Oracle FastFormula [™] User's Guide

Release 11.0

Part No. A58346-01



Oracle FastFormula User's Guide, Release 11.0

Part No. A58346-01

Copyright © Oracle Corporation 1995

All rights reserved. Printed in the U.S.A.

Primary Author: John Cafolla, Juliette Fleming and Louise Raffo

Contributors: Mark Callaghan, Michael O'Shea

This software was not developed for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It is the customer's responsibility to take all appropriate measures to ensure the safe use of such applications if the programs are used for such purposes.

This software/documentation contains proprietary information of Oracle Corporation; it is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited.

If this software/documentation is delivered to a U.S. Government Agency of the Department of Defense, then it is delivered with Restricted Rights and the following legend is applicable:

Restricted Rights Legend Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of DFARS 252.227–7013, Rights in Technical Data and Computer Software (October 1988).

Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

If this software/documentation is delivered to a U.S. Government Agency not within the Department of Defense, then it is delivered with "Restricted Rights", as defined in FAR 52.227–14, Rights in Data – General, including Alternate III (June 1987).

The information 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.

ORACLE, Oracle Alert, Oracle Financials, SQL*Plus are registered trademarks of Oracle Corporation

Oracle Application Object Library, Oracle General Ledger, Oracle Human Resources, Oracle Payroll, Oracle Reports, PL/SQL, SQL*TextRetrieval are trademarks of Oracle Corporation.

All other products or company names are used for identification purposes only, and may be trademarks of their respective owners.

Contents

Preface	Preface i About This User's Guide ii Finding the Latest Information ii Assumptions iii Do Not Use Database Tools to Modify Oracle Applications Data iv Other Information Sources v
	About Oracle viii Thank You viii
Chapter 1	Oracle FastFormula 1 – 1 Oracle FastFormula Overview 1 – 2
	Writing a Simple Payroll Calculation
	Types of Input
	Writing More Complex Calculations
	Incorporating Conditions
	Commenting your Formulas 1 – 8
	Formula Structure
	Using the Formula Window 1 – 10
	Using the Globals Window 1 – 12
	Defining Functions
	Using Oracle FastFormula for Payroll Calculations 1 – 16
	Accessing Input Values in Payroll Formulas
	Writing Efficient Payroll Calculation Formulas 1 – 19
	Writing Formulas for Element Skip Rules 1 – 2

	Using Oracle FastFormula for Validation	23
	Examples	24
	Using Oracle FastFormula to Define QuickPaint Reports 1 -	26
	Using Oracle FastFormula to Define Assignment Sets 1 –	28
	Oracle FastFormula Reference	29
	Constants and Variables	30
	Constants	31
	Variables	32
	Expressions	35
	Arithmetic Operators	36
	Functions 1 -	36
	Text Functions	37
	Numeric Functions	39
	Date Functions	40
	Data Conversion Functions 1 –	41
	GB Functions	44
	Comments	45
	Statements	45
	Alias Statement	45
	Default Statement1 -	46
	Inputs Statement	46
	Assignment Statement	
	IF Statement	48
	Return Statement	50
	Compiling Formulas	50
	Formula Errors	51
Appendix A	Database Items	- 1
• •	Static Database Items	- 2
	Applicant Information A -	- 2
	Employee Assignment Information	
	Contact Addresses	
	Contact Information	- 8
	Employee Hire Information	- 10
	Location Details	
	Payroll Details A -	- 12
	People Addresses A -	
	People Information	
	Recruiter Information A -	
	Supervisor Information	- 18
	Date Information A -	

Dynamic Database Items	. A – 19
Element Database Items	. A – 19
Grade Rate Database Items	. A – 22
Pay Scale Rate Database Items	. A – 22
Descriptive Flexfield Database Items	. A – 23
Key Flexfield Database Items	. A – 25
Absence Database Items	. A – 26

Glossary

Index

Preface

Welcome to Release 11.0 of the Oracle FastFormula User's Guide.

This guide includes 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 the guide is organized and introduces other sources of information that can help you.

About This User's Guide

The main body of information you need to understand Oracle FastFormula together with the reference information for each command or function is contained in Chapter 1.

The Database Item Appendix provides a reference source that lists all static and dynamic database items that are available to you in FastFormula. This list of items provides a simple method of accessing your HRMS data in FastFormula.

This user's guide is available online

All Oracle Applications user's guides are available online, in both HTML and Adobe Acrobat format. Most other Oracle Applications documentation is available in Adobe Acrobat format.

The paper and online versions of this manual have identical content, and you can use whichever format is more convenient.

The HTML version of this book is optimized for on–screen reading, and lets you follow hypertext links for easy access to books across our entire library. You can also search for words and phrases if your national language is supported by Oracle's Information Navigator. The HTML documentation is available from the Oracle Applications toolbar, or from a URL provided by your system administrator. Note that the HTML documentation is translated into over twenty languages.

You can order an Oracle Applications Documentation Library CD containing Adobe Acrobat versions of each manual in the Oracle Applications documentation set. Using this CD, you can search for information, read it on–screen, and print individual pages, sections, or entire books. When you print from Adobe Acrobat, the resulting printouts look just like pages from an Oracle Applications hardcopy manual.

Finding the Latest Information

For information about any new features that were not available when this user's guide was printed, look at the What's New? section on the main Help menu. This information is updated for each new release of Oracle FastFormula HTML Help.

Assumptions

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

- · the principles and customary practices of your business area
- the places where FastFormula are used in the Oracle HRMS product set.
 - If you are not familiar with Oracle HR or Oracle Payroll we suggest you attend the introduction or implementation training classes for these product available through Oracle Education.
- the Oracle Applications graphical user interface.
 To learn more about this, read the *Oracle Applications User's Guide*.

See Other Information Sources below for further details of Oracle Applications product information.

