taxable income at source by claiming eligible credits and also provides payroll with such important information as current address, birth date, and SIN. These credits determine the amount of provincial tax to withhold from the employee's wages.

**Special Information Types** Categories of personal information, such as skills, that you define in the Personal Analysis key flexfield.

**SSHR** Oracle Self-Service Human Resources. An HR management system using an intranet and web browser to deliver functionality to employees and their managers.

**SSP** See: *Statutory Sick Pay*

**SSP Qualifying Pattern** In the UK, an SSP qualifying pattern is a series of qualifying days that may be repeated weekly, monthly or some other frequency. Each week in a pattern must include at least one qualifying day. Qualifying days are the only days for which Statutory Sick Pay (SSP) can be paid, and you define SSP qualifying patterns for all the employees in your organization so that their entitlement to SSP can be calculated.

**Standard Link** Recurring elements with standard links have their element entries automatically created for all employees whose assignment components match the link. See also: *Element Link, Recurring Elements*

**Statement of Commissions and Expenses for Source Deduction Purposes (TP 1015.R.13.1)** A Ministere du Revenu du Quebec form which allows an employee who is paid partly or entirely by commissions to pay a constant percentage of income tax based on his or her estimated commissions for the year, less allowable business expenses.

**Statement of Remuneration and Expenses (TD1X)** In Canada, the Statement of Remuneration and Expenses allows an employee who is paid partly or entirely by commission to pay a constant percentage of income tax, based on his or her estimated income for the year, less business-related expenses.

**Statutory Maternity Pay** In the UK, you pay Statutory Maternity Pay (SMP) to female employees who take time off work to have a baby, providing they meet the statutory requirements set out in the legislation for SMP.

**Standard HRMS Security** The standard security model. Using this security model you must log on as a different user to see a different Business Group.

**Statutory Sick Pay** In the UK, you pay Statutory Sick Pay (SSP) to employees who are off work for four or more days because they are sick, providing they meet the statutory requirements set out in the legislation for SSP.

**Succession Planning** An SSHR function which enables a manager to prepare a succession plan.



**Suitability Matching** An SSHR function which enables a manager to compare and rank a persons competencies.

**Superannuation Guarantee** An Australian system whereby employers are required to contribute a percentage of an eligible employee's earnings to a superannuation fund to provide for their retirement.

## T

**Tabbed Regions** Parts of a window that appear in a stack so that only one is visible at any time. You click on the tab of the required region to bring it to the top of the stack.

**Task Flows** A sequence of windows linked by buttons to take you through the steps required to complete a task, such as hiring a new recruit. System administrators can create task flows to meet the needs of groups of users.

**Terminating Employees** You terminate an employee when he or she leaves your organization. Information about the employee remains on the system but all current assignments are ended.

**Termination Rule** Specifies when entries of an element should close down for an employee who leaves your enterprise. You can define that entries end on the employee's actual termination date or remain open until a final processing date.

**Tips** An SSHR user assistance component that provides information about a field.

## U

**User Assistance Components** SSHR online help comprising tips and instructions.

**User Balances** Users can create, update and delete their own balances, including dimensions and balance feeds. See also: *Balances*

**User Profile Options** Features that allow system administrators and users to tailor Oracle HRMS to their exact requirements. See also: *Responsibility, Security Profile*

## V

**Viewer (SSHR)** A person with view only access to an appraisal. An appraising manager or an employee in a 360 Degree Self appraisal can appoint view only access to an appraisal.

## W

**WCB Account Number** In Canada, this is the account number of the provincially administered Worker's Compensation Board that the employer would use to make remittances. There would be a unique number for each of the provincially controlled boards i.e. Workplace Safety & Insurance Board of Ontario, CSST, etc.

**Waiting Days** In the UK, statutory Sick Pay is not payable for the first three qualifying days in period of incapacity for work (PIW), which are called waiting days. They are not necessarily the same as the first three days of sickness, as waiting days can be carried forward from a previous PIW if the linking interval between the two PIWs is less than 56 days.

**Work Choices** Also known as Work Preferences, Deployment Factors, or Work Factors. These can affect a person's capacity to be deployed within an enterprise, such willingness to travel or relocate. You can hold work choices at both job and position level, or at person level.

**Worker's Compensation Board** In Canada, this is a provincially governed legislative body which provides benefits to employees upon injury, disability, or death while performing the duties of the employer. Worker's Compensation Board premiums are paid entirely by the employer.

**Workflow** An Oracle application which uses charts to manage approval processes and in addition is used in SSHR to configure display values of sections within a web page and instructions.

**Work Structures** The fundamental definitions of organizations, jobs, positions, grades, payrolls and other employee groups within your enterprise that provide the framework for defining the work assignments of your employees.

