

Oracle® Applications

User Interface Standards for Forms-Based Products

Release 12.1

Part No. E12900-02

April 2009

Oracle Applications User Interface Standards for Forms-Based Products, Release 12.1

Part No. E12900-02

Copyright © 1994, 2009, Oracle and/or its affiliates. All rights reserved.

Primary Author: Sara Woodhull, Peter Wallack, Mildred Wang, Maxine Zasowski

Contributing Author: David Lane, Trevor Mathers, Vicki Shimizu, Susan Stratton, Leslie Studdard

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third party content, products or services.

Contents

Send Us Your Comments

Preface

1 Introduction

Introduction to Oracle Applications User Interface Standards for Forms-Based Products	
.....	1-1
User Interface Goals.....	1-1
Runtime Environment.....	1-2
Conventions.....	1-2
Implementing These Standards.....	1-3

2 General Design and Layout

Overview	2-1
Elements of the Interface	2-3
Information Presentation Problems	2-7
Information Presentation Models	2-8
Regions.....	2-8
Single-Record Formats.....	2-9
Multi-Record Formats.....	2-9
Hybrid Formats.....	2-10
Window and Block Relations.....	2-11
Dynamic Layouts.....	2-11
Wizards.....	2-12
Other Design Considerations	2-13
General Layout Rules.....	2-14
Context	2-15

Retrieving Records.....	2-15
Indicating Attributes.....	2-17
Navigation and Function Invocation.....	2-17

3 Global Settings and Objects

Overview	3-1
Visual Attributes	3-1
Property Classes	3-2
Libraries	3-3
Menus	3-3
File Menu.....	3-4
Edit Menu.....	3-5
View Menu.....	3-5
Folder Menu.....	3-6
Tools, Reports and Actions Menus.....	3-7
Window Menu.....	3-7
Help Menu.....	3-8
Pop-up Menus.....	3-9
Toolbar	3-9
Status Bar	3-10
Calendar	3-11
Folder Objects	3-12

4 Container Objects

Overview	4-1
Modules	4-1
Windows	4-4
General Look and Feel.....	4-4
Non-Modal Windows.....	4-6
Find Windows.....	4-10
Modal Windows.....	4-14
Semi-Modal Windows.....	4-16
Canvases	4-17
Content Canvases.....	4-17
Stacked Canvases.....	4-17
Blocks	4-18
General Block Rules.....	4-18
Context Blocks.....	4-19
Dialog Blocks.....	4-21
Single-Record Blocks.....	4-21

Multi-Record Blocks.....	4-23
Combination Blocks.....	4-26
Folder Blocks.....	4-29
Find Blocks.....	4-33
Alternative Blocks.....	4-35
Master-Detail Characteristics.....	4-35
Regions	4-37
Frames in Single-Record Formats.....	4-39
Overflow Regions.....	4-40
Frames in Multi-Record Blocks.....	4-42
Regions that Scroll.....	4-42
Tab Regions.....	4-43

5 Widgets

Overview.....	5-1
General Properties.....	5-2
Text Items.....	5-2
Display Items	5-5
Poplists	5-7
T-Lists	5-9
Option Groups	5-9
Check Boxes	5-12
Buttons	5-14
Lists of Values (LOV)	5-17
Editors	5-19
Descriptive Flexfields	5-19
Key Flexfields.....	5-21

6 Text

Overview.....	6-1
General Properties.....	6-2
Properties of Prompts and Titles.....	6-4
Single-Record Block Prompts	6-5
Multi-Record Block Prompts	6-6
Conventions	6-8
Access Keys (Mnemonics).....	6-10
Using Data as Prompts.....	6-11
Errors	6-12
Warnings.....	6-13
Questions.....	6-13

Information	6-14
Hints	6-14
Messages.....	6-15

7 Runtime Behaviors

Overview.....	7-1
Validation Models.....	7-2
Saving Changes	7-3
Navigation Within Forms	7-4
The Navigator.....	7-8
Linked Forms	7-10
Open Form.....	7-10
Zoom.....	7-10
Call Form.....	7-10
Disabled Functions	7-11
Irreversible Actions	7-13
Running Totals.....	7-13
Field Ranges (From/To Field Pairs).....	7-14
Currency.....	7-16
Multiple-Record Selection	7-18
Highlighting Information.....	7-19
Printing.....	7-20
Long-Running Processes.....	7-21
Ordering of Displayed Records.....	7-21
Record History.....	7-22
About Oracle Applications.....	7-22

A Definitions

Definitions.....	A-1
------------------	-----

B Approved Terminology

Overview.....	B-1
Abbreviations.....	B-1
Applications Product Names.....	B-1
Term Abbreviations.....	B-2
Terms With No Abbreviation.....	B-4
Acronyms.....	B-5

Index

Send Us Your Comments

Oracle Applications User Interface Standards for Forms-Based Products, Release 12.1

Part No. E12900-02

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document. Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

Note: Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the new Applications Release Online Documentation CD available on My Oracle Support and www.oracle.com. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: appsdoc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at www.oracle.com.

Preface

Intended Audience

Welcome to Release 12.1 of the *Oracle Applications User Interface Standards for Forms-Based Products*.

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

- The principles and customary practices of your business area.
- Computer desktop application usage and terminology.

If you have never used Oracle Applications, we suggest you attend one or more of the Oracle Applications training classes available through Oracle University.

See Related Information Sources on page x for more Oracle Applications product information.

Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at <http://www.fcc.gov/cgb/consumerfacts/trs.html>, and a list of phone numbers is available at <http://www.fcc.gov/cgb/dro/trsphonebk.html>.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by

the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

Structure

- 1 Introduction**
- 2 General Design and Layout**
- 3 Global Settings and Objects**
- 4 Container Objects**
- 5 Widgets**
- 6 Text**
- 7 Runtime Behaviors**
- A Definitions**
- B Approved Terminology**

Related Information Sources

This book is included on the Oracle Applications Documentation Library, which is supplied in the Release 12 Media Pack. You can download soft-copy documentation as PDF files from the Oracle Technology Network at <http://otn.oracle.com/documentation>, or you can purchase hard-copy documentation from the Oracle Store at <http://oraclestore.oracle.com>. The Oracle Applications Release 12 Documentation Library contains the latest information, including any documents that have changed significantly between releases. If substantial changes to this book are necessary, a revised version will be made available on the "virtual" documentation library on My Oracle Support (formerly Oracle*MetaLink*).

If this guide refers you to other Oracle Applications documentation, use only the latest Release 12 versions of those guides.

Online Documentation

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

- **Online Help** - Online help patches (HTML) are available on My Oracle Support.
- **PDF Documentation** - See the Oracle Applications Documentation Library for current PDF documentation for your product with each release. The Oracle Applications Documentation Library is also available on My Oracle Support and is updated frequently.
- **Oracle Electronic Technical Reference Manual** - The Oracle Electronic Technical Reference Manual (eTRM) contains database diagrams and a detailed description of database tables, forms, reports, and programs for each Oracle Applications product. This information helps you convert data from your existing applications and integrate Oracle Applications data with non-Oracle applications, and write custom reports for Oracle Applications products. The Oracle eTRM is available on My Oracle Support.

Related Guides

You should have the following related books on hand. Depending on the requirements of your particular installation, you may also need additional manuals or guides.

Oracle Applications Developer's Guide

This guide contains the coding standards followed by the Oracle Applications development staff. It describes the Oracle Application Object Library components needed to implement the Oracle Applications user interface described in the *Oracle Applications User Interface Standards for Forms-Based Products*. It provides information to help you build your custom Oracle Forms Developer forms so that they integrate with Oracle Applications. In addition, this guide has information for customizations in features such as concurrent programs, flexfields, messages, and logging.

Oracle Applications Flexfields Guide

This guide provides flexfields planning, setup, and reference information for the Oracle Applications 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's Guide

This guide explains how to navigate, enter data, query, and run reports using the user interface (UI) of Oracle Applications. This guide also includes information on setting user profiles, as well as running and reviewing concurrent requests.

Integration Repository

The Oracle Integration Repository is a compilation of information about the service endpoints exposed by the Oracle E-Business Suite of applications. It provides a complete catalog of Oracle E-Business Suite's business service interfaces. The tool lets users easily discover and deploy the appropriate business service interface for

integration with any system, application, or business partner.

The Oracle Integration Repository is shipped as part of the E-Business Suite. As your instance is patched, the repository is automatically updated with content appropriate for the precise revisions of interfaces in your environment.

Do Not Use Database Tools to Modify Oracle Applications Data

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

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

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

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

Introduction

Introduction to Oracle Applications User Interface Standards for Forms-Based Products

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

This manual is intended for product designers (including user interface engineers), application developers, quality assurance, and usability engineers. Each plays a critical role in ensuring that the application is built to satisfy the end user's requirements.

The companion document to this manual is the *Oracle Applications Developer's Guide*. It describes the exact coding practices needed to adhere to these guidelines.

User Interface Goals

These standards address the development of a UI built using Oracle Forms, and are designed to meet the needs of a trained professional in a business environment. In contrast, the UI standards for Oracle's Self-Service applications are directed at the novice or infrequent user. (See the Oracle Self-Service Applications manuals.) To best serve the professional user, the Oracle Applications UI seeks to maximize the following aspects of usability:

Productivity

The application should allow the end user to be more productive than with the prior system they were using. It should assist the user to perform their job by presenting actions and information familiar to the intended user of each screen.

Ease of Learning

The application must be easy to learn and present a consistent, familiar and predictable

UI. Novice users should find clear and obvious methods while experts should be able to increase their efficiency as they learn the more powerful features of the system.

The "User Experience"

The application should be inviting to use. It should reward a user by providing them the information and tools necessary to complete tasks rapidly, and it should guide them where appropriate with timely and meaningful feedback. The application should allow exploration without the fear of irreversible consequences.

Runtime Environment

With a few exceptions, all Oracle Applications products must run in the following environments:

- Color Monitor, with greater than 256 colors
- Screen resolution of at least 1024 x 768 pixels

For information on specific versions of browsers and Java Runtime Environments (JRE), consult the *Installing Oracle Applications* manual.

Note that Oracle Applications do not support a client/server environment; they only operate in an internet configuration.

Conventions

Translation-related Standards

For ease of use, this book indicates which standards are necessary specifically for translation purposes by including the italicized and bolded word **[*Translation*]** at the beginning of the standard. Designers who are creating applications that will not run in other countries can use this indication to identify standards they may choose to ignore for their applications. It may be a good idea, however, to read those standards in case the applications become multilingual in the future.

Standards vs. Guidelines

This book presents standards that must be followed in order to conform to Oracle Applications specifications, as well as guidelines that have some flexibility. Although the guidelines are not required, adhering to them will usually result in a better user interface.

Throughout this book, each International Standardization Organization (ISO) standard is indicated by a number next to the text describing the standard. When the number (preceded by the letters "OMS", for example, "OMS-00000") is shown next to the text, that text then represents the official standard. When the number appears in parentheses in the course of the text, this indicates a reference to the official standard. To find a

particular standard, you may refer to the ISO standards index included in addition to the regular index.

Text that is not labeled as a standard and does not reference a standard is considered a guideline.

Implementing These Standards

Documentation on implementing these standards using Oracle Forms and the libraries provided by Oracle Application Object Library is provided in the *Oracle Applications Developer's Guide*. Accordingly, there are frequent references to the *Oracle Applications Developer's Guide* throughout this manual.

The Oracle Application Object Library code designed to support these standards assumes that they will be followed without exception. No attempt has been made to analyze the impact of deviations from these standards, therefore Oracle strongly recommends that these standards never be violated if used with Oracle Application Object Library.

General Design and Layout

Overview

Before an effective user interface can be developed for a product, thorough requirements analysis is required. This analysis should include identification of business flows and typical task flows, development of user profiles, and studies of the user's environment and system.

This manual does not address methods and details of requirements analysis except to note that any software product is only as good as the ability of the user to operate it, and that the goal of creating a usable system can only be met by working with and listening to the intended users. This chapter lists several issues to consider when designing a product, and further suggests that the developer research and follow the principles of user-centered design whenever possible.

The remainder of the chapter discusses translation issues and fundamental information presentation problems and how the product designer can provide solutions to these issues.

The following topics are covered:

- Design analysis
- Elements of the interface, including a brief overview of interface elements such as windows, menus, fields, and LOV's
- Information presentation problems
- Information presentation models, including information on regions, single-record formats, multi-record formats, hybrid formats, window and block relations, dynamic layouts, and wizards
- Other design considerations, including guidelines regarding general layout and querying records

- Accessibility features for users who are visually or physically challenged

Design Analysis

For every task that a product is designed to support, build a flowchart of the steps necessary to perform the task. Identifying all the aspects of a particular task will reveal the opportunities for optimization. Once the flow of a task is documented, consider the following issues to lead you to the proper user interface for the screen(s) designed to support that task:

- What related information is needed to complete the task? What amount of information does the user need to ignore?
- What is the frequency of use and volume of data for the screen?
- What widgets are most appropriate? For example, if the screen will be used by high-speed data entry clerks, widgets requiring the mouse are inappropriate.
- What level of training is expected of the user performing the task? Does the screen need to be optimized for first-time or infrequent use (activities which may be better included in a self-serve application), or for a highly trained clerk doing repetitive tasks?
- What if a user makes a mistake at a particular step, or attempts to bypass a step? Must the task steps be done sequentially, or can they be done in parallel?
- What decisions is the user required to make along the way? What if those decisions are only in exceptional cases?
- What other tools might the user be familiar with? What are their expectations of your product based on these other tools?

Designing for Portability

Restrictions imposed to support multiple environments and languages must be evaluated early in the design phase. Any decision to deviate from these restrictions must be considered carefully.

Web Browser Environment

Oracle Forms does not support all window and mouse triggers such as WHEN-MOUSE-ENTER and WHEN-MOUSE-MOVE. Products must be designed to provide alternative methods to invoke the functionality that such triggers afforded.

Application Server Environment

Extreme caution must be exercised when relying on operating system specific

functionality such as OLE automation, VBX controls, or hosting commands to the operating system. Besides being non-portable, they may produce an awkward result, such as launching a window that will appear on the forms server machine rather than the user's machine.

Screen Characteristics

The maximum allowed window size is 10.3" (width) by 6.25" (height). This maximum size comes from the requirement to operate with 1024x768 screen resolutions.

(OMS-71002) To support various screen resolutions on bitmap monitors, forms should be built with coordinate systems based on logical, not physical, measures. For that reason, lay out all screens in inches rather than pixels.

Translation

[Translation] Oracle products run in the native language in all countries. Therefore, all prompts, titles, messages, and data presented to the user, other than data they enter while using the product, must be translatable so that it is presented in their native language.

For more information on expansion requirements, see: General Properties [for Text], page 6-1

Anything that is translated must have sufficient space available to expand when translated from English to other languages (assuming that English is the base language for development). Depending on the widget and its placement, this space must be available either to the left, right, or both sides.

Numeric data and dates must also be presented to the user in their proper format for the current language. For example, numbers in German use "," as the radix character, whereas in English they use ".".

Elements of the Interface

The following are the basic elements to use when designing screens. Each is discussed in much greater detail in later chapters of this document.

Windows

Windows or multiple document interface (MDI) windows are frames in which information is presented. Except for the simplest of forms, most forms have several windows associated with them. There are several recommendations for what information is contained in each window, specific behaviors of how and where windows open at runtime, and usage rules for modal windows that constrain a user to act only within that window.

Menus

A menu is a list of actions from which the user can choose. A pull-down menu is associated with each form, but always displays in the MDI window. There are specific rules for dynamically enabling and disabling menu entries. Oracle Applications use a generic menu for all products to invoke standard functions. This menu also includes three special menus for product-specific functions that can be enabled and altered at runtime. Right-mouse pop-up menus are also available in all text fields.

The Toolbar

The toolbar is a set of iconic buttons that allow quick access to common functions that are also on the menu. It is attached to the MDI window immediately below the pull-down menu.

Canvases

Canvases are surfaces or areas within windows on which objects are displayed. A window can show one or more canvases, either one at a time or simultaneously. The choice of canvas type, and placement, is based on the information presentation model chosen.

Blocks

A block is the logical representation of a set of related data items referred to as a database entity. In Oracle Forms, interface items are assigned to a block. The block controls the consistency of the data which is posted to the database by requiring that all items of a record in the block be completed before the user can continue. The items of a block may be distributed among several canvases or windows, but usually this logical grouping is also reflected in the design, maintaining the relationship of the block to the window. Blocks may show more than one record of a database entity at a time, but within that structure the integrity of each record is maintained by the Oracle Forms logic.

Regions

Regions are logical groupings of fields which are generally designated by an outline and a title. The specific cosmetics used to indicate regions are discussed in Chapter 3. Tabbed regions are a specific type of region that uses the Tab control.

Master-Detail Relations

A master-detail relation between two blocks controls the behavior of the detail block based on the current state of the master record. Oracle Forms will automatically ask the user to save changes if any are pending in a detail block if an attempt is made to switch

the master record. Additionally, Oracle Forms will prevent querying in the detail block if the master is not yet saved, and entry in the detail block if no master row exists. There are specific behaviors for the timing of querying child records, based on the visibility of the detail block, that form logic must account for.

Text Items

Text items allow a user to type alphanumeric characters. They can be shown in several states, including active, inactive, and display-only. These states are color-coded automatically. Often a text item has a list of values (LOV) associated with it to assist the user in locating a valid value.

Lists of Values (LOV)

An LOV is a set of valid values for a field. It is associated with a text item both for validation and to assist the user in selecting a value by displaying the list in a pop-up window at the user's request.

Buttons

Buttons are used to invoke actions and to navigate. For example, they can be used to begin the "Approve" function, or to move to the "Order Lines" entity. Usually buttons pertain to a block, but occasionally they pertain to a field or a window. There are specific guidelines for placement, size, and labeling.

Check Boxes

Check boxes are used to indicate a two-state value. They should only be used when the label on the check box can be clearly thought of as having true (checked) and false (unchecked) states.

Lists

List items allow the user to select a value from a small set of choices (the recommended maximum is 15 entries). A list is not to be confused with an LOV; it is a simple widget which does not support the volume of data or autoreduction capabilities of an LOV. Two types of lists are available: Poplists (commonly called a "drop-down list") and T-Lists (commonly called "scroll lists" or "list boxes").

Images

Image fields support the display of images in various formats, with the image source retrieved from either the database or from the file system.

Radio Groups

Radio groups (sometimes called option groups) allow the selection of one value from a choice of several, represented by several radio buttons.

Scroll Bars

There are three types of scroll bars that are used:

- Block scroll bars allow the user to move through the records of a block by dragging the scroll box; all multi-record blocks use a vertical block scroll bar, placed to the right of the block (OMS-73078).
- Multi-line text item scroll bars allow scrolling within a multi-line text item; all multi-line text items use a scrollbar. (OMS-74007).
- Canvas scrollbars are used when a canvas cannot be made to fit entirely in the window and scrolling is required.

Messages

Messages, often referred to as "alerts," are shown in pop-up windows, to which the user must respond. Different styles of pop-ups and required responses are discussed, as well as standards for verbiage they contain.

Descriptive and Key Flexfields

The term "flexfield" refers to a declaratively controlled mechanism which allows the administrator of the application to define custom groups of data items which are collected or shown as part of the interface.

Oracle Applications use two types of flexfields, *key flexfields* and *descriptive flexfields*. A key flexfield is used to enter multi-segment values such as part numbers, account numbers, and so on. A descriptive flexfield allows entry of additional information where the product has not already provided a field.

Status Bar

The Status Bar contains the message line and other Oracle Forms status indicators, such as List of Values. It is displayed at the bottom of the MDI window.

Related Topics

Windows, page 4-4

Pull-down Menus, page 3-3

Pop-up Menus , page 3-9
Toolbar, page 3-9
Canvases, page 4-17
Blocks, page 4-18
Regions, page 4-37
Master-Detail Characteristics, page 4-35
Text Items, page 5-2
Lists of Values (LOV), page 5-17
Buttons, page 5-14
Check boxes, page 5-12
Poplists, page 5-7
T-Lists, page 5-9
Radio Groups, page 5-9
Multi-Record Blocks, page 4-23
Regions that Scroll, page 4-42
Messages: Errors, page 6-12
Messages: Warnings, page 6-13
Messages: Questions, page 6-13
Messages: Information, page 6-14
Descriptive Flexfields, page 5-19
Key Flexfields, page 5-21

Information Presentation Problems

The fundamental problems when designing a user interface are how to present information to the user and how to allow the user to access the information they need. This section discusses some high-level issues that must be considered while optimizing windows for a task, and presents models that attempt to resolve these issues. Each screen must be designed by carefully analyzing and applying the available presentation models, and tuning them as required to meet the needs of the specific task.