Do Not Use Database Tools to Modify Oracle Applications Data

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 like 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 forms, 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 forms to modify your data, Oracle Applications automatically checks that your changes are valid. Oracle Applications also keeps track of who changes information. But, 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.

Consequently, we STRONGLY RECOMMEND that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle Applications tables, unless we tell you to do so in our manuals.

Other Information Sources

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

Most Oracle Applications documentation is available in Adobe Acrobat format on the *Oracle Applications Documentation Library* CD. We supply this CD with every software shipment.

If this guide refers you to other Oracle Applications documentation, use only the Release 11 versions of those books unless we specify otherwise.

Oracle Applications User's Guide

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

You can access this guide online by choosing "Getting Started with Oracle Applications" from any Oracle Applications help file.

Related User's Guides

Oracle FastFormula is integrated into the Oracle HRMS product set. Therefore, you may want to refer to other HRMS user's guides when you use Oracle FastFormula.

If you do not have the hardcopy versions of these manuals, you can read them by choosing Library from the Help menu, or by reading from the Oracle Applications Document Library CD, or by using a web browser with a URL that your system administrator provides.

Oracle Payroll User's Guide

This guide is aimed at Payroll managers and explains how to set up and use Oracle Payroll to pay your employees and meet necessary legislative requirements.

Oracle Human Resources User's Guide

This guide is aimed at HR managers and explains how to set up and use Oracle Human Resources to meet the requirements of your enterprise. It describes how you can represent your enterprise

structures, policies, and people on the system and use this information to manage your human resources.

Oracle HRMS Implementation Guide

This guide includes modular implementation flowcharts and checklists to assist with your project planning. It contains a summary of the sequence of recommended steps for implementing Oracle FastFormula and Oracle Payroll. Full instructions for each implementation step are contained in the User's Guide.

Oracle Applications Flexfields Guide

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

Oracle Applications User Interface Standards

This manual 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 4.5.

Installation and System Administration

Oracle Applications Installation Manual

This manual and the accompanying release notes provide information you need to successfully install Oracle Financials, Oracle Public Sector Financials, Oracle Manufacturing, or Oracle Human Resources in your specific hardware and operating system software environment.

Oracle Applications Upgrade Manual

This manual explains how to prepare your Oracle Applications products for an upgrade. It also contains information on finishing the upgrade procedure for each product. Refer to this manual and the *Oracle Applications Installation Manual* when you plan to upgrade your products.

Oracle Applications System Administrator's Guide

This manual 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 processing.

Oracle HRMS Technical Reference Manual

This manual contains database diagrams and a description of Oracle HRMS database tables, forms, reports, and programs. This information helps you convert data from your existing applications, integrate Oracle HRMS with other non–Oracle applications, and write custom reports.

You can order a technical reference manual for any product you have licensed. Technical reference manuals are available in paper format only.

Other Information

Training

Oracle Education offers a complete set of training courses to help you and your staff master Oracle Applications. We can help you develop a training plan that provides thorough training for both your project team and your end users. We will work with you to organize courses appropriate to your job or area of responsibility.

Training professionals can show you how to plan your training throughout the implementation process so that the right amount of information is delivered to key people when they need it the most. You can attend courses at any one of our many Educational Centers, or you can arrange for our trainers to teach at your facility. In addition, we can tailor standard courses or develop custom courses to meet your needs.

Support

From on–site support to central support, our team of experienced professionals provides the help and information you need. 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 Oracle server, and your hardware and software environment.

About Oracle

Oracle develops and markets an integrated line of software products for database management, applications development, decision support, and office automation, as well as a complete family of financial, manufacturing, and human resource applications.

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 offers its products, along with related consulting, education, and support services, in over 140 countries around the world. Oracle Corporation is the world's leading supplier of software for information management, and is the world's second largest software company.

Thank You

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

We value your comments and feedback. At the end of this manual is a Reader's Comment Form you can use to explain what you like or dislike about Payables 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.

CHAPTER

1

Oracle FastFormula

O racle 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. The following topics explain how to write formulas using Oracle FastFormula:

- Oracle FastFormula Overview assumes no knowledge of formula writing or editing. It progressively introduces the features of Oracle FastFormula using examples.
- Using Oracle FastFormula for Payroll Calculations explains some additional considerations for using formulas in the payroll run.
- Using Oracle FastFormula for Validation explains the required format for formulas to check element entry values and user table entry values.
- Using Oracle FastFormula to Define QuickPaint Reports explains how you can edit the formulas generated from your definitions of QuickPaint reports.
- Using Oracle FastFormula to Define Assignment Sets explains how you can view and edit the formulas generated from your definitions of assignment sets.
- Oracle FastFormula Reference is a comprehensive reference to all the features of Oracle FastFormula and the rules you must follow in formula writing.

Oracle FastFormula Overview

Formulas are generic expressions of calculations or comparisons you want to repeat with different input values. They can take input from the window, database, or a process, such as a payroll run. They can return values or messages.

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
- 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.

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

In Oracle Payroll, the system generates most of the formulas needed for payroll processing, in response to your entry of information about your earnings and deductions. When necessary you can edit these generated payroll formulas.

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.

Writing a Simple Payroll Calculation

There are strict rules as to the words and punctuation you must use in a formula so that Oracle FastFormula can correctly perform the calculations and validation you require.

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 his hourly rate. You could write this formula:

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

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.

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 get values:

- it can receive them as input when the formula is called
- it can find the values in the database from database items
- it can use *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*.