# Index

## A

- Absence types, database items for, 2 – 65
- Alias statements, 1 – 16
  - format of, 2 – 34
- Arithmetic operators, in formulas, 2 – 12
- Assignment set formulas, 1 – 6, 1 – 36
- Assignment statements, 1 – 10
  - format of, 2 – 36

## B

- Benefits, formula types used for, 2 – 69 to 2 – 87

## C

- Compiling formulas, 1 – 20
- Compiling multi-version formulas, 1 – 37
- Conditions, format of, 2 – 37
- Constants, 2 – 5
  - date, 2 – 6
  - numeric, 2 – 5
  - text, 2 – 5

## D

- Database items, 2 – 8
  - from element input values, 2 – 4
  - in formulas, 1 – 9
  - list of dynamic items, 2 – 61
  - list of static items, 2 – 44

- Default statements, 1 – 16
  - format of, 2 – 34
- Define global values, 1 – 22
- Descriptive flexfields, enabling for QuickPaint, 2 – 64
- Dynamic database items
  - for descriptive flexfield segments, 2 – 64 to 2 – 66
  - for key flexfield segments, 2 – 65
  - for new absence types, 2 – 65
  - for new elements, 2 – 61 to 2 – 64
  - for new grade rates, 2 – 63
  - for new pay scale rates, 2 – 63

## E

- Element entries, validation, 1 – 6
- Element skip rules, 1 – 33
- Elements, database items for, 2 – 61 to 2 – 64
- Example
  - accrual formula, 2 – 93
  - checking a user table entry, 1 – 40
  - checking an element entry, 1 – 39
  - editing a QuickPaint formula, 1 – 38
- Expressions, in formulas, 2 – 11

## F

- Formula types, for total compensation, 2 – 69 to 2 – 87
- Formulas
  - arithmetic operators in, 2 – 12

- assignment set formulas, 1 – 36
- comments in, 1 – 15 to 1 – 17
- compiling multi-version formulas, 1 – 37
- conditions in, 1 – 11 to 1 – 14
- context, 1 – 9
- errors in, 2 – 41 to 2 – 42
- for assignment sets, 1 – 6
- for benefits administration, 2 – 68
- for element skip rules, 1 – 33
- for payroll calculations, 1 – 5 to 1 – 7
- for PTO accrual plans, 1 – 26
- for QuickPaint, 1 – 6
- for skip rules, 1 – 33
- for validating element entries and user tables, 1 – 6, 1 – 35
- functions in, 2 – 13 to 2 – 18
- inputs, 1 – 9 to 1 – 14, 2 – 3 to 2 – 4
- making formulas efficient, 1 – 17 to 1 – 19
- structure, 2 – 2
- types of statement, 2 – 34
- using comments, 2 – 33
- variables in, 2 – 7 to 2 – 10
- writing and editing, 1 – 20

#### Functions

- in formulas, 2 – 13
- registering, 1 – 23
- user defined, 2 – 88

## G

- Global values, 1 – 10, 2 – 7
  - defining, 1 – 22
- Grade rates, database items for, 2 – 63

## H

- HR:Execute Legislative Check Formula within Run, 2 – 67

## I

- If statements, format of, 2 – 36
- Input values
  - in payroll formulas, 2 – 3
  - validation, 1 – 6

- Inputs statement, 1 – 9
- Inputs statements, format of, 2 – 35

## K

- Key flexfields, enabling for QuickPaint, 2 – 65

## L

- Local variables, 1 – 10, 2 – 7

## O

- Oracle FastFormula
  - overview, 1 – 4
  - predefined formulas, 1 – 3
  - summary of uses, 1 – 2

## P

- Pay scale rates, database items for, 2 – 63
- Payroll formulas, 1 – 5 to 1 – 7
  - writing for elements, 1 – 25
- Processes
  - Bulk Compile Formulas, 2 – 40
  - Create Descriptive Flexfield DB Items, 2 – 64
  - Create Key Flexfield DB Items, 2 – 65
- PTO accrual plans
  - changing plan rules, 2 – 93
  - formulas for, 1 – 26
  - sample formula, 2 – 93

## Q

- QuickPaint formula, copying and adding features to, 1 – 34
- QuickPaint reports, writing formulas for, 1 – 6

## R

- Registering, functions, 1 – 23

Return statements, in formulas, 2 – 39  
Rule types, for total compensation, 2 – 69 to 2  
– 87

## **S**

Skip rules, writing formulas for, 1 – 33  
Statements, in formulas, 2 – 34 to 2 – 39

## **T**

Total compensation, formula types used for, 2  
– 69 to 2 – 87

## **U**

User tables, validating entries, 1 – 6

## **V**

Validation, writing formulas for, 1 – 6, 1 – 35  
Variables, in formulas, 2 – 7

## **W**

WAS DEFAULTED condition, 1 – 12

### Windows

Database Items, 1 – 20  
Define Function, 2 – 88  
Formula, 1 – 20  
Globals, 1 – 22