The User's Perception of Information

Window layouts should be based on the logical relation of data, not the physical database tables. For example, in the sales and marketing product, a user perceives an "Account" as the specific data of the account, as well as the "Contacts" and "Executives" of that account. These three entities are stored in separate physical tables, but by placing

them all in a single window, the user continues to perceive them as one logical object.

Summary Versus Detail

Placing information in a window always involves a compromise between the presentation of many records of an entity and the display of many details about each record. A summary view has the advantage of showing more records of an entity, but at the loss of detailed information about each record. A careful analysis must be done to determine the best compromise of rows and detail for each function. Often a hybrid of the two views is the best answer.

See: Hybrid Formats, page 2-10

Avoiding Information Clutter

Many of the Oracle Applications products contain large and complex data structures. However, even large amounts of information can be organized in a way which allows the user easy access to what they need. If this same information is not well organized, it simply becomes clutter. At the very least, data should be grouped and organized in such a way as to help the user quickly identify the general location of the information they need, and then allow them to quickly locate that particular item without having to visually scan an entire screen - or worse, search from page to page.

Information Visibility

An analysis of the task flows should reveal what information a user must have access to at each point in time. For example, if a user must see "Set of Books" information to enter a journal, a decision must be made as to whether the Set of Books information must be visible at all times, or the user merely needs the ability to see it upon request. This in turn impacts how much of each entity should be displayed at once, and how many windows are necessary. A good principle to follow is to present information in decreasing order of importance, so that the most commonly needed data is readily visible first. Then, design the presentation model to allow the user to easily and selectively drill down to greater levels of detail.

Information Presentation Models

This section contains various methods available for laying out windows. The models presented here are guidelines which have been successful for common tasks. Alternative layouts can be used to better accommodate unique tasks.

Regions

A region is a visual grouping of logically related fields. All large entities should be broken down into two or more regions, so that information can be presented to the user in organized sets of fields. Titles make the groupings easier to identify; however, you

are not required to use these identifiers if the relationship of the fields in the group is obvious. There are some situations though, where a title is always required; these are discussed in the detailed section on Regions.

After an entity is divided into groups, if all groups cannot be displayed simultaneously, it becomes necessary for two or more regions to be displayed in the exact same space, one at a time, using a Tab control. The user can jump to a specific region by selecting a tab or by progressing continuously through all fields of all Tab regions. A set of fields, minimally the primary key fields, should remain visible so the user can maintain context at all times. There, "frozen" fields must be placed inside the Tab control on the left side.

When best to use Tab regions:

- When the user must see all fields of a record in a single window, but they do not need to see them all simultaneously.
- When entire regions need to be hidden from a user due to security restrictions, the current state of data, or other product installations.

Scrolling of regions is not desirable since it may leave the user laboriously switching to and scrolling through each region searching for a field. A region should only scroll when the logical grouping of fields results in a field set that simply cannot be shown simultaneously, or when the fields that are initially out of view are seldom used. Care should be taken to avoid this unless there is no other possible user interface alternative.

If it must scroll, a region should never scroll more than two times the width (in multi-record blocks) or height (in single-record blocks) of the viewport (OMS-73122).

Single-Record Formats

Single-record formats allow the maximum number of fields for a single record to be displayed at once. In general, single-record formats should be used for any of the following cases:

- There is only one record possible.
- The user commonly works with only one record at a time.
- The user must see many attributes of one record at the same time.

Multi-Record Formats

Multi-record formats allow the maximum number of records for a single entity to be displayed at once. In general, multi-record formats should be used for any of the following cases:

- When the user must see multiple records to perform the transaction, typically because there is a relationship between the records.

- When the user must see summary attributes of many rows at the same time, typically to scan for information quickly.
- When the user normally perceives of the entity in a multi-record format (for example, the lines of a requisition).
- When the relevant data in a record will easily fit on one line.
- When the format is meant to convey that more than one record can be entered.

Hybrid Formats

Often a combination of single and multi-record formats is the best, allowing both summary and detail views of the same record.

Overflow Regions

When information is presented in a multi-record format, but additional detail about the current record is displayed in a single record format in the same window, the additional detail is referred to as an "overflow region." The fields of the overflow region should be non-navigable to avoid unexpected navigation sequences. They may be enterable if they are not applicable to all records and thus cannot be displayed in the multi-record block, or have unique sizing requirements, such as multi-line text items. Fields within overflow regions may be queryable even if they are display-only.

The overflow fields are always shown in the same window as the multi-record block and below it, ideally separated by a gap. Generally, they are on the same tab region as the multi-record block they belong with (OMS-73118).

When best to use:

- When the user must see data in a multi-record format, but also needs to see a small amount of additional detail about the current record.
- When the fields associated with each record vary based on some context of the record itself.
- When records include multi-line text items, the multi-line items may be displayed in the overflow region.

Combination Blocks

In a combination block, the user can view the data in a multi-record (summary) format, but can also view a single-record (detail) format to see complete information about the current record in a separate window. There are specific behaviors governing which format is shown after a query, and what occurs when the cursor moves to a new record, as well as how the windows are positioned relative to one another. Ideally, fields in either format should allow update, but only allowing update in the single-record format

is acceptable. Fields in each format may be presented in differing sequence if appropriate.

A specific implementation of this style is the "gateway", which is the opening screens a user sees when navigating to a complex entity, such as a Purchase Order form. The gateway has a summary window and a detail window, as well as a find window, and standard textual buttons (New and Open) in the summary screen. Details regarding the behavior of gateways are discussed in the section Combination Blocks, page 4-26.

Although this technique is the most costly to develop, it is the most flexible for the user.

It is best to use the combination block layout for frequently used entities, where a multi-record format is most appropriate for some tasks and a single-record format is best for others, and each needs to be available.

Window and Block Relations

Each window should contain a logical entity, often requiring that more than one block be shown in it. Although it is preferable to show an entity in one window, there are occasions when splitting it across two or more windows is appropriate.

Displaying several blocks in one window

If all blocks for an entity can be displayed in the desired format in a single window, and adhere to window size and other standards, then using a single window is the preferred method. Typically, master blocks are displayed above detail blocks, but this is not required if the meaning and relationship of each block is clear. A zero-height frame may be used to identify each block if the content of the blocks is not obvious.

When the blocks cannot all be displayed at once, then tabs are employed. Selecting a tab causes the appropriate block to display. Each set of blocks on a given form may have their own tab region, and each tab control may contain one or more blocks.

Displaying one block in several windows

Combination blocks require that the different formats of the same block be shown in different windows. Additional times when this is acceptable are as follows:

- As a variant to overflow regions when there are additional fields about the current record that a user may want to see, but only on request (thus the number of rows in the multi-record block is not always compromised).
- When the user doesn't necessarily perceive some fields as part of the same block. For example, the credit card information region of a sales order is part of the header block, but is not necessarily perceived as additional attributes of the header.

Dynamic Layouts

Because of the wide variety of customers that Oracle Applications serves, many

attributes are not appropriate for all customers. There are several mechanisms to handle this:

Profiles or Setup Rules

The screen layout can be dynamically modified based on the value of a particular profile, product installation, or setup parameter. Typically, this is limited to showing or hiding different sets of fields, but may be as elaborate as presenting different blocks or flows to the user. This approach may be used with any of the information presentation techniques discussed previously.

When best to use:

- When an attribute appears in multiple forms, and a single point of control is necessary to establish its behavior. For example, if a customer is not using Encumbrance accounting, they should be able to declare that once, and have all related fields on all screens respond appropriately.
- When, based on the setup of a product, known optimal combinations of layouts can be inferred. For example, if a customer uses serial number control, the Serial Number field should be presented more prominently.

Folders

Folders allow each user to change which attributes are on a particular screen, as well as their order, size and other behavioral characteristics. This ability to customize means that a user can tailor the screen to show just the information they are interested in.

When best to use:

- All gateways use folders in the results block (OMS-73093).
- If a product expects that the default layout of a particular block is not optimal for most customers, and customization would substantially improve the usability of the screen.

Note: Folders may be coded only within the Oracle Applications division: The required APIs have not been productized and are not supported by Oracle Support.

Wizards

Wizards are a series of screens that guide the user through a particular task. Each screen typically has few fields for the user to interact with. Many times just one field is used. In addition to the field or fields, detailed instructions are provided for the user.

Wizards should be available for all complex tasks that can be performed in a serial order, especially if they will be performed infrequently. Wizard windows should follow

the guidelines for modal windows in general, but must also include the following buttons:

Cancel	Aborts all work performed in the Wizard, reverting data to its prior state. If significant work may be lost then a confirmation dialog should open.
Next	Advances the Wizard to the next step. Shown on all but the last window.
Back	Returns the user to the previous window in the wizard. This button appears on all but the first window.
Finish	Used to indicate that all steps have been completed. Always appears on the last window in place of the Next button. This button may appear on other windows within the wizard, if all of the steps are not required. Pressing "Finish" should save the changes the user has entered, unless this would be considered an unexpected or undesirable behavior.
Help	Invokes the online help system to provide the user help on the current page.

Related Topics

Regions, page 4-37

Tab Regions, page 4-43

Regions That Scroll, page 4-42

Single-Record Blocks, page 4-21

Regions in Single-Record Formats, page 4-39

Multi-Record Blocks, page 4-23

Regions in Multi-Record Blocks, page 4-42

Overflow Regions, page 4-40

Combination Blocks, page 4-26

Folder Blocks, page 4-29

Other Design Considerations

The following topics also affect your form designs:

- General Layout Rules, page 2-14

- Context, page 2-15
- Retrieving Records, page 2-15
- Indicating Attributes, page 2-17
- Navigation and Function Invocation, page 2-17

General Layout Rules

The following rules apply to all screens. Specific exceptions are noted throughout this document.

- Orient screen layouts based on a top-to-bottom task sequence. Blocks, regions, and items are arranged by order of precedence, from left-to-right, then top-to-bottom. (OMS-71005)
- Do not waste screen space. Make windows only as large as is necessary. Use blank space as a way to group information.
- Most widgets are 0.25" high, and multiples of 0.1" wide. Two-dimensional widgets (multi-line text items, T-Lists) are multiples of 0.25" high. Textual buttons are 0.3" high. (OMS-74044)
- In single-record blocks, items should be left aligned to form margins or columns where possible. (OMS-73174)
- In multi-record blocks items should be stacked horizontally and aligned along their top. (OMS-73180)
- Place titles and prompts toward the top or left of the element they are describing.
Prompts in single-record blocks are always to the left of the field. (OMS-75009)
Prompts in multi-record blocks are above the field, and aligned similarly to the data within the field they describe. (OMS-75109, OMS-75011)
Exceptional cases exist for matrix-style layouts, fields that do not require prompts, connecting prompts, and multi-line prompts.
- At least a one-character space should be left between the items on a canvas and the window frame. (OMS-75008)
It is also preferable to leave this space around the edge of Tab regions. Exceptional cases exist for regions used to denote blocks. (OMS-73007)
- Make data stand out. Also, make the controls to access and identify that data intuitive but non-obtrusive.

- Use the same widget for the same attribute in all windows. There are specific exceptions to this rule documented in the section on Widgets, page 5-1.
- When displaying currency (such as US Dollars) and amount fields, display the currency field first.
- When displaying unit of measure and quantity fields together, display the unit of measure field first.

Context

A user must always be able to easily identify the current record(s) they are working with. This is done in a variety of ways:

- The primary key, or other identifying information, of the topmost master record of a form is repeated in the title of all non-modal detail windows. For example, in a form used to enter Sales Orders, the current sales order number should be displayed in the window titles for other windows of that form that directly relate to that sales order (OMS-58505).
- Context of master records, other than the topmost master record, is drawn above any detail blocks that do not occupy the same window as the master. For example, in the Shipments window of the Sales Orders form, the current sales order number is displayed in the window title, but the current order line is shown as context fields drawn within the shipments window (OMS-73072).
- All multi-record blocks include a current record indicator field, to clearly point out the current record (OMS-73178).

Retrieving Records

Oracle Applications provide three methods to locate previously entered data. When using Query by Example, the same block serves as the location where a user enters search criteria and the application displays any retrieved record(s) when the query is run. When using View Find, a separate window appears to allow the user to enter search criteria. "Direct Entry" blocks are a special case to locate a master record.

All blocks that are queryable should respond to View Find with either a Find window or a Row LOV as described below. (OMS- 71105)

View Find (also known as Query Find)

When the user invokes View Find, one of two types of windows will appear. Because the user must first retrieve records before continuing, the Find window automatically appears upon navigating to a gateway or to any query-only block that does not autoquery. View Find can be invoked from the toolbar or menu at any time while in a queryable block. The two types of windows are:

1. A Find window that allows the user to enter search criteria for more than one attribute, then initiate the search. Most fields within a Find window provide LOV validation.

When a Find window should be used (OMS-73645):

- For a multi-record block or combination block that does not autoquery (automatically queries and displays data).
- For allowing the user to find a subset of records in a detail block.
- For any single-record or multi-record block where the user needs to search by criteria other than the primary key (for example, if the primary key is relatively obscure). For single-record blocks, though, the results of the search must be reduced to a single-record (using an LOV after the Find Window if necessary) before being displayed.

2. A "Row LOV" that shows all possible records the user can query within an LOV window.

When a Row LOV should be used (OMS-73557):

- For a single-row block display.
- For allowing the user to select a single record in a multi-record block or combination block that autoqueries. In general, the types of multi-record or combination blocks that autoquery include detail blocks, and master blocks where the number of records is small.

Query by Example (QBE)

QBE mode is the native Oracle Forms mechanism for querying data, and allows complex queries. All blocks that allow the user to retrieve data support this. QBE is a power-user feature and can only be invoked from the menu or keyboard (it is not on the toolbar, and no product buttons should ever be coded to invoke it). The user should never be required to use this mode as the only way to locate data.

Note: LOVs are not always available in QBE mode. However, the LOV indicator in the status bar will still display as the user navigates through the fields, which can be somewhat misleading. If the user persists and activates an LOV by keystroke, and the LOV has not been specially coded to be appropriately active in the QBE, then an error message will be shown.

View Find and QBE operate exclusively of each other. In other words, criteria entered in View Find does not affect criteria entered in QBE, and vice versa.

Direct Entry

(OMS-71018) A screen that requires a master record to be identified, but only allows entry or maintenance of the corresponding detail records, does not need to provide View Find or QBE. Instead, direct entry into the primary key fields of the master can be used to uniquely specify the master record.

Indicating Attributes

The following characteristics of fields, records, and blocks are indicated:

- Text and Display items that act as fields (as opposed to prompts) are displayed with an inset bevel.
- Text items that cannot be edited are displayed in gray; editable text items have a white background if optional, or yellow background if required (OMS-74014, OMS-74130).
- Poplist items are always gray.
- Fields that respond to the LOV function show the LOV indicator in the status bar, and have the LOV icon enabled on the field when it has focus. (OMS-71019)
- Text items always show the Edit indicator in the status bar, and have the Edit Field entries enabled in the menu and toolbar. (OMS-71020)
- Entries on the menu and toolbar are disabled if the function is not available for the current field, record, or block. (OMS-71021)

Navigation and Function Invocation

Several elements are used to facilitate movement among the screens and blocks of a product, as well as to invoke product-specific functions:

The Navigator

The Navigator (or "Navigate Window") is the primary means for opening another form while in an application. The Navigate window is always available during an Oracle Applications session.

Buttons

Buttons are used to invoke product-specific functions, and to facilitate navigation through the various windows of a form (often drilling down to detail blocks). In general, buttons are placed at the bottom of a window. Buttons that pertain to a specific field, or must be pressed in logical sequence among the fields, should be placed near the appropriate field.

For complex forms, frequently there will be too many buttons than would reasonably fit on the screen. Avoid creating more than one row of buttons. If necessary, a button may invoke another window in which to further qualify the action (either an LOV or a specific modal window), or the action should be invoked by a different mechanism (such as the Tools menu).

Within each window, the button a user is most likely to press is made the Default button (OMS-74039). The default button may represent a child block, a function, or a "drill-down" action.

Tab Controls

Tabs are used to switch between regions occupying the same window space. They allow the user to select a particular region or group, and they also serve as the title of that group.

Query Control Poplists

A poplist may be used to control the set of records queried in a particular block. In this case it is referred to as a Query Control Poplist.

Next/Previous Block

Next Block is the mechanism for drilling down the hierarchy of a complex object. When a block has more than one detail block, Next Block moves to the most commonly used block (OMS-76016). Previous Block always moves to the previous block in the hierarchy, not necessarily the actual path taken. If sibling blocks exist in the same window, Next Block and Previous Block must move between them; this is to ensure that all blocks are accessible with the keyboard. These functions are only available by function keys.

Drill Down

Detail screens can be accessed in a variety of ways. As described previously, buttons and Next Block may be used to drill down through a complex object. In multi-record blocks that support drill down, the user can click on the current record indicator to perform this function (OMS-73080). In addition, fields that allow drill down are indicated by green underlined text; the link can be activated by using the right-mouse pop-up menu.

[Tab] (Tab key on the keyboard)

[Tab] moves the cursor to the next item in the current block. When invoked from the last item of the block it may move to one of the following:

- The first item of the same record (OMS-58140).
- The first item of the next record (OMS-58527).

- The first item of the next block (OMS-58140).

[Shift][Tab] moves the cursor to the previous item in the block.

Product Specific Menus and Toolbar

Product-specific functions that apply to all or most windows of a form may be placed in one of the optional pull-down menu entries on the toolbar. In addition to the standard menu entries (File, Edit, etc.), these additional menus may be added: "Tools," "Actions," and "Reports." The menus labeled "Actions" and "Reports" may be relabeled by the specific form. Specific functions found on these menus may also be replicated as textual buttons within windows to increase their visibility and accessibility. Up to 15 entries may be added to each menu.

Accessibility

To meet the needs of users who are physically or visually challenged, Oracle Applications are designed as follows:

- The product is completely operable without a mouse. Every item can take focus and/or be activated with a keyboard mnemonic (access key).
- All screen text is available to screen readers. Fields which cannot take focus are presented within a Prompt/Value List of Values window, which can be interpreted by a screen reader.
- Hidden text, which does not render on the screen but is presented to a screen reader, is available on certain fields which typically would not have a prompt.
- Any use of color coding must only be used to augment. For example, a financials total may be displayed in red if negative, but should also be drawn such that it is understandable without the aid of color (such as with a negative sign preceding the value) (OMS-76046)

Related Topics

Navigation and Function Invocation, page 2-17

Blocks, page 4-18

Widgets, page 5-1

Global Settings and Objects

Overview

This section describes standard settings and objects that exist in every form. Some of these objects are building blocks that a developer must use (such as Visual Attributes and Property Classes); others are fully encapsulated objects in themselves that require little or no additional work by the developer (such as the Toolbar and Menu).

The coding standards discuss how these objects get placed into each form, and additional logic that is necessary to fully utilize them (see the *Oracle Applications Developer's Guide* for more information).

The following topics are covered in this chapter:

- Visual Attributes, page 3-1
- Property Classes, page 3-2
- Libraries, page 3-3
- Menus, page 3-3
- Toolbar, page 3-9
- Status Bar, page 3-10
- Calendar, page 3-11
- Folder Objects, page 3-12

Visual Attributes

The Application Object Library includes predefined visual attributes which are used in every form.

Visual attributes which are provided by the Application Object Library include colors and text styles which produce the Oracle look and feel. This look is generally characterized by dark canvases and gray tabs. The following items receive visual attributes from the Application Object Library:

- Prompts
- Titles
- Canvases
- Check boxes
- Radio Groups
- Textual buttons
- Iconic buttons
- Poplists
- Text Items
- Selected Record attributes
- Current Record Indicator

In many cases the attributes are specified as automatic, which causes Oracle Forms to automatically select the proper color.

See the *Oracle Applications Developer's Guide* for details regarding these and other visual attributes.

Property Classes

Property Classes are sets of attributes that can be applied to every object. The Application Object Library property classes enforce standard cosmetic appearances and behaviors for all Oracle Forms objects, such as:

- All enterable fields are 0.25" high, and have a lowered bevel (OMS-74009).
- All prompts use Helvetica 10 point medium weight.
- All LOVs have Autoselect and Autoposition set to Yes.

Every item in Oracle Forms has a corresponding Property Class that must be applied to it.

For more information on Property Classes, see the *Oracle Applications Developer's Guide*.