Using Inputs Statement

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)
```

Using 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, as well as 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 – 19.

Using Global Values

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: Using the Globals Window: page 1 - 12

Using 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.

Writing More Complex Calculations

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 1 – 36

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.

Using Simple Conditions

For example:

```
IF age < 20 THEN
   training_allowance = 30
ELSE
   training_allowance = 0</pre>
```

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.

Making Conditions More Complex

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: Oracle FastFormula Reference: page 1 – 29

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
   MESG = '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.

Combining 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:

```
(days_between(end_period_date, start_date) + 1) >= 183
    employee status = 'FULL TIME'
AND
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:

```
X = 6 AND NOT Y = 7 OR P >= 6
is interpreted as:
           (X = 6 \text{ AND (NOT } Y = 7)) \text{ 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.

Actions based on **Conditions**

Incorporating Multiple 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:

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.

Commenting your Formulas

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:

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.

Formula Structure

A valid formula has a well defined structure. It is made up of comments and statements. Oracle FastFormula ignores comments and carries out the instructions in the statements.

We have already looked at four types of statement:

- Assignment statement
- Inputs statement
- Return statement
- If statement

There are two other types of statement you can use:

- Alias statement
- · Default statement

Alias Statement

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 Statement

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
    MESG = '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.



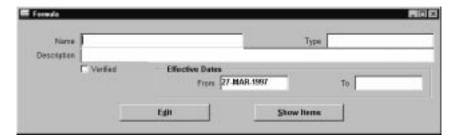
Attention: You must use the Default statement for database items that can be empty. The Database Items window includes a check box labelled Default Required. This check box is checked for database items that can be empty. The Database Items window appears when you choose the Show Items button on the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

Order of Statements

Statements in your formula must appear in the following order:

- 1) ALIAS statements (if any)
- 2) DEFAULT FOR statements (if any)
- 3) INPUT statement (if any)
- 4) Other statements

Using the Formula Window



➤ 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.

- 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.

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.

See Also

Defining an Element (HR Users), *Oracle Human Resources User's Guide*Defining an Element (Payroll Users), *Oracle Payroll User's Guide*

Defining an Element's Input Values, *Oracle Human Resources User's Guide*

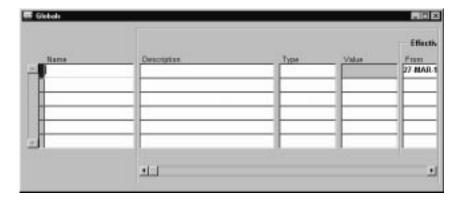
Setting Up User Tables, Oracle Human Resources User's Guide

Using the Globals Window

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 datetracked 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.

Defining Functions

Many functions are supplied with Oracle FastFormula. See: Functions: page 1 – 36. 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 allows 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, they should not not perform any data manipulation as FastFormula is intended as a read only tool.

Registering the 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 three classes of functions you can register:

- external: These are further PL/SQL functions in addition to the ones already delivered with FastFormula. They do not take contexts.
- **formula:** These functions are formulas called from other formulas. This class of function is not supported in Release 11.
- user defined: In practice, this is the only class of function you need define. User defined functions require both contexts and parameters.

► To register a new function for Oracle FastFormula:

- 1. Enter a unique name for the new function.
- 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.
- 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 checkbox if you want the corresponding parameter to be optional.
- 7. Check the Continuing checkbox 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.

Using Oracle FastFormula for Payroll Calculations

You use Oracle FastFormula to define and maintain the formulas you need for pay and pay-related calculations.

You use Oracle FastFormula:

 when you want 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).

See: Using the Formula Window: page 1 – 10

 if you decide to define an element for an earnings or deduction and write its formula yourself, instead of initiating the earnings or deduction. You would do this only for an earnings or deduction with a highly complex calculation requiring a number of different calls to the database.

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.



Additional Information: Refer to your Oracle Payroll manual for information about predefined payroll formulas.

▶ To write formulas for elements you define:

- 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
- Write the formula or formulas for calculating the run results (formula type = Oracle Payroll).
- 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.

Defining an Element (HR Users), *Oracle Human Resources User's Guide* Defining an Element (Payroll Users), *Oracle Payroll User's Guide*

Accessing 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.

As a simple 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.

As a simple 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 fires 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 fires 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.

Writing Efficient Payroll Calculation Formulas

The following statements are true as general guidelines 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.

Formula Format

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

Variable Names and Aliases

Use names that are brief yet meaningful to improve readability. 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:

Date Literals

Use the TO_DATE function only when the operand is a variable.

Inefficient:

```
Start_date = TO_DATE ( '12-JAN-1992' )
Efficient:
Start_date = '12-JAN-1992' (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</pre>
```

Efficient:

```
IF Salary < 20000 THEN
   IF Age < 20 THEN
        Training_allowance = 30
ELSE</pre>
```

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

Balance Dimensions

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.

Writing Formulas for Element Skip Rules

If your payroll policies require periodic or 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 three periods, or
- process the Union Fees element every run unless the Union_Fees_Paid balance is greater than 10 000.

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
```

Using Oracle FastFormula for Validation

You can use Oracle FastFormula to validate user entries into the element input values, and to user tables that you define. In both cases, you must write and validate the formula before you define the element or table, so that you can select the formula from a list in the Element window or Columns window.

In the Formulas window, select formula type Element Input Validation or User Table Validation.

When writing either type of formula, you must observe the following rules:

- 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')
```

Examples

Checking an Element Entry

Figure A – 1 Salary Element The formula below checks that the entry value of the Salary element does not exceed 200 000.

Salary		
Classification	Priority	Туре
Earnings	2500	R
Value Name	Туре	Valid
- Pay Value	Money	F
		L

```
/* 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

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</pre>
  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
```

See Also

Defining an Element (HR Users), *Oracle Human Resources User's Guide*Defining an Element (Payroll Users), *Oracle Payroll User's Guide*Defining an Element's Input Values, *Oracle Human Resources User's Guide*

Setting Up User Tables, Oracle Human Resources User's Guide

Using Oracle FastFormula to Define 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.



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.

Example

In this example, the 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)
LINE10=''
LINE11=''
LINE12=''
RETURN LINE01, LINE02, LINE03, LINE04, LINE05, LINE06, LINE07,
       LINE08, LINE09, LINE10, LINE11, LINE12,
```

► 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.
- 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.
- Paste the text of the QuickPaint formula into the Edit Formula window.

6. Save your work.

See Also

Defining QuickPaint Reports, Oracle Human Resources User's Guide

Using Oracle FastFormula to Define Assignment Sets

You define an assignment set in the Assignment Set window. You can select database items that you can use to define your set. When you choose the generate button in the Assignment Set window, a formula is generated from the criteria you entered.

You should not normally need to edit assignment set formulas.

If you enter multiple criteria to define an assignment set, the conditions that are generated are joined by AND or OR. You might 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.

See Also

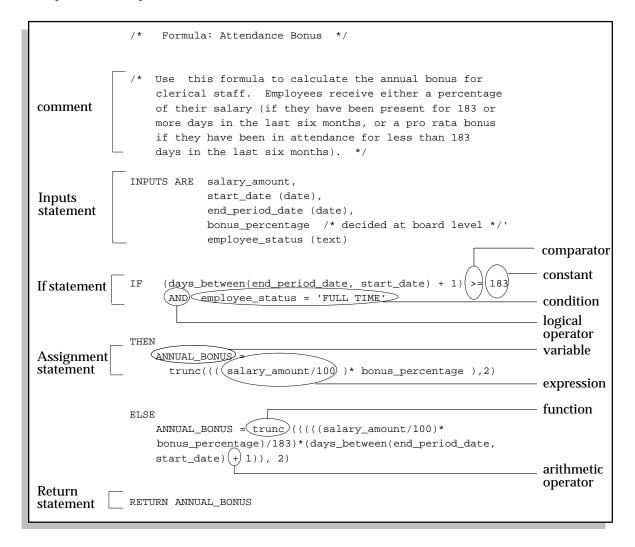
Creating an Assignment Set, Oracle Human Resources User's Guide

Oracle FastFormula 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*.

An example of each of these formula components is identified in the sample formula in Figure A-2. There are rules about how you use each of these components, and these rules are explained in this reference section.

Figure A – 2 Components in a Sample Formula



Constants and Variables

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

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 Glebe Lane'
- '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 DD-MON-YYYY.

Examples of valid date constants are:

- '12-MAR-1989' (DATE)
- '4-FEB-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 tell Oracle FastFormula what type you want a variable to be. It works this out by seeing how you use the variable. For example, if you set a variable to 'J. Smith', Oracle FastFormula knows it is a text variable.

Oracle FastFormula 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:

Local Variable Variables that occur in a single formula only.

Global Value Values that can occur in any formula.

Database Item 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: Using the Globals Window: page 1 – 12

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

Dynamic Database Items: page - 29.

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:

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.
- 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.

There are a number of words you cannot use as the names of variables. These reserved words are:

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

Any word consisting only of digits (that is, that could be treated as a number)

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 1 – 45

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.

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

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_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)
```

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.

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.

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 ...</pre>
```

Or, to check if Tax_code is a string of at least two characters starting with 'GG':

```
IF Tax_code LIKE 'GG%' THEN ...
```

UPPER

UPPER(expr)

The UPPER function convert 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.

ROUND

ROUND(n [, m])

The ROUND function rounds off a numeric value n to m decimal places. The first operand is the value Oracle FastFormula rounds off, the second the number of places Oracle FastFormula rounds off to.

Examples:

```
ROUND (2.3401, 2) returns 2.34

ROUND (2.3461, 2) returns 2.35

ROUND (2.345,2) returns 2.35
```

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.

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

Example:

TRUNC(2.3401, 2) returns 2.34.

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', 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.

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('27-JUN-1995', '03-JUL-1995') 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.

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.

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 'DD–MON–YYYY' if no format is provided. The day and year must be in numeric form. For example:

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

```
date_1 = '12-JAN-1989'(date)
```

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'.

TO TEXT

```
TO_TEXT(n) TO_TEXT (date1 [, format])
```

TO_CHAR(n) TO_CHAR(date1 [, format])

The TO_TEXT function converts:

- the number *n* from number data type to text data type
- the date date1 from date data type to text data type. The optional format should be a text string like 'DD-MON-YYYY'. This is the default.

For example:

```
birthdate = '21-JAN-1960' (date)
     mesg = 'Birthdate is: ' + TO_CHAR (birthdate)
/* sets mesg to 'Birthdate is: 21-JAN-1960' */
     mesg = 'Birthdate is: ' + TO_CHAR (birthdate,
            'DD-MON-YY')
/* sets mesg to 'Birthdate is: 21-JAN-60' */
     mesg = 'Birthdate is: ' + TO_CHAR (birthdate,
             'DD Month Year')
/ \, ^{\star} sets mesg to 'Birthdate is: 21 January Nineteen-Sixty' ^{\star} /
```

GB Functions

Function calls to PL*SQL packaged functions provide useful extensions to Oracle FastFormula. The following functions have been registered

for use in GB legislative formulas:

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.

Comments

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

Comments start with a slash asterisk (/*) sequence, and finish with an asterisk slash sequence (*/). 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
   MESG = '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 varnamel(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:

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 in 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 set 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 IF statement lets you check one or more conditions.
- 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. A logical operator is either AND or OR. These have the effect of combining the conditions logically:

- 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.

Also, each condition can be preceded by the NOT operator, which 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.
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	LIKE	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. – percent sign (%) matches any number of characters in that position – underscore (_) matches a single character occurrence in that position.
Not Like	NOT LIKE	Text	The condition is true if the two text expressions do NOT match according to the rules of the LIKE syntax.

Table A – 1

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.

Compiling Formulas

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.

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.

Below is a list of the basic Oracle FastFormula errors that can occur at run-time.