Libraries

Libraries contain reusable code for forms. They support these standards by allowing the exact same code to be used for all forms to enforce specific validation, navigation, behaviors and cosmetic appearances. The libraries that are available with Oracle Application Object Library are listed in the following table:

Library Name	When Used?	Library Contents
FNDSQF	Always	Routines for message dictionary, profiles, flexfields, and various other utilities
APPCORE	Always	Packages specifically designed to implement these standards, and various other utilities, such as multi-select and Export
APPDAYPK	For any form that contains date fields	Packages to implement the calendar, used on most date fields
APPFLDR	For any form with Folder blocks	Reserved for use by Oracle only.
CUSTOM	For writing Zooms or other custom code	NULL routines until customized

Every form requires several standard triggers and procedures to link the form with the library. Many of these triggers and libraries have a default behavior that a developer overrides for specific items or blocks. The *Oracle Applications Developer's Guide* discusses the use of these libraries in detail.

Menus

The Oracle Applications pull-down menu includes the following menus:

- File, page 3-4
- Edit, page 3-5
- View, page 3-5

- Folder, page 3-6
- Tools, page 3-7
- Reports (optional), page 3-7
- Actions (optional), page 3-7
- Window, page 3-7
- Help, page 3-8

File Menu

The File menu lets you perform several actions in Oracle Applications:

New	Opens a new record in the active form.
Open	Opens the detail screen for the current selection. Typically only used with the Tree item.
Save	Saves any pending changes in the active form.
Save and Proceed	Saves any pending changes in the active form and advances to the next record.
Next Step	Updates the Process workflow in the Navigator by advancing to the next step in the process.
Export	Exports information in the active block to a tab-separated file which can be opened in a browser or spreadsheet.
Place on Navigator	Creates an Icon in the Documents tab of the Navigator which can be used to recall the active form and its current record.
Log on as a Different User...	Exits the application and displays the Oracle Applications Logon window.
Switch Responsibility...	Closes all active windows and displays the Responsibility window.
Print...	Prints your current window. An application may override this action to allow printing of one or more specific reports instead.
Close Form	Closes all windows of the current form.

Exit Oracle Applications

Quits Oracle Applications.

Edit Menu

Use the Edit menu to edit your data in Oracle Applications:

Undo Typing

Returns the field to the value it had when it gained focus.

Cut

Cuts the current selection to the clipboard.

Copy

Copies the current selection to the clipboard.

Paste

Pastes the contents of the clipboard into the current field.

Duplicate

- Record Above - Copies all values from the prior record to the current record.
- Field Above - Copies the value of the current field from the prior row.

Clear

- Record - Erases the current record from the window.
- Field - Clears the data from the current field.
- Block - Erases all records from the current block.
- Form - Erases any pending changes from the current form.

Delete

Deletes the current record from the database.

Select All

Selects all records (for blocks with multi-select).

Deselect All

Deselects all selected records except for the current record (for blocks with multi-select).

Edit Field...

Displays the Editor window for the current field.

Preferences

- Change Password... - Displays the Change Password dialog box.
- Profiles - Displays the Profiles form.

View Menu

The View menu provides you with the following options:

Show Navigator	Displays the Navigator window.
Zoom	Invokes custom defined zooms.
Find...	Displays the Find window to retrieve records.
Find All	Retrieves all records.
Query by Example	<ul style="list-style-type: none"> • Enter - Invokes Enter Query mode to enter search criteria for a query by example. • Run - Executes the query by example. • Cancel - Cancels the query by example by exiting from Enter Query mode. • Show Last Criteria - Recovers the search criteria used in the previous query by example. • Count Matching Records - Counts the number of records that would be retrieved if you ran the current query by example.
Record	<ul style="list-style-type: none"> • First - Moves the cursor to the first record. • Last - Moves the cursor to the last record.
Translations...	Displays the Translations window.
Attachments...	Displays the Attachments window.
Summary/Detail	Switches between the summary and detail views of a combination block.
Requests	Shows the Request form.

Folder Menu

The Folder menu lets you customize the presentation of data in a folder:

New...	Creates a new folder definition.
Open...	Opens an existing folder definition.
Save	Saves changes to the current folder definition.
Save As...	Saves current folder definition to a new name.

Delete...	Deletes an existing folder definition.
Show Field...	Displays a field that is currently hidden.
Hide Field	Hides the current field.
Move Right	Moves the current field to the right.
Move Left	Moves the current field to the left.
Move Up	Moves the current field up.
Move Down	Moves the current field down.
Widen Field	Increases the width of the current field.
Shrink Field	Decreases the width of the current field.
Change Prompt...	Changes the prompt for the current field.
Autosize All...	Adjusts all field widths to completely show their data.
Sort Data...	Specifies sort order for the first three fields by ascending, descending and no ordering.
View Query...	Displays the query criteria for the current folder definition.
Reset Query	Erases the current query criteria.
Folder Tools	Displays the folder tool palette.

Tools, Reports and Actions Menu

The Tools menu displayed on the left represents the pull-down options available when the Navigator window is active.

Two additional menus with default labels of Reports and Actions may also be used. These menus are not displayed by default.

Each menu may contain up to fifteen product-specific entries. Examples of product-specific entries may include a list of commonly used Inquiry windows in the application, or a commonly used window that a user may want to display for a quick reference.

Window Menu

The Window menu displays the names of all open application windows and window placement options.

Cascade	Displays any open windows in a "cascaded" or stair-stepped fashion.
Tile Horizontally	Displays any open windows in a horizontally "tiled" (non-overlapping) fashion.
Tile Vertically	Displays any open windows in a vertically "tiled" (non-overlapping) fashion.
1 (Open Window)	Displays a list of open windows.

Help Menu

Use the Help menu to get additional information about Oracle Applications:

Window Help	Displays help for the current window.
Oracle Applications Library	Displays a window that lists all available Oracle Applications help text.
Keyboard Help...	Displays the current key mappings of specific functions and menu options.
Diagnostics	<p>All of the entries in these menus are used for debugging. You may be asked to use some of these functions if you need to contact Oracle Support. Before you can use these functions, your system administrator must give you access.</p> <ul style="list-style-type: none"> • Display Database Error - Displays the last database error. • Examine - Displays the underlying Oracle Forms block and item corresponding to a field on the form. • Test Web Agent - Performs a check of the Web Agent to determine if it is configured properly. • Trace - Toggles on or off the SQL Trace facility for the current session. SQL Trace provides performance information on individual SQL statements. • Debug - Turns on the Oracle Forms Debugger, provided the form you are currently in was started in debug mode. • Properties - Provides diagnostic information for Items and Folders.

- Item - Displays the values for properties of the current item.
- Folder - Writes diagnostic information about the current folder to a file in the temporary directory.
- Custom Code - The Custom Code submenu is an option group, allowing you to change the mode of the custom library.

Normal indicates that the custom library is used. This is the default.

The Off setting turns the custom library off, disabling Zooms and any other custom code.

Record History

Displays information about the current record.

About Oracle Applications...

Displays information about the current application, including version and environment information.

Pop-up Menus

A right-mouse pop-up menu is available on all text items. It replicates a subset of functionality contained on the main pull-down menu, including up to 10 product-specific entries which would also be on the Tools, Actions or Reports menus.

The pop-up menu should only appear if the user does a right-mouse click in the field that already has focus; right-mouse clicking on the noncurrent field should do nothing.

Related Topics

Searching for Information, page 2-16

Customizing the Presentation of Data in a Folder, page 4-29

Toolbar

The toolbar replicates the most commonly used actions on the menu with iconic buttons. Entries on the toolbar are disabled and enabled automatically based on the current context (OMS-71021). Hints for the toolbar buttons are displayed when the mouse moves over each button.

The toolbar is attached to the MDI window.

(OMS-72005) The tool bar consists of the following actions:

- New – Opens a new record in the active form.

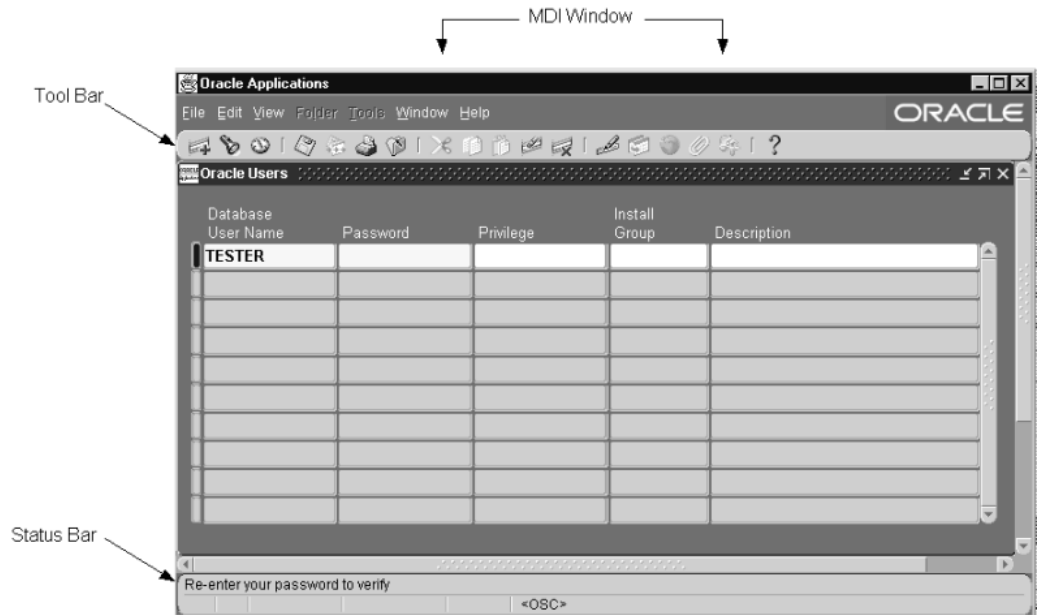
- Find... – Displays the Find window to retrieve records.
- Show Navigator – Displays the Navigator window.
- Save – Saves any pending changes in the active form.
- Next Step – Updates the Process workflow in the Navigator by advancing to the next step in the process.
- Print... – Prints your current window. An application may override this action to allow printing of one or more specific reports instead.
- Close Form – Closes all windows of the current form.
- Cut – Cuts the current selection to the clipboard.
- Copy – Copies the current selection to the clipboard.
- Paste – Pastes the contents of the clipboard into the current field.
- Clear Record – Erases the current record from the window.
- Delete – Deletes the current record from the database.
- Edit Field... – Displays the Editor window for the current field.
- Zoom – Invokes custom-defined zooms.
- Translations... – Displays the Translations window.
- Attachments... – Displays the Attachments window.
- Folder Tools – Displays the folder tool palette.
- Window Help – Displays help for the current window.

(OMS-72006) Additional buttons are available for application-specific functions. These functions would also be replicated on the Tools, Actions, and Reports menus.

For more information on the toolbar, see the *Oracle Applications Developer's Guide* and the *Oracle Applications User's Guide*.

Status Bar

The status bar contains the message and status line. The status bar is attached to the bottom of the MDI window.



Calendar

The Calendar is a standard object that allows selection of date and time values from a calendar. It also allows the developer to specify validation rules ensuring that only valid dates can be selected.

The List or Edit function invokes the Calendar on any date field (OMS-74012, OMS-74350).

The Calendar may also be invoked in a display-only mode, where it shows specific date values as highlighted. For instance, it can be used to show the vacation days taken by an employee.



For more information on the Calendar, see the *Oracle Applications Developer's Guide*.

Folder Objects

For each form that contains a folder block, there are several required objects to allow folders to be saved, prompts to be changed, the query to be viewed, and the Folder tools palette to be displayed. These objects appear to the user automatically during normal operation of a folder block.

Related Topics

Folder Blocks, page 4-29

Container Objects

Overview

This section describes the standard properties and behaviors for Modules, Windows, Canvases, Blocks, and Regions.

These characteristics may be set in the following ways:

- Inherited through property classes, which cause certain properties to be identical in all forms (such as widget heights)
- At the discretion of the developer (such as widget widths)
- At runtime, by calling standard library routines (such as window positions)

For details of the implementation of these properties, see the *Oracle Applications Developer's Guide*.

The following objects are discussed in this chapter:

- Modules, page 4-1
- Windows, page 4-4, including modal and non-modal windows
- Canvases, page 4-17, including content and stacked canvases
- Blocks, page 4-18
- Regions, page 4-37, including Tab canvases

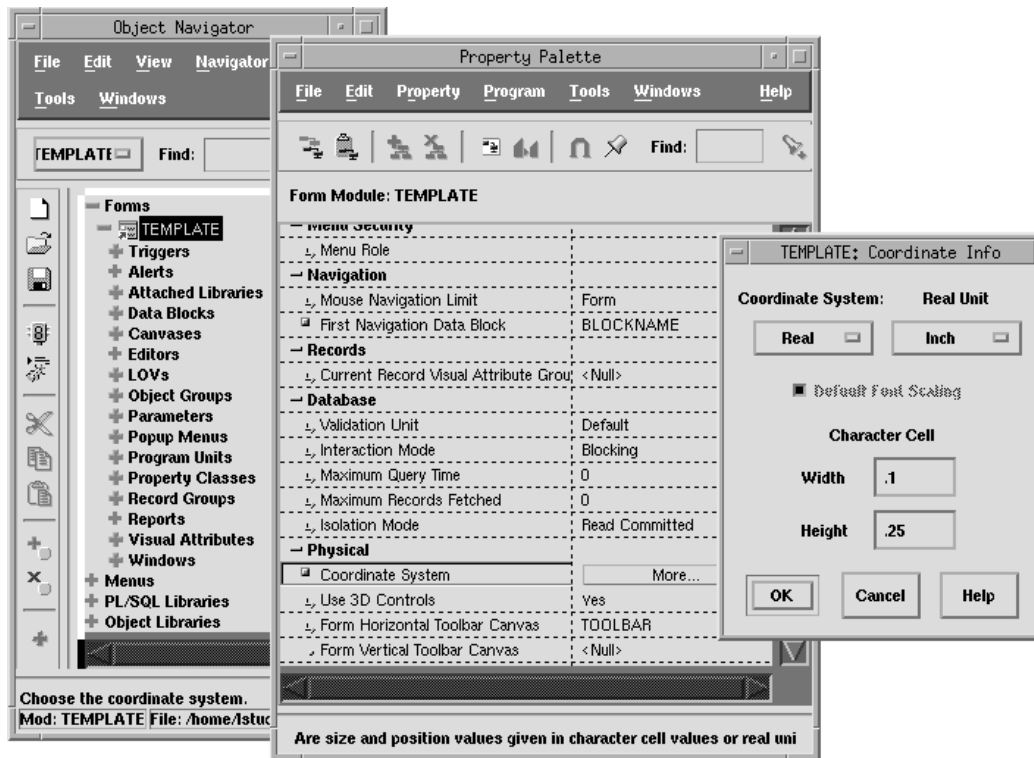
Modules

Module properties establish an overall framework for the look and feel of each form.

For information on implementing modules, see the *Oracle Applications Developer's Guide*.

Display Attributes

The coordinate system is established to allow any form to render the same size regardless of screen resolution. "Real" coordinates are used, because they are based on a logical, not physical, measure (OMS-71002). The following picture shows the relevant Oracle Forms settings: Coordinate System is set to Real, Real Unit is set to Inch, Character Cell Width is set to .1, and Height is set to .25.



(OMS-73001) The character cell height and width values (0.1" and 0.25," respectively), which govern the grid to which all items are snapped, are derived as follows:

- The width is based on the average width of the fonts used (Helvetica 10 point).
- The height is derived from the number of pixels necessary to fully display international characters (14 pixels excluding the text item bevel).

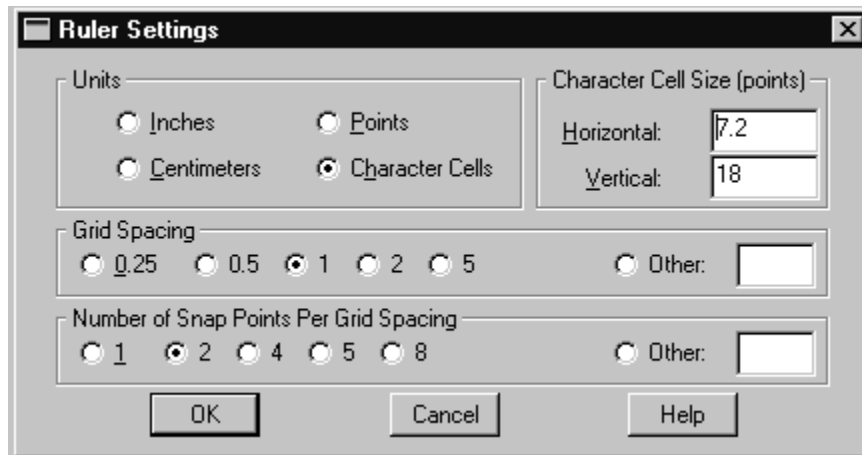
(OMS-73900) Ruler settings determine the spacing and alignment of all elements in the Oracle Forms. It is crucial that these settings are established correctly before any layout is performed and that the "Grid Snap" setting is enabled.

All references relative to cell positioning in this manual are based upon the following ruler settings:

- Units should be in character cells.

- The character cell size should be specified as 18 points vertically and 7.2 points horizontally.
- Grid spacing, which determines how the grid is drawn relative to the cell size, should be set to 1 and the number of snap points per grid should be 2.

The following picture shows these correct Oracle Forms ruler settings, which must be set for every canvas being developed.



These settings allow objects to be drawn snapped to a character cell grid, with well defined "rows" and "columns." Use of this character cell grid facilitates rapid building of, and often more aesthetically pleasing, screens than could otherwise be achieved if no grid were imposed.

Functional Attributes

The following settings determine some basic interaction rules for all forms:

- (OMS-73002) The Mouse Navigation Limit is set to "Form" so that a user can point-and-click to any field at any time.
- (OMS-73003) The Validation Level is set to "Default," allowing fields to be validated at an item level. This ensures immediate feedback to the user if errors are made.
- (OMS-73004) Database rows are immediately locked upon the user making a change to a record, thereby preventing two users from attempting to update the same row.

Related Topics

Validation Models, page 7-2

Windows

The MDI window inherits the look and feel of the GUI in which it is running. Within the MDI, all windows inherit Oracle's look and feel, which has specific characteristics for fonts, window bevel, and window manager buttons. This section describes features common to all Oracle Applications windows, as well as behaviors for modal and non-modal windows.

For information on implementing windows, see the *Oracle Applications Developer's Guide*.

General Look and Feel

Visual Attributes

All applications windows share the following visual attributes:

- Inherit their color settings from the Oracle look and feel. These attributes are not visible in the Form Builder environment as they are only rendered by the Forms Runtime engine.
- Do not use bevels around the edge of the window.
- Leave "white space" along the perimeter of the window and its contents.

Icons

Icons that are shown in the window title bar are determined in the following manner:

- The Navigator, Folder Tools, and Calendar windows have unique icons.
- All other windows use the Oracle Applications logo icon.
- The icon title is never specified, so that it will be inherited automatically from the window title, which may change at runtime.

Titles

Each window in a form must be titled uniquely. (OMS-73012).

In addition, detail windows should display context information in the window title. For example, "Benefits - John Doe" (OMS-58505).

Scroll Bars

No scroll bars are attached to individual windows within the MDI window, although the MDI window itself may have them (scroll bars may appear inside the window, attached to canvases, blocks, or items). (OMS-73013).

Toolbars

The toolbar is attached to the MDI window, not to any individual product window.

Window Style

Most application windows appear inside the MDI window ("Document" style) except dialogs and the Folder tool palette ("Dialog" style).

Size

Guidelines and limits for window size are as follows:

- The maximum window size is 10.3" (width) x 6.25" (height), to support an 1024x768 screen resolution.
- The minimum size of a window is 2" x 2." (OMS-73016)
- A window may be drawn any size between the maximum and minimum, with the following recommendations (OMS-73017):
 - Make windows only as large as is necessary.
 - Make non-modal windows within a form approximately the same size.
 - Make modal windows smaller than the non-modal windows that invoke them.

Position

The following points pertain to window position:

- A window must be fully visible when it is first opened. (OMS-73018).
- All windows are moveable within the MDI window. (OMS-73019)
- Window positions are not retained when they are closed or when a form is exited. (OMS-73020)

Button Placement

Sequence buttons for non-modal windows as follows:



- Position the default button to the left of all other buttons.

- Position the Cancel button immediately to the right of the default button to form a group.
- Position all other buttons in a group to the right.
- Exceptions:
 - If the Cancel button is the default, do not move it to the left in its group.
 - In cases where there are other actions which are logically related to the default button, place those buttons immediately to the right of the default button. Place the Cancel button in the rightmost position of the group.

Buttons for modal windows follow the same placement rules for non-modal windows with the following exception: The Help button displayed in complex modal windows is always placed in the leftmost position.



Non-Modal Windows

Non-modal windows are used for the display of most application entities. A non-modal window allows the user to interact with any other window, as well as the toolbar and the menu.

For information on implementing non-modal windows, see the *Oracle Applications Developer's Guide*.