Uninitialized Variables An uninitialized local variable is one that has no value when the formula runs. This causes an error in all statements except the Return statement. For example:

```
(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 indicating an error back to the application (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:

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 database item contains a bad 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.

APPENDIX



Database Items

This appendix 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.

Applicant Information

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

Table B – 1

Employee Assignment Information

Database item	Description
ASG_ASSIGNMENT_SEQUENCE	This is used as a default for assignment number
ASG_DATE_FROM	The date from which this assignment information is effective
ASG_DATE_TO	The date to which this assignment information is effective
ASG_EMPLOYMENT_CATEGORY	The employment category for the assignment
ASG_END_TIME	The standard end time for the assignment
ASG_FREQ	The frequency for which the assignment working hours are measured
ASG_GRADE	The employee's grade
ASG_GRADE_DATE_FROM	The date from which this assignment grade information is effective

Table B – 2

ASG_GROUP ASG_GROUP The employee's group The standard number of working hours for the assignment ASG_INT_ADDR_LINE The internal address of the assignment ASG_JOB ASG_JOB_DATE_FROM The date from which this assignment job information is effective ASG_JOB_DATE_TO The date to which this assignment job information is effective ASG_LAST_CHANGE_REASON The reason the salary was change Last performance review date ASG_LAST_PERFORMANCE_DATE ASG_LAST_PERFORMANCE_LOCATION ASG_LAST_PERFORMANCE_RATING ASG_LAST_PERFORMANCE_TYPE Last performance review type ASG_LAST_PROC_PAYROLL_NAME The payroll name the assignment was last processed ASG_LAST_PROC_PERIOD_ID The time period ID the assignment was last processed The period name the assignment was last processed The period number the assignment was last processed	Database item	Description
ASG_INT_ADDR_LINE The internal address of the assignment ASG_JOB The employee's job The date from which this assignment job information is effective ASG_JOB_DATE_FROM The date to which this assignment job information is effective ASG_LAST_CHANGE_REASON The reason the salary was change Last performance review date ASG_LAST_PERFORMANCE_DATE ASG_LAST_PERFORMANCE_LOCATION ASG_LAST_PERFORMANCE_RATING ASG_LAST_PERFORMANCE_TYPE ASG_LAST_PERFORMANCE_TYPE ASG_LAST_PERFORMANCE_TYPE ASG_LAST_PROC_PAYROLL_NAME The payroll name the assignment was last processed ASG_LAST_PROC_PERIOD_ID The time period ID the assignment was last processed The period name the assignment was last processed	ASG_GRADE_DATE_TO	The date to which this assignment grade information is effective
hours for the assignment ASG_INT_ADDR_LINE The internal address of the assignment The internal address of the assignment The employee's job ASG_JOB_DATE_FROM The date from which this assignment job information is effective ASG_JOB_DATE_TO The date to which this assignment job information is effective ASG_LAST_CHANGE_REASON The reason the salary was change ASG_LAST_PERFORMANCE_DATE Last performance review date ASG_LAST_PERFORMANCE_LOCATION ASG_LAST_PERFORMANCE_RATING Last performance review rating ASG_LAST_PERFORMANCE_TYPE Last performance review type ASG_LAST_PROC_PAYROLL_NAME The payroll name the assignment was last processed ASG_LAST_PROC_PERIOD_ID The time period ID the assignment was last processed ASG_LAST_PROC_PERIOD_NAME The period name the assignment was last processed The period name the assignment was last processed The period name the assignment was last processed	ASG_GROUP	The employee's group
ASG_JOB The employee's job ASG_JOB_DATE_FROM The date from which this assignment job information is effective ASG_JOB_DATE_TO The date to which this assignment job information is effective ASG_LAST_CHANGE_REASON The reason the salary was change Last performance review date ASG_LAST_PERFORMANCE_DATE Last performance review location ASG_LAST_PERFORMANCE_LOCATION Last performance review rating Last performance review rating Last performance review type ASG_LAST_PERFORMANCE_TYPE Last performance review type ASG_LAST_PERFORMANCE_TYPE Last performance review type The payroll name the assignment was last processed ASG_LAST_PROC_PERIOD_ID The time period ID the assignment was last processed ASG_LAST_PROC_PERIOD_NAME The period name the assignment was last processed ASG_LAST_PROC_PERIOD_NUMBER The period number the assignment was last processed	ASG_HOURS	The standard number of working hours for the assignment
ASG_JOB_DATE_FROM The date from which this assignment job information is effective ASG_JOB_DATE_TO The date to which this assignment job information is effective ASG_LAST_CHANGE_REASON The reason the salary was change Last performance review date ASG_LAST_PERFORMANCE_DATE ASG_LAST_PERFORMANCE_LOCATION Last performance review location ASG_LAST_PERFORMANCE_RATING ASG_LAST_PERFORMANCE_TYPE Last performance review type ASG_LAST_PROC_PAYROLL_NAME The payroll name the assignment was last processed ASG_LAST_PROC_PERIOD_ID The time period ID the assignment was last processed ASG_LAST_PROC_PERIOD_NAME The period name the assignment was last processed ASG_LAST_PROC_PERIOD_NAME The period name the assignment was last processed	ASG_INT_ADDR_LINE	
assignment job information is effective ASG_JOB_DATE_TO The date to which this assignment job information is effective ASG_LAST_CHANGE_REASON The reason the salary was change Last performance review date ASG_LAST_PERFORMANCE_DATE ASG_LAST_PERFORMANCE_LOCATION Last performance review location Last performance review rating ASG_LAST_PERFORMANCE_RATING Last performance review type ASG_LAST_PERFORMANCE_TYPE Last performance review type ASG_LAST_PROC_PAYROLL_NAME The payroll name the assignment was last processed ASG_LAST_PROC_PERIOD_ID The time period ID the assignment was last processed ASG_LAST_PROC_PERIOD_NAME The period name the assignment was last processed ASG_LAST_PROC_PERIOD_NUMBER The period number the assignment was last processed	ASG_JOB	The employee's job
ASG_LAST_PERFORMANCE_DATE ASG_LAST_PERFORMANCE_LOCATION ASG_LAST_PERFORMANCE_LOCATION Last performance review location ASG_LAST_PERFORMANCE_RATING ASG_LAST_PERFORMANCE_TYPE Last performance review rating ASG_LAST_PERFORMANCE_TYPE Last performance review type Last performance review type Last performance review type The payroll name the assignment was last processed ASG_LAST_PROC_PERIOD_ID The time period ID the assignment was last processed ASG_LAST_PROC_PERIOD_NAME The period name the assignment was last processed ASG_LAST_PROC_PERIOD_NUMBER The period number the assignment was last processed	ASG_JOB_DATE_FROM	assignment job information is