Position

The position of a non-modal window is determined as follows:

- The first window of each form is positioned immediately below the toolbar. (OMS-73021).
- A child or detail window opens in relation to the parent window that invoked it, in one of the following styles (OMS-73022):

CASCADE: Child window overlaps the parent window, offset to the right and down by 0.3."

RIGHT or BELOW: Child window opens to the right of, or below, the parent window without obscuring it.

OVERLAP: Detail window overlaps the parent window, aligned with the left edge of the parent window, but offset down by 0.3."

CENTER: Window opens centered relative to another window (typically the MDI

window).

If these rules would cause a window to be positioned partially off the screen, then it should be shifted toward the middle of the screen until it is fully visible (OMS-73018).

- A non-modal window is programmatically positioned each time it is opened. Thereafter, it retains that position (or a position determined by the user if they move it) for the remainder of the session, but if it is closed, the altered position will not be remembered (OMS-73020).
- Sibling windows, which are windows showing entities at the same level of a hierarchy, initially open in the exact same position. That is, each sibling window may be set to CASCADE relative to the same parent window. (OMS-73024)

Title

You should adhere to the following standards and suggestions when choosing titles for non-modal windows:

- Most windows are titled with the name of the object shown in the window. (OMS-58501)

Examples:

Sales Orders

Journal Entries

- If the form performs one specific task, the name of the First Window is of the format *<Verb> <Noun>*. The verb may be a real word such as "Transfer" or a contrived but descriptive word such as "AutoCreate." (OMS-58500)

If the window is display only, use the format *View <Noun>*.

Avoid the terms "Maintain," "Update," or "Define" unless the terms substantially clarify the use of the form, and are required to differentiate them from other forms that act on the same objects.

Examples:

View Requests

View Receipts

- Window names are always plural, except when there is only one instance of data such as "Site Configuration". Whether the display is limited to one record at a time is irrelevant. (OMS-58503)
- Titles may also include the Organization Code (not the full Organization Name), Set of Books Code, or Business Group, in the format *<window title> (<Org>)*. (OMS-73027)

Examples:

Transfer Items (WR1)

Post Journals (AR1)

- If the product uses Datetrack, and the Datetrack date is different from the current date, then the title is formatted *<window title> (<Org> <Date>)*. (OMS-73028)

Example:

Hire Person (SF1 12-31-2008)

- Titles of all but the First Window of a form should also display context of the "topmost" master record. (OMS-58505)

Context is shown in the format *<window title> - <context>*. Up to three context values may be displayed in one window title.

A new record, whose primary key is not yet available, is shown with the context "[New]."

Examples:

Journal Lines - NOV08, 123

Assignments (OR1 12/30/2008) - Jane Doe

Benefits - John Doe

Purchase Order Lines - [New]

- Do not add labels to a title's context. For example, do not add "Order" to "123" to display "Order 123" in the title of the detail screen. If a label is necessary, the information should be moved to a context block. (OMS-73030)
- Generally display only one key value on the title bar. If there are two important values to be displayed and at least one of them is non-numeric, displaying both is allowed. Never show two numbers together or show three values of any sort since the combination becomes too cryptic to decipher. For example, showing an order number and customer name as window context works well, but showing the order number and customer number would be confusing. (OMS-73930)
- Whenever a master record is changed, or primary key information on the record is changed, context shown in window titles must be immediately updated. (OMS-73031)

Important: Overuse of the window title can lead to clutter and misinterpretation. If context cannot be easily shown, a context block must be used instead. Avoid any temptation to build phrases to support context in window titles as this most likely will be

untranslatable. If it is expected that the context will be frequently or always truncated, then it is best to show the context in context fields rather than the window title.

Closing

The following standards determine how non-modal windows should behave when closing:

- A user closes any window with the window close box. Do not replicate this standard method by providing "Close" or "Dismiss" buttons in any non-modal window. (OMS-58147)
- Closing a non-modal window does not force a save of the data contained in it. It merely means "do not show this information right now." (OMS-58149)
- If the cursor is in the window that the user requests to be closed, then it must be programmatically moved to a previous block that is not in that window before the window can be closed. (OMS-58152)
- When a user closes a master window the program must ensure that all of its details are closed, as well as any associated Find windows. (OMS-58150)
- If both a Summary and a Detail window exist (as with Combination Blocks), once they are both closed the program should close all details as well. Often this coincides with exiting the form. (OMS-73036)
- A Detail window must be closed if a context change is made in a master window such that the objects shown in the detail window are no longer applicable. (OMS-73037)
- Closing the first window of a form is the method to exit the form, in addition to the menu option, File -->Close Form. (OMS-73038)
- Closing a parent window displays a message asking whether to save if changes are pending. Child windows do not require the message because the data is just hidden and can be saved later. The user will be prompted to save when the parent window is closed if the data in the child window were not previously saved. (OMS-73039)
- No windows can be closed while in Query By Example (QBE) mode. (OMS-73040)

Resizing

Guidelines for resizing windows are as follows:

- All non-modal windows should allow resize, so that a user can make a window

smaller if they only want to see a portion of it. (OMS-73041)

- Only in folders and exceptional cases do windows respond to resize events and reformat the items in the window. (OMS-73042)
- All non-modal windows should allow minimization (iconification). (OMS-73043)
- Only windows that respond to resizing (such as folders) allow maximization; all others do not. (OMS-73044)
- Windows that do support resizing, either by explicitly resizing or by maximization, may "bounce back" to specific widths or heights as appropriate. For example, a window might allow any width, but may place a constraint on the maximum height.

Menus

All windows inherit the menu. (OMS-73045)

Find Windows

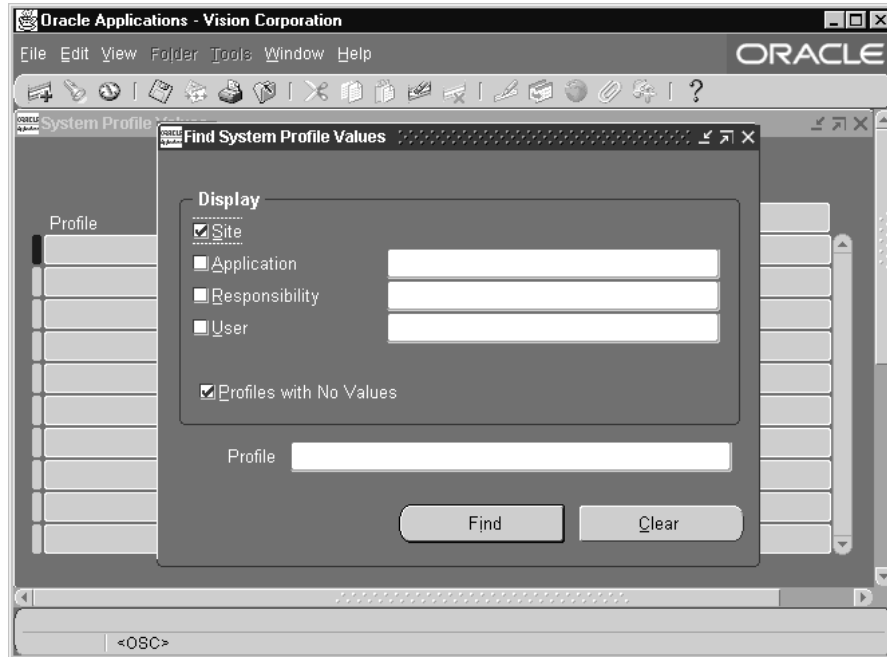
A Find window is a type of window that enables the user to locate one or more records without having to invoke Query By Example. For example, a Receiving form may allow a user to enter various criteria to locate one or more receiving headers that match the criteria, then subsequently allow the user to operate on each retrieved header and its associated lines.

A Find window should be used in the following situations (OMS-73645):

- For multi-row blocks and Combination blocks that do not autoquery.
- When the user may need to find a subset of records in a detail block.
- For any single-record or multi-record block where the user needs to search by criteria other than the primary key (for example, if the primary key is relatively obscure). Find operations for single record blocks must be more specific and result in selection of one single record rather than returning a set. For this reason, use of the Find window is discouraged when qualifying single records. A Row LOV should be used instead. See: Row LOV Windows, page 4-13.

Other blocks may use an LOV (called a "Row LOV") that shows all possible records a user can query.

The following picture shows a Find window opened centered on its Results window.



For information on implementing Find windows, see the *Oracle Applications Developer's Guide*.

Characteristics and layout of the Find Window

Find windows should have the following characteristics:

- A Find window is always drawn in single-record format. (OMS-73046)
- All fields for which a user may want to enter search criteria should be placed in the Find window so that the user does not need to use Query by Example. Tab regions should be used when the number of criteria is very large. (OMS-73546).
- Find windows are non-modal. (OMS-73047).
- The window title should be of the format "Find <objects>." (OMS-73048).
- A Find window initially opens centered on its Results window, but retains its position if moved by the user until closed. It may open to the right of, or below the Results window if it will be used frequently (for example, in inquiry-only forms). If the Find window is as large or larger than the Results window, the Results window should open cascaded in relation to the Find window. (OMS-73548).

Note: In some rare instances, the Find window includes information that determines the layout of the Results window or is otherwise required before proceeding. In those cases, only the Find

window appears when you first enter the form. If the user closes the initial Find window rather than doing a Find, the Results window does not appear (the form closes). It is also acceptable in those cases to allow the user to return to the Find window upon closing the Results window by displaying a message "Close *formname* or return to the Find window?", with the buttons "Cancel," "Find," and "Close." This message should not appear if the user explicitly chooses Close Form from the menu (only if the user chooses to close the window using the operations menu).

- A Find window automatically closes when its corresponding Results window is closed (OMS-73049, OMS-58150).
- If a Find window opens automatically when the user first enters the form, then the Find window should also open automatically when the user performs a Clear Form. (OMS-73149).

Buttons

Find windows have the following buttons (OMS-73050):

Clear	Clears the Find window for entering new search criteria. Defaults should be reapplied.
New	<p>Places the cursor on a new record in the transaction block. If the form allows entry of more than one entity or if the action can be made clearer, this button can be more fully qualified, such as "New RFQ" or "New Quotation" (there can be more than one button beginning with "New" in the window).</p> <p>If the form does not allow entry of new data, do not include this button. However, a gap should be left between the Clear and Find buttons.</p>
Find	Performs the query with the current criteria. This is the default button for the window.

Search Behaviors

- Next Block performs the same function as the Find button. (OMS-73052)
- [Tab] navigation is only within the block, and always cycles among the fields of the Find window, tabbing through all buttons except Clear and then returns to the top.
- Previous Block moves to the Results block without executing the query.

- The criteria in the Find window may be out of synchronization with the information displayed in the Results block. Only when the user presses Find or Next Block is the query performed using the current criteria. (OMS-73053)
- Multiple sets of search criteria (records) are allowed in Find windows. (OMS-73054)
Use Next Record or Previous Record to display a previously entered search or enter a new search. This allows users to make slight modifications to a prior search and run it again. (OMS-73554)
- Despite the existence of a Find window, the block being queried should also allow Query By Example. (OMS-73055)

Important: The records retrieved from a Query By Example search are not restricted by any criteria that might be in the Find window. In other words, Find window search criteria is independent of Query By Example search criteria and vice versa.

- Most fields within a Find window should validate, but with minimal cross-field validation.
- In the case where there is a free-form text field (for example, a field that holds a description) in a Find window, that field should allow wildcard entry ("% and "_") rather than using an LOV to validate. (OMS-73555).
- If no records are located after pressing "Find," the cursor focus remains in the Find window. (OMS-73556).

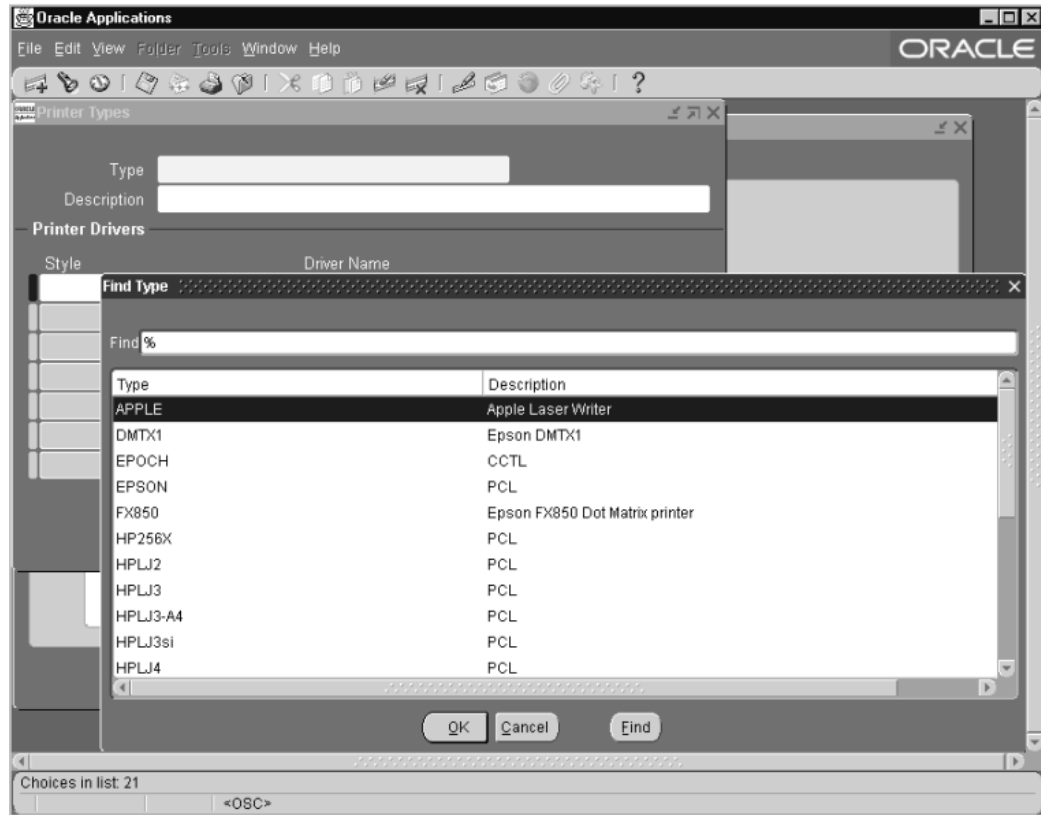
Row LOV Windows

When the user invokes View Find in certain cases a Row LOV window appears.

A Row LOV should be used in the following situations (OMS-73557):

- For a single-row block display
- In order to allow the user to select a single record in a multi-record block or combination block that autoqueries. In general, the types of multi-record or combination blocks that autoquery include detail blocks, and master blocks where the number of records is small.

The following figure shows a Row LOV Window opened for its Results window.



A "Row LOV" is a specific type of LOV that shows all possible records the user can query. In general, Row LOVs should follow the standards for LOVs as well as specific title standards for Row LOVs given in Chapter 4.

Modal Windows

Modal windows force the user to work within a single window, then either accept or cancel the changes they have made. Modal windows should only be used when a user is required to enter specific information to complete an action. Modal windows use the Dialog Block mechanism to control access and navigation.

For information on implementing modal windows, see the *Oracle Applications Developer's Guide*.

Position

Modal windows are always opened centered on the MDI window or centered relative to other windows. (OMS-60053).

Title

Modal window titles should be closely related to the labels of the widgets that open them. (OMS-73558)

Menus and Toolbar

Modal windows have the menu associated with them, but the user cannot have access to it. There are a few legacy screens that allow limited access to the toolbar and menu, but no new instances should be designed or coded. (OMS-73057)

Because the menu is not accessible from a modal window, function-key access to the following actions must be supported (OMS-73058):

- Next Item
- Previous Item
- Clear Record
- Edit
- List
- Enter
- Help
- Print

If the modal window allows multiple records, then the following should also be enabled via function keys:

- New Record
- Next Record
- Previous Record
- Up
- Down

All other function keys should issue a "beep" when the user attempts to invoke them.

Closing

Modal windows are generally closed in response to the user pressing a button that concludes work in that window, although the window close box may still be used. Typical close actions and how they are used are as follows (OMS-73060):

OK or Done	Closes the window. In some cases, it may perform a save as well.
-------------------	--

Note: A specific verb can be substituted in place of "OK." For instance, in a window designed to record additional information before posting, buttons of "Post" and "Cancel" are clearer to the user than just "OK" and "Cancel."

Cancel	Clears the data without asking for confirmation, and closes the window.
Apply	Processes the changes made in the window, but does not close it.
Window Close Box	Performs the same action as "Cancel."

Size

Modal windows do not allow resize, maximization, or minimization. (OMS-73061)

Semi-Modal Windows

This term refers to one window in a set of two or more windows opened among which the user can navigate as a modal group. In this case, a user may not return to a calling window until an action is completed or canceled, but needs to be able to move among multiple windows for doing that action. In those cases, "Semi-Modal" windows are used.

After invoking a "Semi-Modal" window, the user may move among the subset of windows freely, but cannot move back to the calling window without completing or canceling the action (typically with Done or Cancel buttons). Clicking on the calling window results in a "beep," and the user's previous position is restored. (OMS-73561)

Semi-modal windows are not a native Oracle Forms feature. See the *Oracle Applications Developer's Guide* for further information.

Related Topics

Context Blocks, page 4-19

View Find, page 2-15

Lists of Values (LOV), page 5-17

Dialog Blocks, page 4-21

Implicit Saves, page 7-3

Canvases

Canvases are the surfaces on which all interface items are placed. In general, all canvases share the following property:

- No bevels are used around the edge of the canvas. (OMS-73062)

This section describes aspects of canvases that depend on whether the canvas is a content canvas or a stacked canvas. For information on Tab canvases, see *Regions*, page 4-37.

For information on implementing canvases, see the *Oracle Applications Developer's Guide*.

Content Canvases

Each window contains one content canvas, which fully occupies the window. Additional stacked and Tab canvases may be displayed in front of the content canvas.

Display Characteristics

Content canvases have the following display characteristics:

- All content canvases are set to display immediately. (OMS-73064)
- Content canvases do not raise on entry. (OMS-73065)

Size

The size should be the same as the window it will be shown in. (OMS-73565)

Stacked Canvases

Stacked canvases are the mechanism for placing content in a portion of the window which may be alternated to show another set of content. One or more stacked canvases may be displayed in front of the content canvas of a particular window. If needed, a stacked canvas may fully occupy a window.

For information on implementing stacked canvases, see the *Oracle Applications Developer's Guide*.

Display Characteristics

Stacked canvases should adhere to these display characteristics:

- Only the one stacked canvas that is to be shown when its window is first opened should be set to "Visible." (OMS-73066)
- Stacked canvases always raise on entry. (OMS-73067)

Size

The Size should be set to the exact size necessary to contain all of the items on the canvas. (OMS-73567)

View

The View specifies the size of the view port in which the stacked canvas will be displayed. Ideally it is the same as the canvas size, otherwise scrolling is implied.

Scrolling should be avoided if possible, but if required, it must be limited to twice the width or height of the viewport (OMS-73122).

Sequence

When multiple stacked canvases occupy the same window, and may overlap, the sequence must be set so that the proper canvases, or portions of canvases, are displayed.

Scroll Bar

If a stacked canvas is scrollable, then a scroll bar must be enabled. Also, if any canvas requires a scroll bar in Tab regions, all of the stacked canvases corresponding to that Tab region should then have a scroll bar. Canvases in single-record formats scroll vertically; those in multi-record formats scroll horizontally. Never allow both vertical and horizontal scrolling of a stacked canvas, except when needed to display a graphic image. (OMS-73267)

Blocks

Most properties of blocks are form-specific, such as the ability to insert or update data. Only basic cosmetic properties are common to all.

For information on implementing blocks, see the *Oracle Applications Developer's Guide*.

General Block Rules

Title

Block titles should be displayed and chosen according to the following guidelines:

- The title is the name of the object displayed in that block. It is optional, unless any of the following are true (OMS-73068):
 - The object represented by the block is not obvious and is not the same as the window title.
 - The block must be distinguished from another similar block in the same

window.

- Titles are singular if the block only displays one record, and plural when more than one record is shown. A single-record block may have a plural title if the user normally accesses more than one record during a single transaction, and the block is not shown in a multi-record format elsewhere in the form. (OMS-73069)

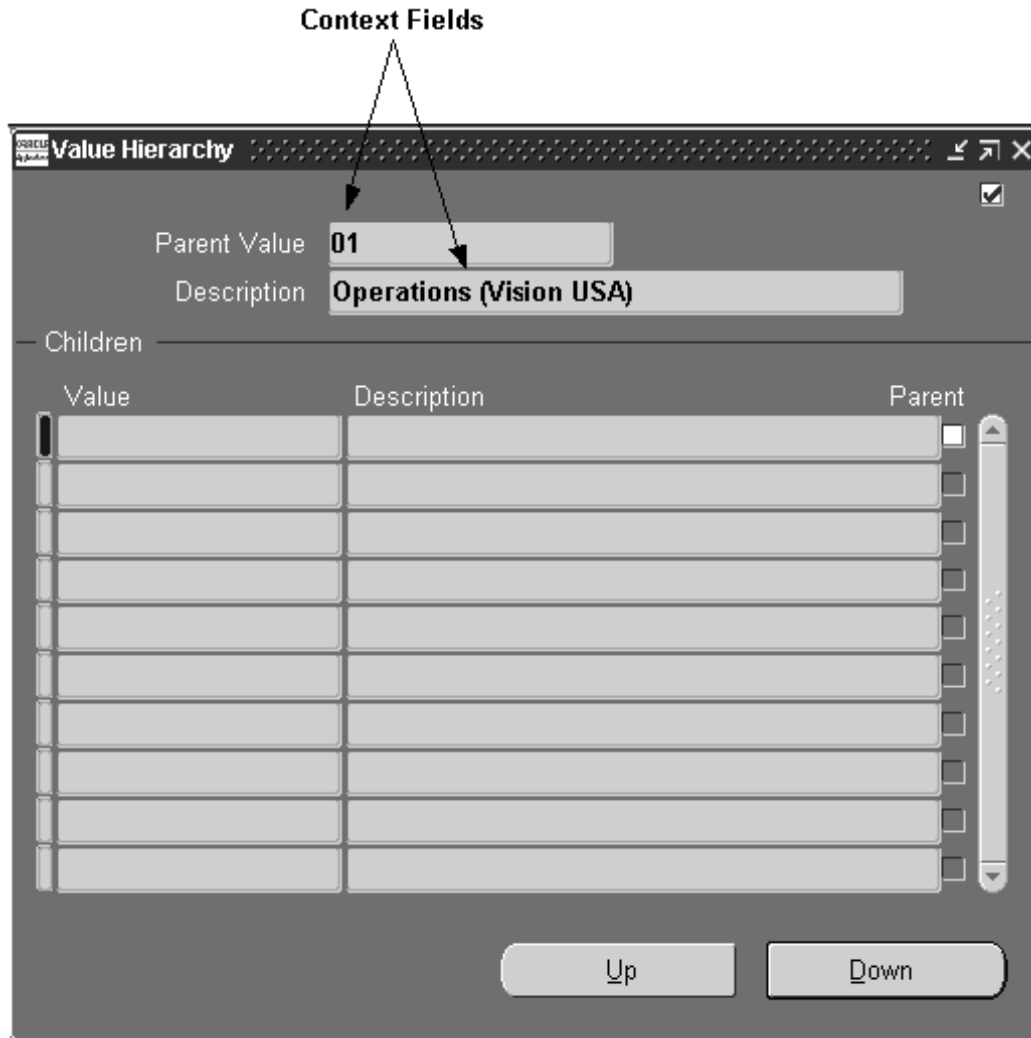
Titles for a block are rendered automatically as the title attributes of a "frame" object. This can be either a full square container, or a zero-height frame that renders as a horizontal line for delimiting blocks. The visual properties of these titles are inherited from the Application Object Library property classes.

Note: These settings apply specifically to titles that are displayed with a frame. Appropriate settings must be applied when the title is a display item, check box or radio button.

- The widget that displays the title may be any of the following (OMS-73572):
 - Zero-height frame title (for static block titles)
 - A display item overlaying a frame, designed to look like boilerplate (for dynamic block titles)
 - A check box or radio button overlaying a frame

Context Blocks

Each non-modal window must be designed so that a user can maintain context merely by viewing that single window. This is necessary because of the multitude of windows, possibly across several forms, that may be on the screen at any one time. Also, because a user may iconify the window containing the context of the data in the current window, each window must display its own scope. The window title is the preferred way to show context, but when it cannot meaningfully or fully display the context for a window, additional context is displayed in a context block.



Context Block Characteristics

Context blocks should have the following characteristics:

- The context fields are placed at the top of the window to indicate primary key information of the master record(s). (OMS-73672)
- Context fields are placed in a single-record format, with the highest master context located first. (OMS-73673)
- Only the information that uniquely identifies the master record(s) is displayed. A user can access more information about the master by viewing the window that contains it. (OMS-73674)
- Additional descriptor fields may be drawn in the context block only if space permits, without compromising the space made available to the primary contents of

the window. (OMS-73675)

- Whenever a master record is changed, or primary key information on a master record is changed, context shown in other windows should be immediately updated. (OMS-73676)

Dialog Blocks

Dialog blocks are the blocks presented in modal windows. They require the user to interact with them before proceeding with other windows of the application.

Dialog Block Characteristics

Dialog blocks should have the following characteristics:

- Dialog blocks always contain buttons to dismiss them, such as "OK" and "Cancel." (OMS-73773)
- Only in exceptional cases should a Dialog block allow multiple records. (OMS-73774)

Enabled Functions

Most standard Oracle Forms functions do not apply in a Dialog block. However, some functions may be enabled under certain conditions.

Navigation

Navigation to items outside a dialog block must be prevented while the modal window is open. The following guidelines prevent the user from navigating out of a Dialog block (OMS-73074):

- The Next and Previous Navigation Data Block must be the same as the Dialog block within the window, and no fields of that block should be shown in other windows. (OMS-73874)
- Navigation by [Tab] key must be restricted to fields within that window. (OMS-73875)
- Navigation style of the block must remain on the current record. (OMS-73876)

Single-Record Blocks

Single-record blocks allow the user to see many attributes of a single record. The single-record format should be used when only one record is possible or the user only works with one record, or when it is necessary that the user see many attributes of one record at the same time.

Single-Record Block Layout

- Items on adjacent rows should be left-aligned where possible. (OMS-73174)
- Items should be sequenced by order of precedence from left-to-right, then top-to-bottom (OMS-71005).
- If the block contains multiple regions, the tabbing sequence should move between the items of a region before moving to another region within the block. Tab regions should be arranged to match the tabbing sequence. (OMS-73175)
- Prompts should be placed to the left of the fields (OMS-75009).
- Fields should be vertically stacked unless space permits gaps between rows. For more information on item spacing, see the chapter called Widgets., page 5-1 (OMS-74009, OMS-74020)
- Gaps can be used to group logically related fields. Visually grouping fields by using gaps can be used as an alternative to a frame if the relationship between the fields in a group is clear.

Note: When information is shown in a single-record format, it is preferable to arrange the fields in as few columns as possible. In other words, if all the fields can be left-aligned in a single column, do so unless such an arrangement would break a strong field-placement consistency you have achieved among different forms. Left-aligning fields in as few columns as possible makes it easier for users to scan the information quickly. (OMS-73176)

Querying

- In cases where there is only one record possible, QBE, View Find, and View Find All should all automatically bring up that one record. (OMS-73075)
- In cases where the user must use the Find window to query records, QBE, View Find, and View Find All should all automatically bring up the Find window. (OMS-73076)

Clearing

In cases where there is only one record possible, Edit Clear Record and Edit Clear Block should automatically requery that one record. These functions should issue a "beep" in this situation to indicate that their behavior is different from normal. (OMS-73077)

Multi-Record Blocks

Multi-record blocks allow the user to see as many records of an entity as possible, usually at the tradeoff of seeing fewer attributes of each record simultaneously. For general information on when multi-record formats should be used, refer to the section General Design and Layout, page 2-1.

For information on implementing Multi-Record Blocks, see the *Oracle Applications Developer's Guide*.

Scroll Bar

Multi-record blocks display a record scroll bar, as follows (OMS-73078):

- The left edge of the scrollbar should be 0.3" inward from the right edge of the window or Tab canvas.
 - Scroll Bar Orientation: Vertical
 - Scroll Bar Width: 0.2"
 - Height: Same as all the records in the block
 - Reverse Direction: False
 - Vertical position: Aligned with the top of the first row of items

Note: The only multi-record blocks that do not have a scroll bar are those that show enough rows on the screen to accommodate the maximum number of records.

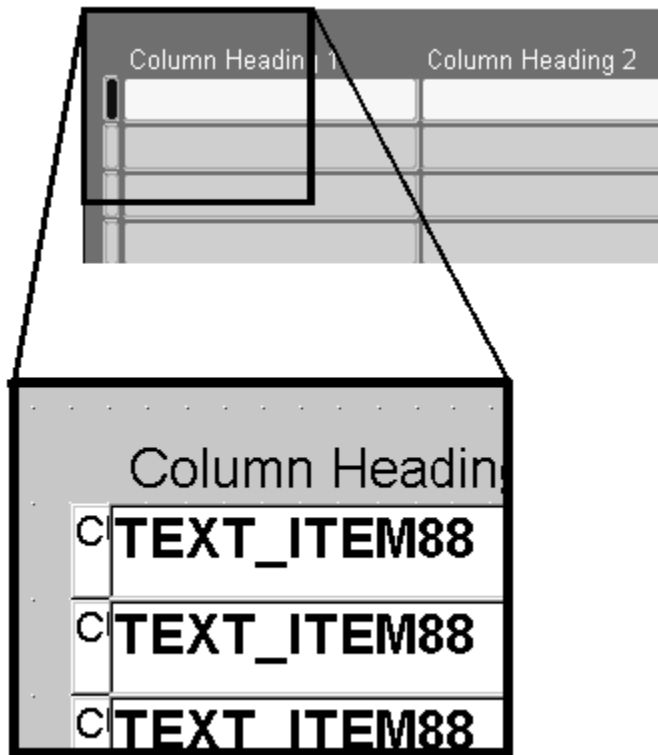
The pictures in the next section, in addition to showing the current record indicator, show the position and width of the scroll bar.

Current Record Indicator

(OMS-73178) All multi-record blocks have an indicator to point out the current record. The indicator looks and behaves in the following ways (OMS-73079):

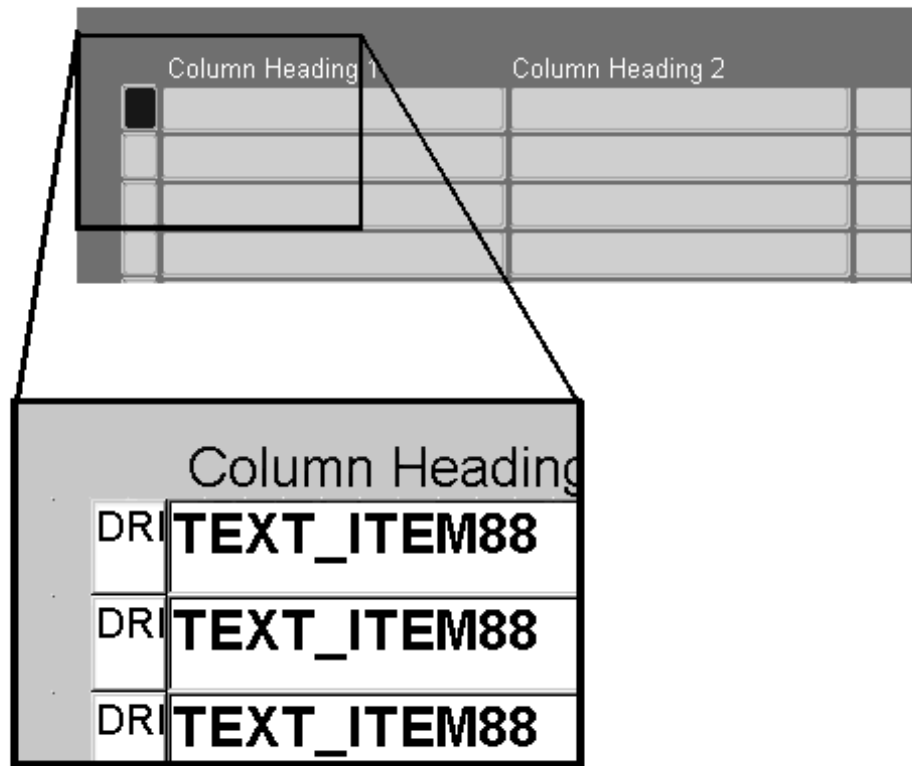
- Width: 0.1" wide.
- Position: Immediately to the left of the first column of the block with no blank space on either side of the indicator.
- Color: Blue on the current row, and gray on all others.
- Bevel: Raised

- Clicking on it moves to the first item in the record.



(OMS-73080) If the block supports a "drill-down" capability, then the indicator has the same characteristics as above, except for the following (OMS-73580):

- Width: 0.2"
- Clicking on the indicator should perform the appropriate "drill-down" function for that block.
- If you are on a Combination block, drill-down takes you to the detail (single row) view of the record.
- If it is not a Combination block, drill-down can go to any further details (the original transaction, for example), where it should autoquery those details.
- In most cases, drill-down in a non-Combination block should take you to the same Detail block that you reach via the default button (if there is one).
- If the drill-down cannot currently be performed, either a message must be displayed (the same as if the corresponding button were pressed), or it must issue a "beep" (if the corresponding button is disabled).



In both cases, single-clicking or double-clicking on the indicator should move the cursor to the first navigable field of the appropriate record.

For information on implementing the Current Record Indicator, see the *Oracle Applications Developer's Guide*.

Layout

- Columns are generally stacked horizontally and aligned along their tops. (OMS-73180)
- Multi-row blocks should display a minimum of three records, unless only a maximum of two records can ever exist.
- Space may be left between columns where prompt sizes or region cosmetics require it. See the section Text, page 6-1 for more information on cases where a field prompt is wider than the field.
- If the window is wider than the multi-record block, then the multi-record block should be positioned on the left side of the window. (OMS-73280)

Querying

- In cases where the maximum number of possible records is less than or equal to the number of rows shown in the window, QBE, View Find, and View Find All should all automatically bring up all possible rows. (OMS-73081)
- In cases where the user must use the Find window to query records, QBE, View Find, and View Find All should all automatically bring up the Find window. (OMS-73082)

Combination Blocks

Combination blocks are hybrid formats, where fields are presented in both multi-record ("Summary") and single-record ("Detail") formats. The Summary and Detail formats are each presented in their own non-modal window. For general information on when Combination blocks should be used, refer to General Design and Layout, page 2-1.

For information on implementing Combination blocks, see the *Oracle Applications Developer's Guide*.

Combination Block Windows

The Summary and Detail windows of a Combination block should behave as follows:

- The Summary and Detail windows should have the same width. (OMS-73084)
- The Detail window is positioned just below the title bar of the Summary window, with the left edges aligned. (OMS-73085)
- Subsequent child windows are positioned cascaded relative to either the Summary or Detail window, depending on where the cursor is when the child window is invoked. (OMS-73086)
- If the Detail window is opened directly from the menu without the Summary window, it will be positioned down slightly as if the Summary window were open, so that if the user subsequently accesses the summary window room is still available. (OMS-73087)
- The user can close either the Summary or Detail window while leaving the other one up, and any child windows will remain open. If the user closes either the Summary or Detail window when the other is already closed, all child windows of that pair will close as well. If this is the top level pair, the whole form will close. (OMS-73088)

Behavior of Combination Blocks

The Summary and Detail blocks of a Combination block should behave in the following

manner:

- The View menu contains a Summary/Detail command for the user to easily switch between the Summary and Detail windows. This command is enabled only for Combination blocks. (OMS-73089)
- The Summary block has a current record indicator that supports drill-down, which moves the cursor to the Detail block for the current record. (OMS-73090)
- Fields in both Summary and Detail format should allow updates. If that is not possible, only allow updates in the Detail format, reflecting these updates in the Summary display. (OMS-73590)
- Selecting Find will open the Find window, and then will display the Summary view once rows are retrieved. The only exception to this occurs when the cursor was in the Detail window, and the Find only retrieves one record. In that case the cursor should remain in the Detail window. Closing the Find window will return the cursor to the block it was in previously. (OMS-73690)
- Delete Record, QBE, or any other function that would normally move the cursor to the "first" item of the block leaves the cursor in the appropriate Summary or Detail window from which the function was invoked. (OMS-73091)
- Changing to a different record in either block changes to that record in the other view as well. (OMS-73092)

Buttons

Besides any product-specific buttons, the following buttons may also be coded on the Summary window (OMS-73592):

New	Creates a new record. Adding a new row by pressing a "New" button will automatically bring the user to the Detail window. The button label may be qualified, such as "New Line," if necessary to clarify its intended function.
Open	Moves the cursor to the detail window. The "Open" button should always open to the detail level of the block that contains the button. For example, if the user has navigated to the line level of a Purchase Order and chooses the "Open" button in that window, then the details for the line should be displayed.

Buttons on the Detail window may include additional actions not available from the Summary window.

Gateways

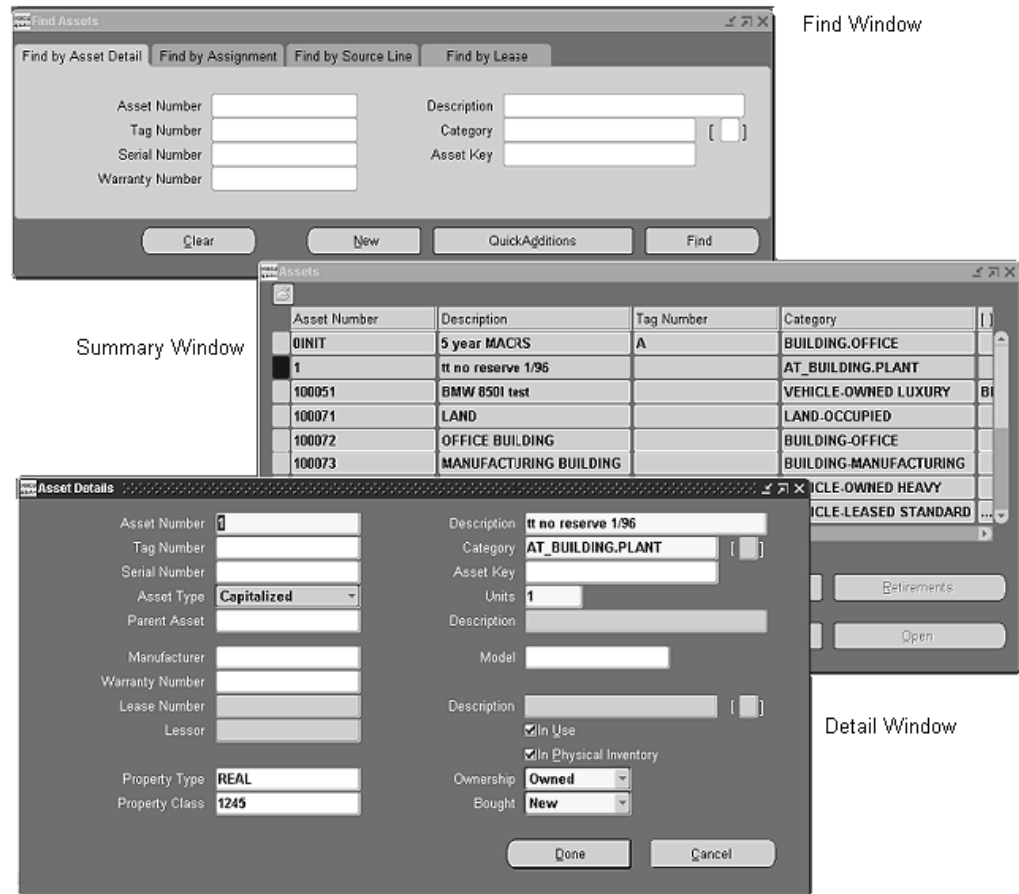
Gateways offer the user flexible methods to locate, view, and operate on records. They are employed for all the major entities of a product, such as Purchase Orders, Sales Orders, Journals, and Quality Plans. The gateway is the first set of windows that a user sees when interacting with any of these entities. It is comprised of a Combination block and a Find window, with the following unique characteristics:

- The summary block of a gateway is always a Folder block. (OMS-73093)
- The Find window automatically appears when the gateway is opened. If, however, a default folder is defined for the block, and it is set to Autoquery, then the Find window does not appear. (OMS-73094)

Workbenches

A workbench is a gateway with sufficient functionality that the user will likely be able to accomplish much of their job from this form. It would typically be left open during a user's entire work session for repeated use.

The following picture shows the set of windows (Find, Summary, and Detail windows) associated with a gateway.



Note: The above picture is not intended to illustrate the actual position in which these windows initially open.

Folder Blocks

A Folder block is a block that allows the user to customize the set of columns and records displayed for a specific entity. It can be thought of as a "file cabinet" that holds all the records of a certain object, with each individual "Folder" being a specific subset of those records shown in a specific way. For example, if a developer provides a Folder block that shows "Requisitions," then a user could create a Folder called "Unapproved Requisitions" which only shows those requisitions with status "Unapproved," and displays the "Creation Date," "Preparer," and "Next Approver" fields. One or more Folder definitions can be saved per entity, such that screens can be designed appropriately for different tasks. Each Folder is, of course, restricted to data that the user is allowed to view based on the security rules of the product.