ASG_LAST_PERFORMANCE_DATE ASG_LAST_PERFORMANCE_LOCATION Last performance review location ASG_LAST_PERFORMANCE_RATING Last performance review rating ASG_LAST_PERFORMANCE_TYPE Last performance review type ASG_LAST_PROC_PAYROLL_NAME The payroll name the assignment was last processed ASG_LAST_PROC_PERIOD_ID The time period ID the assignment was last processed ASG_LAST_PROC_PERIOD_NAME The period name the assignment was last processed ASG_LAST_PROC_PERIOD_NAME The period name the assignment was last processed	ASG_JOB_DATE_TO	The date to which this assignment job information is effective
ASG_LAST_PERFORMANCE_LOCATION ASG_LAST_PERFORMANCE_RATING Last performance review rating ASG_LAST_PERFORMANCE_TYPE Last performance review type ASG_LAST_PROC_PAYROLL_NAME The payroll name the assignment was last processed ASG_LAST_PROC_PERIOD_ID The time period ID the assignment was last processed ASG_LAST_PROC_PERIOD_NAME The period name the assignment was last processed ASG_LAST_PROC_PERIOD_NAME The period name the assignment was last processed The period name the assignment was last processed	ASG_LAST_CHANGE_REASON	The reason the salary was changed
ASG_LAST_PERFORMANCE_RATING ASG_LAST_PERFORMANCE_TYPE Last performance review type ASG_LAST_PROC_PAYROLL_NAME The payroll name the assignment was last processed ASG_LAST_PROC_PERIOD_ID The time period ID the assignment was last processed ASG_LAST_PROC_PERIOD_NAME The period name the assignment was last processed ASG_LAST_PROC_PERIOD_NUMBER The period number the assignment was last processed	ASG_LAST_PERFORMANCE_DATE	Last performance review date
ASG_LAST_PERFORMANCE_TYPE ASG_LAST_PROC_PAYROLL_NAME The payroll name the assignment was last processed ASG_LAST_PROC_PERIOD_ID The time period ID the assignment was last processed ASG_LAST_PROC_PERIOD_NAME The period name the assignment was last processed ASG_LAST_PROC_PERIOD_NUMBER The period number the assignment was last processed	ASG_LAST_PERFORMANCE_LOCATION	Last performance review location
ASG_LAST_PROC_PAYROLL_NAME The payroll name the assignment was last processed ASG_LAST_PROC_PERIOD_ID The time period ID the assignment was last processed ASG_LAST_PROC_PERIOD_NAME The period name the assignment was last processed ASG_LAST_PROC_PERIOD_NUMBER The period number the assignment was last processed	ASG_LAST_PERFORMANCE_RATING	Last performance review rating
ASG_LAST_PROC_PERIOD_ID The time period ID the assignment was last processed ASG_LAST_PROC_PERIOD_NAME The period name the assignment was last processed ASG_LAST_PROC_PERIOD_NUMBER The period number the assignment was last processed	ASG_LAST_PERFORMANCE_TYPE	Last performance review type
ASG_LAST_PROC_PERIOD_NAME The period name the assignment was last processed ASG_LAST_PROC_PERIOD_NUMBER The period number the assignment was last processed	ASG_LAST_PROC_PAYROLL_NAME	The payroll name the assignment was last processed
was last processed ASG_LAST_PROC_PERIOD_NUMBER The period number the assignment	ASG_LAST_PROC_PERIOD_ID	The time period ID the assignment was last processed
ASG_LAST_PROC_PERIOD_NUMBER The period number the assignment	ASG_LAST_PROC_PERIOD_NAME	
was last processed	ASG_LAST_PROC_PERIOD_NUMBER	The period number the assignment was last processed
ASG_LAST_PROPOSED_SALARY_CHANGE The proposed salary chage	ASG_LAST_PROPOSED_SALARY_CHANGE	The proposed salary chage
ASG_LAST_PROPOSED_SALARY_PERCENT The proposed salary change as a percentage	ASG_LAST_PROPOSED_SALARY_PERCENT	
ASG_LAST_SALARY_CHANGE_APPROVED Whether the last proposed salary change has been approved	ASG_LAST_SALARY_CHANGE_APPROVED	Whether the last proposed salary change has been approved
ASG_LAST_SALARY_DATE The last salary change date	ASG_LAST_SALARY_DATE	The last salary change date
ASG_LOCATION The employee's location	ASG_LOCATION	The employee's location

Table B – 2

Database item	Description
ASG_LOC_INACTIVE_DATE	The date to which the location information is effective
ASG_MANAGER	Whether the assignment is a managerial assignment (yes/no)
ASG_NEXT_PERFORMANCE_DATE	Next performance review date
ASG_NEXT_SALARY_DATE	The date of the next salary change
ASG_NUMBER	The assignment number
ASG_ORG	The employee's organization
ASG_ORG_DATE_FROM	The date from which assignment organization information is effective
ASG_ORG_DATE_TO	The date to which assignment organization information is effective
ASG_PAYROLL	The employee's payroll
ASG_PERFORMANCE_REVIEW_FREQUENCY	The performance review frequency for the assignment
ASG_PERFORMANCE_REVIEW_PERIOD	The performance review frequency for the assignment
ASG_PER_STATUS_DP	Personal status for the assignment (as of Date Paid)
ASG_POSITION	The employee's position
ASG_POS_DATE_FROM	The date from which this assignment position information is effective
ASG_POS_DATE_TO	The date to which this assignment position information is effective
ASG_POS_END_TIME	The standard end time for the assignment position
ASG_POS_FREQ	The frequency for which the assignment position's hours is measured
ASG_POS_HOURS	The standard number of working hours for the position

Table B – 2

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

Table B – 2

Database item	Description
ASG_SUCCESSOR	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

Table B – 2

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
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

Table B – 3

Contact Information

Database item	Description
CON_AGE	The contact's age
CON_APP_NUMBER	The contact's applicant number
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_OF_BIRTH	The contact's date of birth
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_FIRST_NAME	The contact's first name
CON_FULL_NAME	The contact's full name
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_PERSON_TYPE	The contact's person type – employee or applicant, for example
CON_RELATIONSHIP	The relationship of the contact to the employee
CON_SEX	The contact's sex
CON_START_DATE	The date from which this contact information is effective

Table B – 4

Database item	Description
CON_TITLE	The contact's title
CON_WORK_PHONE	The contact's work telephone number

Table B - 4

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_TERM_ACCEPTED_BY	The person who accepted the employee's notice
EMP_TERM_DATE	The employee's termination date

Table B – 5

Location Details

Database item	Description
LOC_ADR_LINE_1	The first line os 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
LOC_ADR_REGION_3	The third line of the assignment's region
LOC_ADR_PHONE_1	The assignment's first work tele- phone number
LOC_ADR_PHONE_2	The assignment's second work telephone number

Table B – 6

Database item	Description