Folder Functions

The Folder functions can be invoked from the Folder pull-down menu, the right-mouse menu, or the Folder Tool palette. A profile (Folders: Allow Customization) allows system administrators to prevent individual users from accessing all Folder tools other than Open. Following is a list of the Folder functions and their corresponding Folder Tool palette buttons (where applicable).

Note: Some Folder menu items do not function when you are in QBE mode.

New	Creates a new Folder. The user must enter a new, unique (per entity and user) Folder name.
Open	Loads a previously defined Folder. A user can select from a list of his own Folders and any public Folders for the current entity.
Save	Saves the current Folder. If it has never been named, this function reverts to the "Save As" functionality.
Save As	Allows a user to save a Folder, specifying a name, whether it should autoquery upon loading, whether it should serve as the default for the user upon entry to this form, whether other users can use the same definition, and if the current query should be retained. Folder names must be unique per user. If a user modifies someone else's public Folder in any way, saving makes it a private definition. However, opening a public Folder, and saving it as the Default, with no other changes, merely saves the reference of the Folder as the private default.
Delete	Allows a user to delete any Folder for the current entity that they created. If another user is referencing the Folder as their default, that reference is deleted as well.
Show Field	Opens an LOV displaying fields that can be shown (and are not currently shown).

Selecting a value adds the field after the current cursor position.

Hide Field

Removes the current field. The cursor moves to the field sequenced after the field that was just hidden.

Move Right

In multi-record blocks, swaps the current field with the one to its right. In single-record blocks, moves the current field one character cell to the right.

Move Left

In multi-record blocks, swaps the current field with the one to its left. In single-record blocks, moves the current field one character cell to the left.

Move Up

In single-record blocks, moves the current field one character cell up.

Move Down

In single-record blocks, moves the current field one character cell down.

Widen Field

Increases the width of the current field, up to a maximum size of 20 inches, in 0.2 inch increments.

Shrink Field

Decreases the width of the current field, to a minimum size of 0.3 inches, in 0.2 inch increments.

Change Prompt

Allows the user to alter the prompt of the current field. While in the Prompt modal window, Default allows quick recovery of the prompt initially specified by the developer. Prompts which start with "-" do not appear at runtime. This allows fields to have a prompt associated with them for selecting the field when showing a field, but the prompt is not displayed on the field itself.

Autosize All

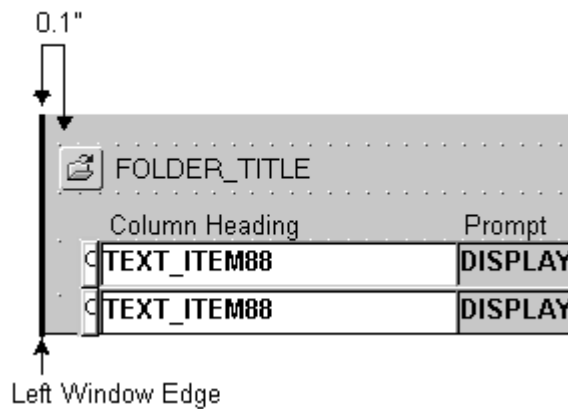
Resizes displayed fields based on a sample of values for the field in the block, ensuring that no field is smaller than the

width of its prompt (in a multi-row block). The number of records is determined by selecting one of the three options: 10, 50, or 100.

Sort Data...	Invokes a modal dialog that allows the user to specify the order and the fields by which to sort the data in the table. Sorting is limited to the first three columns only.
View Query	Allows the user to view the "where" clause of the Folder.
Reset Query	Clears the current "where" clause.
Folder Tools	Shows the Folder Tools palette.

Folder Cosmetics

- An Open Folder icon and the Folder title are displayed on their own row between the block title (if it exists) and the fields or regions of the Folder block itself. (OMS-73095)
- The Open Folder icon is positioned 0.1" from the left edge of the window or Tab canvas. (OMS-73096)
- The title is positioned immediately to the right of the Open Folder icon. (OMS-73097)
- The Folder title is blank until the user creates a new Folder or opens an existing one. (OMS-73098)
- Prompts for a multi-record folder block are drawn with a gray background and a raised bevel.
- Prompts for a single row folder block look like typical prompts.



Developer Folders

A developer may employ the Folder technology to present different layouts to the user, but not expose the Folder functions. In that case, the Folder menu remains disabled, and no Folder title or Open Folder icon is displayed.

Direct Manipulation

In multi-record folder blocks, the following actions are supported by manipulating the prompt above each column:

- A field can be resized by dragging the right-hand edge of the prompt.
- Fields can be resequenced by dragging the prompt to a new location.
- Clicking with the left mouse button on one of the first 3 columns sorts that column in Ascending then Descending order.
- Clicking with the right mouse button opens the Change Prompt dialog.

Find Blocks

A Find block is a block where users can only enter an already-existing primary key to view and maintain details (child records) of one specific master record. Find blocks are very similar to Find windows, except that the search criteria is limited and appears in the same window as the results, and the search is for details of a particular master record. For the situations that Find blocks address, a separate Find window is not appropriate.

Note: The term "Find block" in the Standards refers to the case discussed in this section and should not be confused with the block

underlying a Find window.

Appearance of Find Blocks

The following standards apply to the appearance of a Find block:

- It is displayed in the same window and above the detail records. (OMS-73099)
- It contains one or more fields to specify an already-existing primary key or multi-part primary key. (OMS-73100)
- It has a single button labeled "Find" located on the right side of the block. (OMS-73101)
- It is separated from its detail records by a zero-height frame. (OMS-73102)
- When the entity being found is not clear (for example, when one must enter a multi-part primary key in the Find block), the Find block should have a frame above it with a title like "Find Object" where the object being found is identified in singular. (OMS-73103)

Behavior of Find Blocks

- When the user chooses the Find button or navigates to the next block after specifying the primary key information in the Find block, the detail records are immediately displayed in the Detail block. (OMS-73550)
- When the user leaves the Find block and enters the Detail block without explicitly choosing Find or navigating to the next block, the information in the Find block should be checked and the details should be re-queried if the Find information has changed. This must be done because in a Find block, the Find information also serves as the only context for the details that are queried. (OMS-73551)

Note: This situation is different than the behavior of a separate find window.

- Find blocks, like Find windows, allow multiple records, so that a user can easily make slight modifications to a prior search and run it again. The user can choose from previously entered Find criteria by navigating between records in the Find block. (OMS-73552)
- Invoking the "Previous Block" command from within a Find block causes the message "At first block" to be displayed. (OMS-73553)

Alternative Blocks

Normally it is desirable to show all blocks for an entity in a single window, if possible. When all the blocks cannot be shown at once, then "alternative blocks" can be employed.

Use the Tab control to show multiple blocks within the same window when they all cannot be rendered simultaneously. (OMS-73203)

Master-Detail Characteristics

Master-Detail relations describe how detail records behave as a result of changes to Master records. For more information on Master-Detail Characteristics, see the *Oracle Applications Developer's Guide*.

Titles

To indicate master-detail relations, and for general clarity, try to repeat the master block name in the Detail block title. For instance, use "Order Lines" rather than just "Lines." (OMS-73559)

Coordination

Coordination between master and Detail blocks should follow these standards:

- When a Detail block is in a different window than its master, and each window is non-modal, then the Detail block must provide a mechanism for the user to toggle between immediate and deferred coordination. This allows a user to keep a block visible, but control the performance cost of coordinating detail records when the master record is changed. (OMS-73104)

A coordination check box is drawn with no label or prompt, and is positioned as follows (OMS-73105):

- In a tab region, the check box is drawn in the top row of the tab region and in the right corner of the region.
- If coordination applies to the whole window, the check box is placed on the top line of the window, in the upper right corner, in the rightmost three character cells.
- If coordination applies to all regions of a tab control, the check box is placed above the tabs of the tab control, aligned with the right edge of the tab.
- If there is a frame separating the Detail block, and the coordination check box applies only to that block, then the coordination check box should be placed on the frame.

- If there is a Folder title, and no frame, it is positioned on the Folder title line in the rightmost three character cells.

The coordination check box controls the coordination behavior as follows (OMS-73106):

- When the check box is checked, coordination is immediate (that is, when the master record is changed, detail records are immediately queried).
 - When it is not checked, coordination is deferred (that is detail records are only queried upon entering the Detail block).
 - When the button that leads to the Detail block is pressed, query the child records regardless of whether the coordination check box is checked or not (if they have not been queried already).
 - Whenever the window containing the Detail block is opened, the relation coordination is set to the current value of the coordination check box.
 - Whenever the window containing the Detail block is closed, the relation coordination is always set to deferred, but the coordination check box value is left unchanged.
 - Checking the check box automatically queries the detail records. Unchecking the coordination check box, however, should not automatically clear the detail records.
 - Clear Form resets all coordination back to the initial value.
 - When the coordination check box is changed by the user, a message describing the new coordination status is shown on the message line.
- When a Detail block is in a different window than its master, but the detail window is modal, the Detail block should only query upon navigation to the block. (OMS-73107)
 - When a Detail block is in the same window as its master, and both blocks are visible simultaneously, they should usually be immediately coordinated. (OMS-73108)
- In the case where such a query can be very costly, allow such a query to be deferred until the cursor enters the block, or allow the user to set the block coordination (as described above). Violating this rule should be done with care - perhaps the two blocks really do not belong in the same window, or do not need to be visible at the same time.
- When a Detail block is in the same window as its master, but both blocks are not visible simultaneously (as in alternative blocks), the Detail block should only query upon navigation to it. (OMS-73109)

Masterless Operations

A user cannot enter or query detail records until in the context of a master record.
(OMS-73110)

Other Rules and Behaviors

The following are other things to keep in mind when implementing Master and Detail blocks in your forms:

- The "topmost" master block of a form does not autoquery unless (OMS-73650):
 - only a very small number of records will be returned.
 - the query will be fast.
 - the user will most likely operate on one or more of the queried records.
 - it is a Folder block. Folder blocks may autoquery, but this is determined by each user as part of the Folder definition.
- Do not code anything specific to windows being iconified, although iconifying a window that contains a master block may make it difficult to operate with a Detail block. (OMS-73651)

Related Topics

Modal Windows, page 4-14

Single-Record Block Prompts, page 4-21

Text Items, page 5-2

Display Items, page 5-5

Multi-Record Formats, page 2-9

Multi-Record Block Prompts, page 6-6

Combination Blocks, page 2-10

Non-Modal Windows, page 4-6

Non-Modal Windows, Position, page 4-6

Tab Regions, page 4-43

Regions

Regions are logical groups of fields, represented with either the frame or tab control.

For information on implementing regions, see the *Oracle Applications Developer's Guide*.

Title

You should adhere to the following standards and suggestions when choosing titles for frames and tabs:

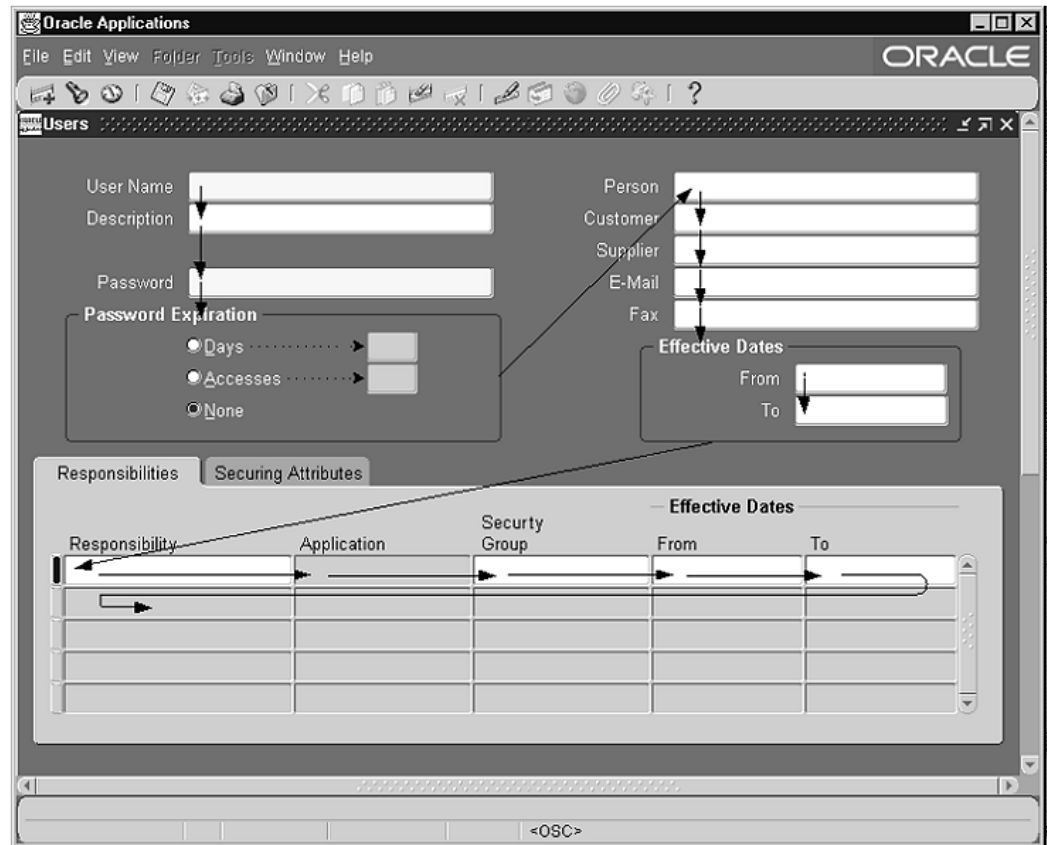
- The title is the name of the group of items it contains (usually a noun). (OMS-73111)
- The title is required for most frames, except where the contents have an obvious function. (OMS-73750)
- For a frame, the control which displays the title may be any of the following (OMS-73751):
 - Frame title (for static region titles)
 - Display item, designed to look like a frame title (for dynamic titles)
 - Check boxes or Radio groups (for when an entire region may be applicable or inapplicable)

Navigation

Navigation within and between regions should follow these rules:

- Tab order proceeds left-to-right then top-to-bottom within a region, and left-to-right then top-to-bottom between regions (OMS-71005).
- Avoid arrangements of groups or fields where the tab sequence may be unpredictable.

The following picture illustrates navigation order between regions and within a region.



Frames in Single-Record Formats

Appearance

Frames in single-record blocks should have the following look:

- A frame is drawn around the region, with its corners drawn in the center of character cells. (OMS-73117)
- The vertical sides of the frame may be displayed in the first and last column within the window.
- Multiple frames may be drawn side-by-side, or top-to-bottom, but ideally should form rectangles when viewed together. (OMS-73350)
- If two frames are drawn contiguously, adjacent edges may be drawn within the same character cell to conserve space.

Overflow Regions

Overflow regions show additional detail about the current record of a multi-record block in a single-record format below the multi-record fields.

For general information on when overflow regions should be used, refer to the section on Overflow Regions.

Position

You should use the following standards when creating overflow regions:

- Overflow regions for a block are usually drawn below the items of the multi-record block. Leave a gap of at least one half row between the multi-record block and the overflow region. (OMS-73118)
- If there are many fields to show in the overflow region, or the user may only want to see such fields on request, then the overflow region may be displayed in a separate window, such that the window title serves as the region name. (OMS-73818)

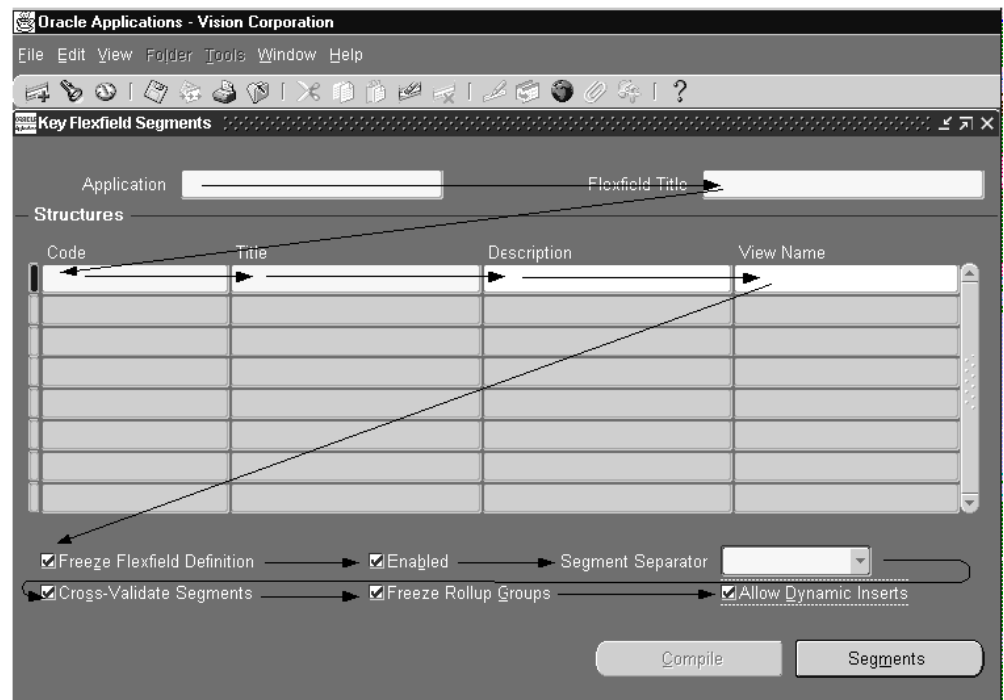
Cosmetics

The region is always drawn without a surrounding frame if it is clear that the fields pertain to the current record of the multi-record block, and no title is necessary. (OMS-73819)

Navigation

Navigation in overflow regions is determined as follows:

- Usually items in an overflow region are display only, and cannot be navigated to.
- Fields in the overflow region may be enterable if they are not applicable to all records and thus cannot be displayed in the multi-record block, or have unique sizing requirements, such as multi-line text items. (OMS-73218)
- When items in the overflow region allow users to enter or edit values, these items should be set to navigable, and the tab sequence should then go from the last field of the record in the multi-row part to the first field in the overflow region. Once in the overflow region, the navigation order is left to right, then top to bottom (as usual). From the last item of the overflow region, [Tab] moves to the first item of the next record in the multi-row part. (OMS-73318)



Other Behaviors

- A different region may be shown with each record of a block if necessary. However, if there are elements that are common to all regions, they should be displayed so as to remain in identical positions for all records. The region title should remain consistent if possible.
- Fields in overflow regions may be queryable, even when they are not navigable in

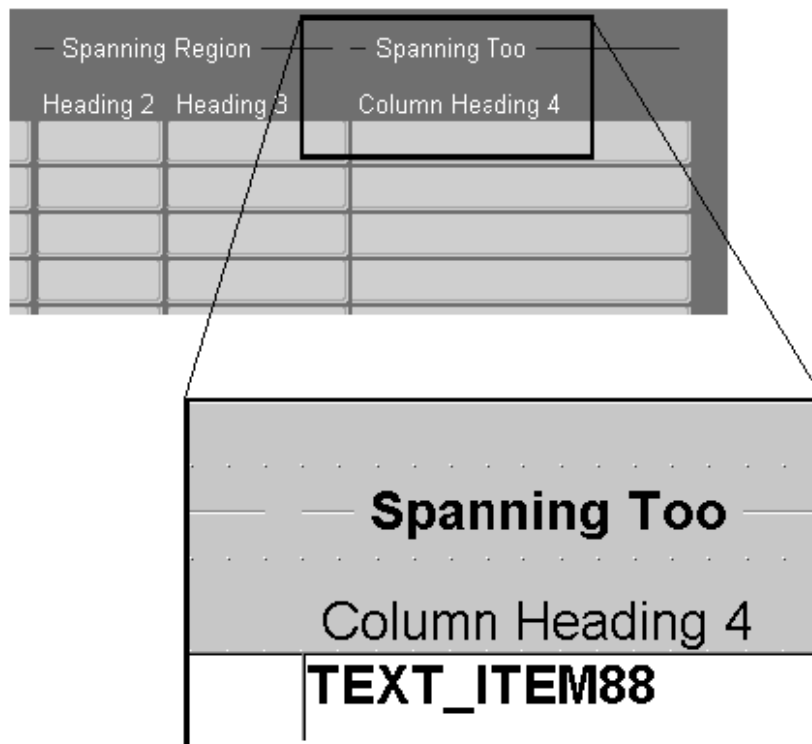
entry mode. (OMS-73250)

Frames in Multi-Record Blocks

Appearance

Frames in multi-record blocks should have the following look:

- A zero-height frame (looks like a line) is drawn in the row immediately above the region, spanning the fields of the region. The frame title is used to show the label. (OMS-73119)
- When two or more regions are drawn contiguously, reduce the width of the frame denoting the leftmost region by one character on the right side. The following picture illustrates this (OMS-73120):



Regions that Scroll

Scrolling to additional fields is not desirable, because it involves complex hand-eye coordination, and can be very frustrating if commonly used fields are always out of sight.

Important: Avoid scrolling to additional fields whenever possible.

Scrolling of regions should only be employed under the following conditions:
(OMS-73220)

- Only a few fields will not fit within the space of the region, and they are used less frequently than the visible fields.
- Any other division of the fields into regions is illogical or clumsy.
- Folders require scrolling.

If a region must scroll, then the following rules apply:

- Avoid making the underlying region more than two times the size of the boundary. In other words, a region should never scroll more than two times the width (in multi-record blocks) or height (in single-record blocks) of the visible area.
(OMS-73122)
- A region that scrolls must show a scroll bar. (OMS-73123)
- If any one of the multiple regions of tab scrolls, then all of those regions should have scroll bars. The scroll bar is visibly disabled for the regions where there is nothing to scroll to. (OMS-73124)
- Do not scroll the primary identification fields. (OMS-73125)
- Regions in multi-record blocks scroll horizontally. Regions in single-record blocks may scroll vertically, but this is discouraged unless absolutely necessary.
(OMS-73126)
- In a multi-record block, regions that scroll are separated from the static fields by a one character gap. (OMS-73127)

Tab Regions

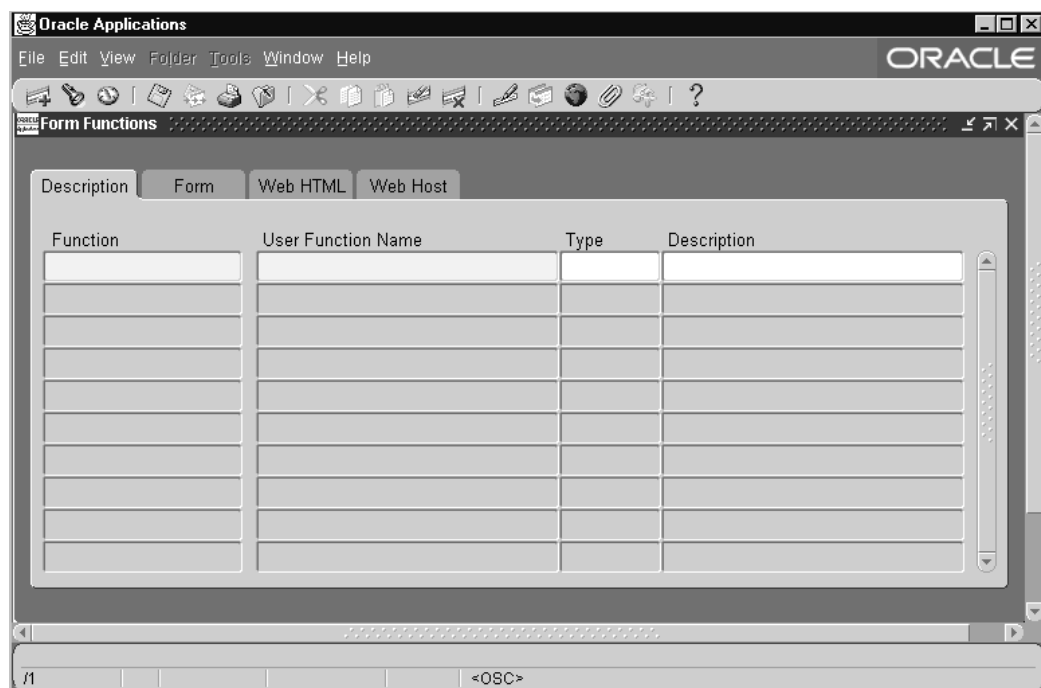
A block with multiple regions containing many fields and controls that cannot be displayed simultaneously uses a series of tabs to display each region one at a time within a single boundary. The tab control area usually spans the entire width of a given form. The block content should be logically divided among these regions such that the title of each tab reflects the grouping of items within it. In prior releases of Oracle Applications these were referred to as "Alternative regions" and were controlled by a poplist.

The tab control is provided by a widget that positions the tab UI mechanism at the top of a set of regions and allows the user to navigate directly to a specific region by selecting one of those tabs. The region then is redisplayed to reflect the user's choice by

raising the selected tab to the front.

Important: A note about conversion: Forms that previously used an alternative region poplist control should all be converted to tab controls. The major exception is the case where a poplist has been used to control a list of available queries (this is referred to as a "Query Control Poplist"). This is not considered true alternative region behavior since the contents of each region also reflect a change in the block content, not just the portion of the block being shown. In this case, leave the poplist control in place and add the label "Show:" in front of it. If the situation is further complicated by the need to show alternative regions within the query, then consider separating the controls, placing a tab region within the poplist control.

An example of a tab region (the multi-row case) is displayed in the following figure:



Layout

- Gaps must be left around a tab control, both inside and outside of it. Ideally these are 0.1" horizontally and 0.25" vertically.
- In single row blocks, place only the fields and controls related to each region within the tab region. All other fields (such as context fields) should be rendered outside the tab region. Ideally, these other fields are only placed above and below the tab region so that the tab region can extend the full window width. Only in exceptional

cases should there be fields to the left and right of the tab region.

- In multirow blocks, all of the fields should be within the tab region, including any overflow fields below the multirow portion.
- If the page also contains context or fixed fields that do not change and relate to the contents of every tab, these can be placed above and below the tab region. If it is necessary to place fixed fields within tab regions, they should be repeated in the same position on every tab.
- Controls, such as buttons that pertain only to a particular field or tab region, should be rendered on that page only. Controls that pertain to all of the tab regions are rendered outside of the tab region.
- If any one of the tab regions requires scrolling, all regions should have a scrollbar even if not all the tabs require scrolling. Oracle Forms will automatically disable the scrollbar for the regions where it is not needed.
- The label of each tab region should be short and concise. Oracle Forms automatically sizes the tab label based on the content.
- If the block has a coordination check box, it should be placed above the tab region and aligned to its right edge.
- If a tab region is the second block in a window, it is acceptable to remove the block frame and to leave the coordination check box "floating" above the tab region. If you do not have a block frame above the tab region you may include the block name in the tab labels if appropriate.
- Avoid placing a tab region within another tab region.

Behavior

- Selecting a tab causes the region associated with that tab to be shown, and the tab itself changes appearance to indicate that it is now in front and attached to the visible region.
- A Tab control list can be activated from the keyboard. Choosing an option from the tab control list performs the same action as selecting a tab directly, with the added behavior that the tab will be scrolled into view if it is not currently visible. This feature is provided automatically by Oracle Forms.
- If the cursor is in the block governed by the tab region, then selecting a tab moves the focus to the first item in the currently displayed tab region.

Note: If the cursor is on any of the fixed fields within a tab, then

selecting that tab merely shows the appropriate new fields and leaves the cursor position the same relative to the fixed fields. If the cursor is not on a fixed field, or if there are no fixed fields, then the focus should move to the first field on the block within the tab, and the fields of the new region should be displayed.

- If the cursor is in some other block or field not within a tab region, then selecting a tab merely shows the fields on that tab but does not change the cursor location.
- As the user presses the [Tab] key, navigation cycles between all the fields of the block, across tabs as appropriate.
- The selected tab must always be synchronized with the current region that is being displayed.
- In special cases where, due to installation, security, or other conditions, one or more tabs should not be shown to the user, the inapplicable tabs should be hidden. It is acceptable to hide tabs even if only one tab remains visible.
- Tabs should be dynamically disabled and enabled if their state is determined by data within each record. Note that due to timing issues you may have to account for a tab being selected, even though it should be disabled. For example, if a user enters a specific value in a field that would cause a tab to become disabled, but presses the tab before the field is validated, the tab is still enabled. In this case, you must validate the field that controls the condition, and, if the tab subsequently becomes disabled, then reverse any actions that occurred by the user having pressed it.
- While in QBE mode, the tabs should remain operable. Care must be taken to account for the current block and the first queryable field on each tab.
- *[Translation]* Tab labels should not include access keys if the application will ever be translated.

Related Topics

Overflow Regions, page 4-40

Widgets

Overview

This section describes the standard properties for the various types of objects that a user interacts with. For details on implementing these widgets, see the *Oracle Applications Developer's Guide*.

The topics covered in this chapter are:

- General Properties, page 5-2
- Text Items, page 5-2
- Display Items, page 5-5
- Poplists, page 5-7
- T-Lists, page 5-9
- Option Groups, page 5-9
- Check Boxes, page 5-12
- Buttons, page 5-14
- Lists of Values (LOV), page 5-17
- Descriptive Flexfields, page 5-19
- Key Flexfields, page 5-21

General Properties

- Each object should be displayed consistently with the same widget, prompt, and general properties everywhere the user will encounter it. Specific exceptions are noted in this chapter.
- In forms that are designed for high speed heads-down data entry, avoid any widget which cannot be operated from the keyboard and without looking at the screen. For example, a check box can be operated from the keyboard, but the user must look at the screen to see the current value because they cannot explicitly set it to checked or unchecked.
- Items are always sequenced left-to-right, top-to-bottom, within a region, block, or window (OMS-71005).
- For accessibility requirements, all functionality must be available from the keyboard.
- The colors of objects are usually controlled automatically by the Oracle look and feel. Only in exceptional cases should these be overridden.

For more information on general properties of widgets, see the *Oracle Applications Developer's Guide*.

Text Items

The following rules apply to text items.

Usage

- When any text is valid.
- When data is validated from a list that may contain more than 15 entries.
- When a field displays textual data that the user cannot alter, but the field must support querying or scrolling.

Recommended Alignment Rules

- Textual values should be Start aligned.
- Date values should be Start aligned.
- Numbers that do not display consistent precision, such as identification or phone numbers, should be Start aligned.

- Numbers with consistent precision, such as quantities and prices, should be Right aligned.

Note: When displaying values "Right Aligned" you must ensure that the field is large enough to display the largest possible value. Right-aligned fields do not allow scrolling and numbers that are not fully visible will not display the most significant digits of the number.

Rules

- A text item numeric field should be wide enough to display the largest possible value for the field. This ensures that the value is not truncated when displayed which could cause the value to be misread by the user. If the field is right-aligned then it must be wide enough to fully display the value. (OMS-70011)
- A text item date field must be 1.2" wide. This allows 11 characters to be displayed plus one cell to display the bevel. (OMS-74001)
- A text item time field must be 0.8" wide. The field can be 0.6" wide if seconds are not important. (OMS-74002)

- A text item used as a datetime field must be 1.7" wide. (OMS-74003)

Note: For date, time, and datetime fields, masks will be read from environment variables. Never supply a format mask for these field types.

- A text item used as a percent field may display percents with a fixed decimal point with a precision of two (preferred). (OMS-74005)

A text item used as a percent field may display percents with a floating decimal point, if extended precision is required. (OMS-74004)

- Text items should allow mixed-case entry, unless there is a business need to enforce either upper or lower case. (OMS-74006)
- If a text item must show multiple lines, then always enable the scroll bar and set the wrap style to "Word." (OMS-74007)
- A text item that must fully show between 1 and 10 characters should be sized one average character wider than the maximum contents to guarantee that the contents will be fully visible. (OMS-74008)

Example: Suppose a text item must fully show three letters. The text item should

then be sized to display four characters (0.4"). If the text item is sized to display only three characters, a word such as "WOW" that contains wide characters will not be fully displayed. Adding the extra character width to the field will also allow for the additional space used by the field bevels.

Cosmetics

- All text items are displayed as follows (OMS-74009):
Height = 0.25"
Inter-row Spacing = 0.0"
Bevel = Lowered
- *[Translation]* To support bidirectional languages such as Arabic which read right to left, you must use "Start" instead of "Left" when setting the alignment property. The alignment properties "Center" and "Right" can be used where appropriate. (OMS-74011)

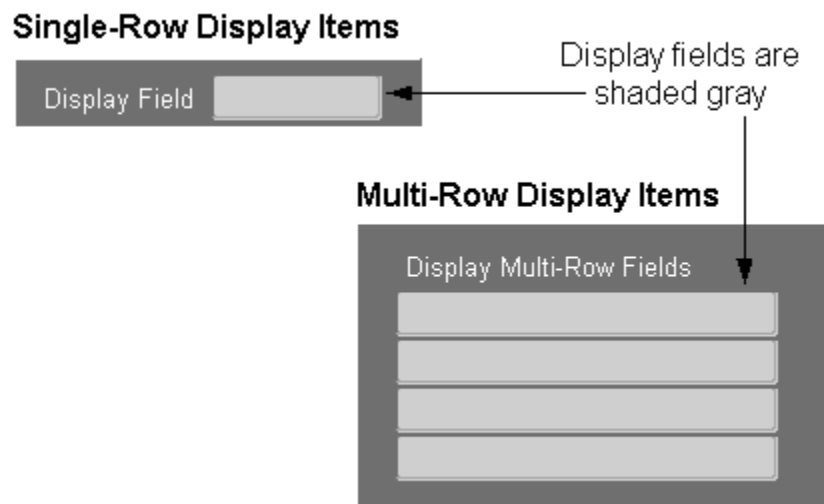
Behaviors

- Text items with an LOV associated with them almost always employ "Validate from List." This allows a user to type a partial value into the field, and it will autoreduce against the list of valid values.
- A text item with an LOV associated with it automatically renders the LOV icon next to it upon taking focus.
- Text item colors are as follows:
If editable and required, then with a yellow background.
The yellow color only appears when the item is in the current record; otherwise it has a white background.
If editable and not required, then with a white background.
If not editable, then with a gray background.
- Text item date and datetime fields must enable the List lamp. Invoking LOV or Edit Field on these fields opens the Calendar window. (OMS-74012)
- A multiple line text item should ideally be sequenced last in a block because of the behavior of [Tab], [Return], [Up Arrow] and [Down Arrow] within the field. (OMS-74013)
- A text item that cannot currently be validated due to a dependency on another field, or cannot be currently entered for any other reason, must have its insert, update,

and navigable properties turned off. (OMS-74014)

- When disabling a text item, make sure that any related fields (such as a field that holds a description) are also disabled. (OMS-74015)
- Text items that hold free-form text such as descriptions and allow querying use "Case Insensitive Query" to allow users to access data without regard to case unless performance would be unacceptable. (OMS-74016)
- In general, only fields that allow input should be included in the [Tab] sequence. However, for a display-only item (such as a description field) that a user may need to scroll, make the field navigable and skip over it in the forward [Tab] sequence but allow [Shift] [Tab] to navigate to it. That way if a user using only the keyboard needs to get into the field to scroll it, they can back tab to it. If a field which the user may need to scroll comes after all enterable fields, it must be made navigable (OMS-76006).
- On inquiry forms, where all fields are display-only, all fields should be navigable. (OMS-74017)

Display Items



Usage

- Only use display items for fields that never require the user to interact with them in any way, including scrolling or querying. (OMS-74018). This will generally restrict their use to:

- context fields
- fields that are sized such that scrolling would be unnecessary (such as total fields)
- fields that may display truncated information, but some other mechanism (such as overflow fields) exists for the user to see the entire contents of the field
- dynamic boilerplate

Rules

- Display items may be used to simulate dynamic boilerplate when the Prompt attribute is not adequate.
- *[Translation]* The width of a display item must be set to display its contents fully and to accommodate translation expansion; or, the item width may be sized for the current text length at runtime. (OMS-74019)

Cosmetics

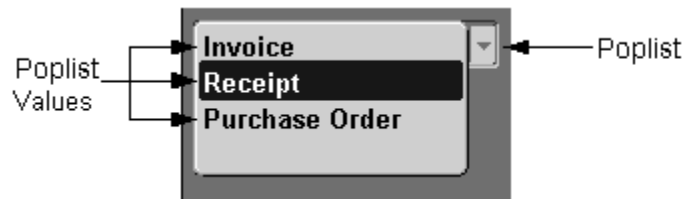
- (OMS-74020) When Display items hold data, they are displayed similarly to text items. See: Text Items, page 5-2.
 - Height: 0.25"
 - Inter-row Spacing: 0.0"
 - Color: gray background
 - Bevel: Inset
- When Display items are used as boilerplate, they are displayed as follows (OMS-74022):
 - Height: 0.2"
 - Inter-row Spacing: 0.0"
 - Color: background matches the current canvas color
 - Bevel: None
 - Y Position: 0.05" below the nearest gridline
- *[Translation]* To support bidirectional languages which read right to left, use the

following rules to set alignment properties (OMS-74021):

- Use "Start," "Center," or "Right" for display items that show data.
- Use "Start," "Center," or "End" for display items that act as boilerplate.

For information on implementing Display Items, see the *Oracle Applications Developer's Guide*.

Poplists



Usage

- Use poplists when only one value is applicable, and the list of choices is never expected to grow beyond 15. (OMS-74023)
- In exceptional cases, poplists may be used as field prompts in single-record blocks.
- A poplist may act as a Query Control Poplist.
- Do not use a dynamic poplist when possible values may become inactive. If a value is validated by an effective date or by an active flag and becomes inactive in a dynamic poplist, the user will not be able to retrieve records with that value until the value becomes active again. Use a Text Item field instead. (OMS-74523)

Rules

- An attribute modeled with a poplist in an entry form may be modeled with a text item in an inquiry-only form. (OMS-74524)
- In a Find window, include a blank row in the poplist to allow the user to specify that any value returned for the poplist during a search should be accepted. This blank row appears automatically if the poplist is made optional. (OMS-74024)
- In a Find window, include "(No Value)" in the list of valid choices if the poplist was optional when it was used for entering values on the form. If "(No Value)" is a choice, it should be placed last in the list of choices and will then appear immediately before the blank "any value" choice (which automatically shows up at

the bottom of the list). This allows the user to search for a "null-valued" row. (OMS-74025)

- **[Translation]** A poplist requires a width of 0.5" just to support the cosmetics of the widget. To adhere to translation requirements, the actual minimum width of a poplist is therefore 1.5." An exception to this is a "Yes/No" poplist which can be 1" wide. (OMS-74026)
- **[Translation]** A poplist is restricted to displaying 30 characters. To adhere to translation requirements, English text should not exceed 23 characters. (OMS-74027)

Cosmetics

- Poplists that hold data are displayed as follows (OMS-74028):
 - Height: 0.25"
- Poplists that are used as field prompts in single-record blocks are displayed as follows (OMS-74030):
 - Height: 0.25"
 - Font Weight: Medium

Behaviors

- A poplist that cannot currently be validated due to a dependency on another field, or cannot be currently entered for any other reason, must have its insert, update, and navigable properties turned off. (OMS-74130)
- Poplist colors are as follows:
 - editable and required: yellow background
The yellow background only appears when the item is in the current record
 - all other cases: gray background
- A Query Control Poplist must either be navigable, or its values must be presented in an LOV invoked from the Block Menu function.

For information on implementing Poplists, see the *Oracle Applications Developer's Guide*.

T-Lists

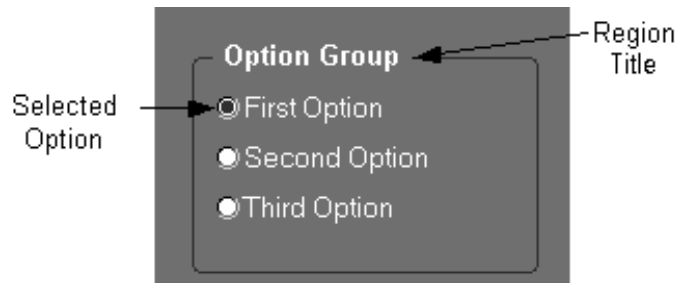
Usage

- Use T-lists when only one value is applicable, the list of choices is never expected to grow beyond 100, and no entry is expected to exceed 30 characters when translated. (OMS-74131)

Rules

- Always show at least five rows of data. (OMS-74132)
- T-lists should only be used in low-volume forms with extensive screen space available. (OMS-74133)
- An attribute modeled with a T-list in an entry form may be modeled with a text item in an inquiry-only form. (OMS-74134)
- T-list colors are the same as Text Items.

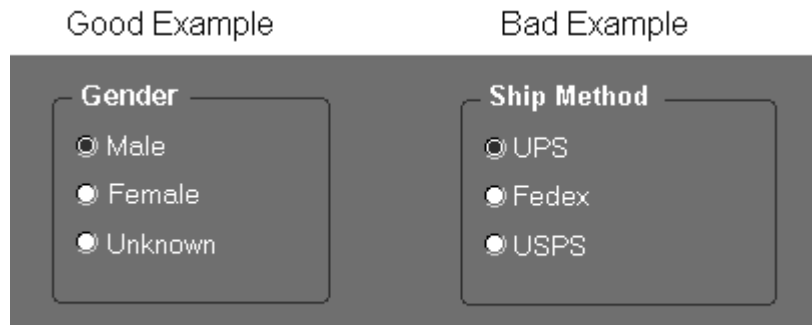
Option Groups



Usage

- Used when only one value is applicable, there are only two to four possible values, and the list will be static throughout the life of the product. (OMS-74031)

For example, selecting sex or gender (with three choices: Male, Female, Unknown) is an appropriate use of an option group because it is a limited selection that would never change.