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

Table B - 6

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)

Table B – 7

Work Address Details (UK only)

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

Table B – 8

Payroll Details

Database item	Description
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

Table B – 9

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_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

Table B – 10

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)

Table B – 11

Home Address Details (UK only)

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

Table B – 12

People Information

Database item	Description
PER_AGE	The person's age
PER_APPLICANT_NUMBER	The person's applicant number
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_VERIFIED	The date the employee last verified his or her personal data
PER_DISABLED	Whether the person is disabled (yes/no)
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_PERSON_TYPE	Type of person – employee or applicant, for example
PER_PREV_LAST_NAME	The person's previous last name
PER_SEND_EXPENSES	Where to send the person's expenses (home/office)
PER_SEX	The person's sex

Table B – 13

Database item	Description
PER_TITLE	The person's title
PER_WORK_PHONE	The person's work telephone number

Table B – 13

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

Table B – 14

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_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

Table B – 15

Date Information

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

Table B - 16

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</ename>	The element's benefit classification
<ename>_CLASSIFICATION</ename>	The element's classification
<ename>_CLOSED_FOR_ENTRY</ename>	Yes/no flag: if yes, the element's entries cannot be modified
<ename>_CLOSED_FOR_ENTRY_CODE</ename>	Yes/no flag: if yes, the element's entries cannot be modified
<ename>_COSTABLE_TYPE</ename>	The element's costable type (from lookup table)
<ename>_COSTABLE_TYPE_CODE</ename>	The element's costable type (code values)
<ename>_COUNT</ename>	The element entry count
<ename>_END_DATE</ename>	The date to which this element is effective
<ename>_INPUT_CURRENCY_CODE</ename>	The element's input currency code
<ename>_LENGTH_OF_SERVICE</ename>	The element's qualifying length of service

Table B - 17

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

Table B – 17

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

Database item	Description
<ename>_<iname>_ENTRY_VALUE</iname></ename>	The element value
<ename>_<iname>_USER_ENTERED_CODE</iname></ename>	Whether a value exists at the element entry level (yes/no)

Table B – 18

Database item	Description
<ename>_<iname>_START_DATE</iname></ename>	The start date of element
<ename>_<iname>_END_DATE</iname></ename>	The end date of element

Table B – 18

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</iname></ename>	The summed element values for the multiple entries

Table B - 19

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</name>	The grade rate's value
GRADE_ <name>_MINIMUM</name>	The grade rate's minimum value
GRADE_ <name>_MAXIMUM</name>	The grade rate's maximum value

Table B - 20

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</name>	The pay scale rates value

Table B - 21

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></segment_name>	People descriptive flexfield database items
PAYROLLS_ <segment_name></segment_name>	Payroll descriptive flexfield database items
ASSIGNMENTS_ <segment_name></segment_name>	Assignment descriptive flexfield database items
GRADES_ <segment_name></segment_name>	Grade descriptive flexfield database items
ABSENCES_ <segment_name></segment_name>	Absence descriptive flexfield database items
ABSENCE_TYPES_ <segment_name></segment_name>	Absence type descriptive flexfield database items
PERSON_ADDRESSES_ <segment_name></segment_name>	Person Address descriptive flexfield database items
EVENTS_ <segment_name></segment_name>	Events descriptive flexfield database items
JOBS_ <segment_name></segment_name>	Jobs descriptive flexfield da- tabase items
CONTACTS_ <segment_name></segment_name>	Contacts descriptive flexfield database items
PERIODS_OF_SERVICE_ <segment_name></segment_name>	Periods of Service descriptive flexfield database items
RECRUITMENT_ACTIVITIES_ <segment_name></segment_name>	Recruitment Activities descriptive flexfield database items
POSITION_ <segment_name></segment_name>	Position descriptive flexfield database items

Table B – 22

Database item	Description	
APPLICATIONS_ <segment_name< th=""><th>Applications descriptive flexfield database items</th></segment_name<>	Applications descriptive flexfield database items	
ORGANIZATION_ <segment_name></segment_name>	Organization descriptive flexfield database items	

Table B – 22

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></segment_name>	Grade key flexfield database items
JOB_KF_ <segment_name></segment_name>	Job key flexfield database items
POS_KF_ <segment_name></segment_name>	Position key flexfield database items
GROUP_KF_ <segment_name></segment_name>	Group key flexfield database items

Table B - 23

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</absence_name>	The cumulative balance for an absence type

Table B - 24

Glossary

Α

- **Absence Types** Categories of absence, such as medical leave or vacation leave, that you define for use in absence windows.
- Alternative Regions Parts of a window that appear in a stack so that only one is visible at any time. You click on the name of the region to pop up a list of the other regions in the stack. Select the name of a region to bring it to the top of the stack.
- **Applicant** A candidate for employment in a Business Group.
- **Appraisal** A 'superset' of recording opinions and setting and achieving objectives, plans and so on. See also: *Assessment*.
- **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.

- **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.
- 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 on 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*

- 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.

C

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.

- 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.
- 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.
- **Career Map** A plan showing the expected routes by which employees can progress from one job to another within the Business Group.
- 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.
- **Compensation** The pay you give to employees, including wages or salary, and bonuses. See also: *Elements*
- 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 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
- **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.
- **Core Competencies** 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*
- Customizable 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 customization.

${f D}$

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*

Deployment Factors See: Work Choices

Descriptive Flexfield A field that your organization can customize to capture additional information required by your business but not otherwise tracked by Oracle Applications. See also: *Key Flexfield*

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*
- **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 customized form, or for distributing costs.
- 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.
- **Event** An activity such as a training day, review, or meeting, for employees or applicants.
- **Expected Week of Confinement (EWC)** 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).

F

- **Field** A view or entry area in a window where you enter, view, update, or delete information. See also: *Block*, *Region*
- **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.
- **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*
- **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.

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

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.

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
- Leaver's Statement 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.
- Linking Interval 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 Linked periods of incapacity for work 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.

Lower Earnings Limit (LEL) 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

Maternity Pay Period 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.

N

NACHA National Automated Clearing House Association. This is the US system for making direct deposit payments to employees.

Nonrecurring Elements Elements that process for one payroll period only unless you make a new entry for an employee. See also: *Recurring Elements*

0

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.

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, which 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.
- 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) 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 then 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 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.

- **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.
- 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*

\mathbf{Q}

- **Qualification Type** An identified qualification method of achieving proficiency in a competence, such as an award, educational qualification, a license or a test. See: *Competence*
- Qualifying Days 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 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.
- QuickCode 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 QuickCode Types, and you can add values to some predefined QuickCode Types.
- 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

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*
- 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*
- **Report Parameters** Inputs you make when submitting a report to control the sorting, formatting, selection, and summarizing of information in the report.
- **Report Security Group** A list of reports and processes that can be submitted by holders of a particular responsibility. See also: *Responsibility*
- **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.

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, Report Security Group

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.

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.

S

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.

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, User Profile Options

SMP See: Statutory Maternity Pay

Special Information Types Categories of personal information, such as skills, that you define in the Personal Analysis key flexfield.

SSP See: Statutory Sick Pay

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*

Statutory Maternity Pay 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.

Statutory Sick Pay 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.

T

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.

U

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

\mathbf{V}

Waiting Days 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 Deployment 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, or at person level.

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

E Α Absence types, database items, A – 37 Element entries skip rules, 1 - 21 Alias statement, 1 - 9, 1 - 45validation, 1 – 23 Arithmetric operators, in formulas, 1 – 36 Element skip rules, 1 – 21 Assignment sets, writing formulas for, 1 - 28 Elements, database items, A – 29 Assignment statement, in formulas, 1 – 4, 1 – Expressions, in formulas, 1 – 35 F B **Formulas** Bulk Compile Formulas process, 1 – 51 context, 1 - 4 errors in, 1 - 51 to 1 - 53for assignment sets, 1 – 28 for element skip rules, 1 – 21 for payroll calculations, 1 – 16 to 1 – 22 Constants, in formulas, 1 – 31 for QuickPaint, 1 - 26 for validation, 1 - 23 to 1 - 25inputs, 1 - 3 to 1 - 5, 1 - 17 to 1 - 19D structure, 1 - 29 types of statement, 1 - 9, 1 - 45Database items, 1 – 33 from element input values, 1 - 18 using arithmetric operators, 1 – 36 using comments, 1 – 8 to 1 – 9, 1 – 45 in formulas, 1 – 4 list of dynamic items, A - 29 using conditions, 1 – 6 to 1 – 8, 1 – 48 list of static items, A – 2 using constants, 1 - 31 using functions, 1 - 36 to 1 - 42Default statement, 1 - 10, 1 - 46

using variables, 1-32 to 1-35 writing efficient formulas, 1-19 to 1-21 Functions in formulas, 1-36 user defined, 1-13

G

Global values, 1 – 4, 1 – 12, 1 – 32

I

If statement, in formulas, 1 – 48 Input values in payroll formuals, 1 – 17 validation, 1 – 23 Inputs statement, 1 – 3, 1 – 46

L

Local variables, 1 - 4, 1 - 32

P

Payroll formulas, 1-16 to 1-22Payroll runs, skip rules, 1-21Processes Create Descriptive Flexfield DB Items, A-34Create Key Flexfield DB Items, A-36

\mathbf{Q}

QuickPaint reports, writing formulas for, 1 - 26

R

Return statement, in formulas, 1 – 50

S

Skip rules, 1-21Statements, in formulas, 1-9 to 1-10, 1-45

U

User tables, validating entries, 1 - 23

\mathbf{V}

Variables, in formulas, 1 – 32

W

WAS DEFAULTED condition, 1 – 6 Windows Database Items, 1 – 11 Define Function, 1 – 13 Formula, 1 – 10 Globals, 1 – 12

Reader's Comment Form

Oracle FastFormula User's Guide A58346-01

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.