An inappropriate use of an option group would be as a Ship Method selection (with choices UPS, Fedex, USPS). This example is a bad use of an option group because it is inappropriate for customer-defined lookups (different customers will use different shipping methods).

Rules

- Avoid using option groups for items that may ever appear in multi-record blocks.
- An option group must always have a default value. (OMS-74032)
- Use in place of a check box when the two states are not accurately modeled as yes/no.
- An option group must always specify its title in the Hint Text property. This text must replicate any title drawn on the screen, and must exist even if no title is drawn.
- Each single-row option button must specify a Label.
- Each multi-row option button must specify a Prompt.
- An option group can be used to set a "mode" of a form, such as what type of information should be displayed.
- An option group can also be used to indicate progression of data through various states, such as a Sales Order moving from "Booked" to "Shipped" to "Billed" to "Paid."
- An option button should have an access key. (OMS-74232)
See: Access Keys (Mnemonics), page 6-10
- **[Translation]** The minimum width of an option group button is 0.3," which displays

just the button with no label. However, in a single-row display, the minimum width is the larger of 1.3" or 30% longer than the label to allow for translation. Always allow as much space as possible for expansion. (OMS-74033)

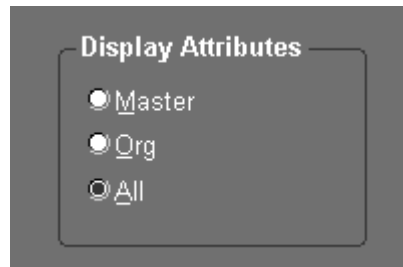
Cosmetics

- Draw the buttons of an option group in their own region, where the name of the item is the title, and the individual buttons are labeled elements within the region. (OMS-74034)

EXCEPTION: If the title of an option group is obvious, then the region title and boundary may be omitted. This should only be done if necessary, however, because the boundary helps to indicate that the options are mutually exclusive and visually different from check boxes.

- Option buttons may be laid out vertically or horizontally. (OMS-74234)

Vertical layout allows more space for translation and is easier to read.



Horizontal layout requires less space and the keyboard access method is more intuitive (the left and right arrow keys are used).



Querying a row with "invalid" option group data

For attributes modeled as option groups, there may be situations where a queried row will contain data that is not a valid choice of options. If such a situation may occur, create an additional option group button labeled "Other," and set it to be disabled. This allows the record to be queried, but prevents a user from setting the value to anything but a valid value. (OMS-74235)

For information on implementing Option Groups, see the *Oracle Applications Developer's Guide*.

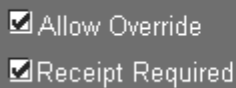
Check Boxes




Usage

- Used when only one value is applicable in a yes/no situation, and the yes/no statement is not contrived or obscure.

Good Examples



Bad Examples



Examples of good uses of check boxes include Allow Override and Receipt Required, where the yes/no response is clear.

For example, the use of a check box with the label Male is contrived to represent the Male/Female choice as a question with a yes/no response, so this is an inappropriate use of a check box (it should be an option group). A second example, with the check box labeled Root Menu, demonstrates inappropriate usage of a check box because the opposite of Root Menu is not obvious.

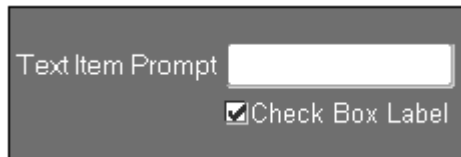
In each of these bad examples, an option group or poplist item would present a more intuitive set of choices to the user.

Rules

- A check box item is mandatory, and must always have a value (including a default value). (OMS-74035)
- In a Find window, an attribute normally modeled with a check box is modeled with a poplist. The list must include values for the checked and unchecked states and should not be required so that a null entry appears as well. (OMS-74036)
- **[Translation]** The recommended minimum width of a check box is 0.2," which displays just the check box with no label. In a single-row display, the minimum width is the larger of 1.3" or 30% wider than the label, to allow for translation. Always allow as much space as possible for expansion. (OMS-74037)

Cosmetics

- The check box itself is normally positioned as a text item would be.

A screenshot of a form with a dark gray background. It contains a text input field with the placeholder text "Text Item Prompt". Below the input field, there is a checked checkbox followed by the text "Check Box Label".

- When the label of a check box is very long, the check box may be positioned starting at the left end of the prompt of the field above.

A screenshot of a form with a dark gray background. It contains a text input field with the placeholder text "Text Item Prompt". Below the input field, there is a checked checkbox followed by the text "Very Very Long Check Box Label". The checkbox is positioned at the start of the line, aligned with the left side of the prompt.

In the above example, the check box and prompt are aligned along their left sides. This alignment may of course not hold after translation to other languages.

- Check boxes in multi-record blocks are 0.2" wide. (OMS-74038)
- For a single-row check box, the Label must be used. For a multi-row check box, the Prompt must be used.

Entering at least one selection

In certain cases, the user has a choice of options to choose from and may select more than one, but must select one at a minimum. If only two check boxes are available and the user deselects the only selected item, the other item should be selected automatically. A confirmation point may also be used later in the form to ensure that at least one item is selected. (OMS-74238)

For information on implementing check boxes, see the *Oracle Applications Developer's Guide*.

Coordination Check Boxes

Coordination check boxes are used to control master-detail relations between blocks. See Master-Detail Characteristics, page 4-35

Buttons

Usage

- Used to initiate an action, such as a product-specific function, or block-to-block navigation.
- Only use buttons for the most commonly needed functions in a form. Put other functions in the Special menus.

Rules

- Try to use only one row of buttons per window. (OMS-74239)
- Provide one default button per window, where that function is the most likely for the user to perform. Always provide a default button in a modal window (typically "OK"). (OMS-74039)
- Sequence buttons within a window as follows:



- Place the default button to the left of all other buttons.
 - Place the Cancel button immediately to the right of the default button to form a group.
 - Place all other buttons in a group to the right.
 - EXCEPTIONS:
If the Cancel button is the default button, do not reposition it within its group.
In cases where there are other actions which are logically related to the default button, place those buttons immediately to the right of the default button. Place the Cancel button in the rightmost position of the group.
- Follow the same button sequence rules for modal windows except when the modal window includes a Help button. Place the Help button in the leftmost position.



- Buttons within a window should be sized similarly and spaced consistently, except when using gaps to group related buttons. If a window contains one button with a particularly long label, but all the other buttons have short labels, only size the short-labeled buttons the same.
- Buttons may be placed on tab regions if they only apply to the contents of that specific region.
- The default button should never be placed in a tab region where it can be accessed by keystroke when not visible.
- The right edge of the rightmost button must be 0.1" from the right edge of the window. In general, leave 0.1" between buttons. If there are logical groupings of buttons, leave 0.5" between the groups (and still place 0.1" between the buttons in each group). If horizontal space does not allow 0.5" between groups of buttons, then leave as much separation as is possible. (OMS-74041)
- **[Translation]** A button has a minimum width of 0.2" just to support the cosmetics of the widget. To adhere to translation requirements, the actual minimum width of a button is therefore 1.2." See the section on Text for more information on expansion requirements for button labels. (OMS-74042)

See: Text, page 6-1

- In general, buttons are navigable except for the following: (OMS-76013):
 - Buttons below a multi-record block
 - Buttons that only apply when the cursor is in a particular field
 - The "Clear" button in a Find window
- All buttons have an access key except "OK" and "Cancel" within dialog windows. "OK" must have an access key if it is not the default button, and "Cancel" must have an access key if it does not perform the same function as "Close Window." (OMS-74043)

See: Access Keys (Mnemonics), page 6-10

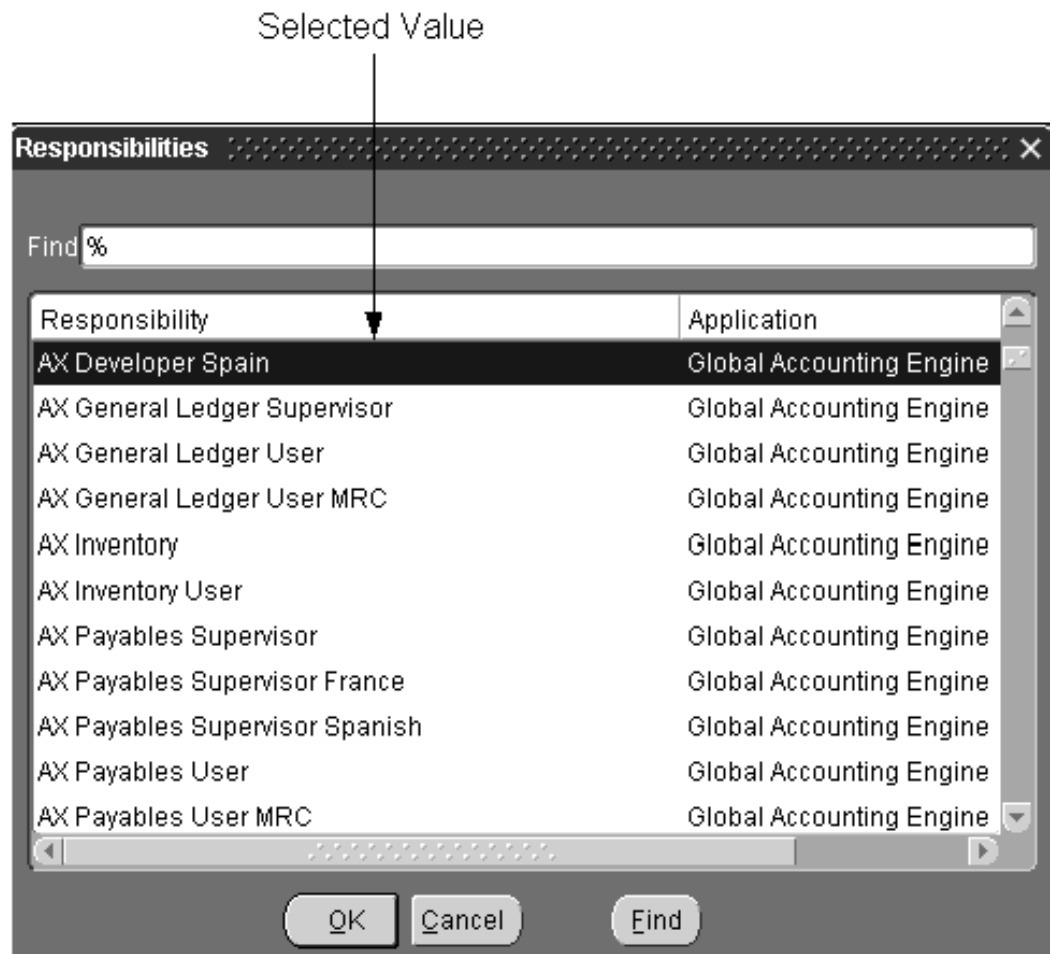
- All but the simplest modal windows should have a Help button in the lower left corner.
- All iconic buttons must provide Tooltip text.

Cosmetics

- Textual Buttons are displayed as follows (OMS-74044):
 - Height: 0.3"
 - Rounding: Buttons that have the same Y-coordinate, and are spaced within 30 pixels of each other, have only their outermost edge rounded. A button with no other button near to it has both edges rounded.
- Iconic buttons within a screen are either 0.25" by 0.25" or 0.3" by 0.3."

For information on implementing Buttons, see the *Oracle Applications Developer's Guide*.

Lists of Values (LOV)



Usage

LOVs are used when the user must select from a list of valid values in a text field.

Rules

- All LOVs open near the item they are associated with, or if there is no specific item, they open centered in the MDI window. (OMS-74045)
- LOV columns are set to use Autosize, which ensures that the column width can fully display its title.
- Width: Maximum: 7.8" Minimum: 3" Any width between these is allowed. (OMS-74046) However, the following guidelines are recommended:

- **[Translation]** The column width in the LOV should match the length of the field in the window.
- The LOV window should be smaller than the window from which it was invoked so the LOV does not totally obscure it.
- The title is the name of the object in the LOV, and is plural, for example, "Customers." For a "Row LOV," where the user chooses a particular record, the title is "Find" plus the name of the entity (with the name of the entity in singular). For example, a Row LOV to find Purchase Order Lines would have a title of "Find Purchase Order Line." (OMS-74047)
- The prompt of the first column is related to, or matches identically, the prompt of the item that invoked it. (OMS-74247)
- **[Translation]** In an LOV, display the value of a true/false value as "*" for true and <blank> (no value shown) for false. This is easier for a user to read quickly than words like "True" or "Yes," and simplifies translation. (OMS-74248)

Cosmetics

All LOVs use the default visual attributes. (OMS-74249)

Behaviors

- LOVs do not display automatically upon navigating to a field. (OMS-74250)
- LOVs automatically select a row when the list of valid choices is reduced to one. However, an LOV that is not associated with a text item (such as one that is invoked by pressing a specific button), should not automatically select a row. (OMS-74451)
- (OMS-74452) After selecting from an LOV, the cursor automatically moves to the next field except when:
 - the next field is in a region not currently displayed.
 - the field is a mirror item of a field in Folders, like in Combination Blocks.
 - the field is the last navigable field in a multi-row block with horizontal scrolling.
- If an LOV may show more than 100 records, then the user must be prompted to reduce the list of valid values first. (OMS-74048)
- In normal (entry) mode, an LOV shows only values that currently can be selected.

EXCEPTION: Validation can be performed after-the-fact if any of the following apply:

- The validation clause cannot be written in a single SQL statement.
- The validation clause is too costly to evaluate in a single SQL statement.
- The reason for exclusion from the list is obscure to the user.

In such cases, after the value is selected show an error message indicating exactly why the value could not be selected.

- Do not provide LOVs in QBE.

For information on implementing Lists of Values, see the *Oracle Applications Developer's Guide*.

Editors

The Editor is invoked by pressing the Edit Field toolbar button, or by selecting Edit Field from the pull-down Edit menu. In response to these actions, Oracle Applications shows the default Forms editor window, or a user-specified editor (see the *Oracle Forms Installation Manual* for more information on specifying an editor).

On a Date field, the Edit Field action opens the Calendar. (OMS-74350)

On a Flexfield, Edit Field opens the Flexfield window (OMS-74058).

Descriptive Flexfields

Usage

All entities should provide a descriptive flexfield to allow customization. (OMS-74049)

Cosmetics

- Code a descriptive flexfield as a text item, displaying two characters (width of 0.2"). (OMS-74050)
- The prompt associated with the descriptive flexfield is "[]".
- In a single row block, the "[" is the prompt, and is drawn .05" to the left of the item and centered vertically. The "]" is boilerplate placed .05" to the right of the item, also centered vertically. (OMS-74051)
- In a multi-row block, the prompt is "[]" drawn .05" above the item and centered horizontally.

- The descriptive flexfield is located as the last item in each block on the content canvas. When regions exist, the descriptive flexfield is located after the region (not as an item within a particular region). (OMS-74052)

In exceptional cases, where the aesthetics of the single-row block are compromised by locating the descriptive flexfield last, it may be located elsewhere, but should always appear at the "end" of a group of fields (for example, as the last field of the context canvas fields before an alternative region). Regardless of its location, it is always the last sequenced item of the block.

Behaviors

- The descriptive flexfield Forms field should be navigable, but should not allow input. It adheres to all text item standards. (OMS-74053)
- Flexfield code will automatically show the concatenated values in the descriptive flexfield Forms field.
- Descriptive flexfield uses a user-level profile that determines whether the flex window should pop open upon the user navigating into the field. (OMS-74054)
EXCEPTION: In folder blocks, the flexfield window must not automatically open because this would prevent the user from resizing the field.
- At form startup, the descriptive flexfield is disabled if the customer has not activated the flexfield by defining and enabling segments.
- If the user invokes the descriptive flexfield before the independent fields of context-sensitive segments are populated, then the following will occur:
 - If there are global segments, the window will open, and text will display indicating that other context-sensitive fields cannot currently be populated.
 - If there are no global segments, then no window will open and a message will indicate that the context-sensitive fields cannot currently be populated.
- Behavior of the descriptive flexfield window itself :
 - Any successful navigation out of the window moves the cursor to the next field.
 - Canceling the window moves the cursor to the previous field.
 - During validation, if mandatory segments are missing, then flex will issue a message. In the cases where it can be done, the flex window will open automatically upon the user acknowledging this message; in those that it cannot, the user will have to invoke the window.

For information on implementing descriptive flexfields, see the *Oracle Applications*

Key Flexfields

Usage

For all attributes that can be modeled as a multi-segment key, such as Item Number or Account.

Behaviors

- All key flexfields use the direct-entry method. (OMS-74055)
- Only key flexfields with Validation set to Full should enable the List lamp. (OMS-74056)
- Invoking a List of Values in a flexfield allows for entry of a combination-level List of Values. (OMS-74057)
- Invoking Edit Field in a flexfield expands the field to allow entry of values in individual segments. (OMS-74058)

EXCEPTION: For a flexfield that allows dynamic inserts, Edit Field and List invoke the same window because it is irrelevant to the user whether a particular combination has ever been entered before.

Cosmetics

- Do not code any special visual indicator or activation button for key flexfields. The field should look like any other text item. (OMS-74059)

For information on implementing Key Flexfields, see the *Oracle Applications Developer's Guide*.

Overview

This chapter describes rules for text, with special considerations for translation requirements. Conventions for displaying messages to the user are also described. Specifically, the topics covered in this chapter are as follows:

- General Properties, page 6-2
- Properties of Prompts and Titles, page 6-4
- Single-Record Block Prompts, page 6-5
- Multi-Record Block Prompts, page 6-6
- Conventions, page 6-8
- Access Keys (Mnemonics), page 6-10
- Using Data as Prompts, page 6-11
- Errors, page 6-12
- Warnings, page 6-13
- Questions, page 6-13
- Information, page 6-14
- Hints, page 6-14
- Messages, page 6-15

General Properties

- Due to the use of proportional fonts, it is difficult to guarantee that a specific number of characters will always be visible in a field. When sizing widgets, assume that an "average" character has a width of 0.1." Each widget has additional sizing requirements described below.
- *[Translation]* (OMS-75001) All text (except text that has a known translated length) must be able to expand by at least 30%. Translation lengths for many common terms are listed in the section Terms with No Abbreviation, page B-4. This 30% expansion rule applies to the number of "average" characters, not inches. The 30% rule applies to different types of text differently, as listed in the following table:

Text type	Requirement
Prompt on left of field	Leave 30% expansion room to left of prompt.
Prompt above left -justified field or list	Leave 30% expansion room to right of prompt.
Prompt above right-justified field	Leave 30% expansion room to left of prompt.
Prompt above center-justified field or check box	Leave 30% expansion room, divided equally to the left and to the right of the prompt (15% on each side) .
Buttons	Size to allow 30% expansion of the label + 0.2."
Check boxes in single-record blocks	Size to allow 30% expansion of the label + 0.3."
Poplists	Size to allow 30% expansion + 0.5."
Option Groups	Size every button the same, such that the widest label can expand 30%.
Frame Titles	Size underlying frame such that title can expand to the right by 30%.

Text type	Requirement
Display-only fields	Size 30% wider than needed in English, except for fixed-width fields (for example, date or time).

- (OMS-75002) *[Translation]* All prompts must have a minimum of 1" (10 "average" characters) total space available in which to expand when translated. This rule may not apply if the translated length is known ahead of time.

The minimum of 1" is in addition to the 0.2," 0.3," or 0.5" necessary for the cosmetics of buttons, check boxes in single-record blocks, and poplists respectively. For example, a button with a short label must be at least 1.2" wide (1" minimum plus 0.2" for the cosmetics of the button).

- Text items that must fully show between 1 and 10 characters should be sized one character wider than the maximum width of the contents to guarantee that the contents will be fully visible. This rule does not apply for fields that will show only numbers or to fields that have standard lengths (the standard lengths should account for the extra size requirement) (OMS-74008).

Important: The two prior standards are commonly referred to as the 30%/1" rule. It is important to remember that these are the minimum spacing requirements. Whenever possible leave additional space for translation.

- (OMS-75003) *[Translation]* Items that show text but are not scrollable (such as poplists and buttons) must be coded to display the maximum expected length of text in all languages.

- (OMS-75022) All prompts and labels are in mixed case, except for the following, which are in lower case:

- Articles (such as "the")
- Coordinate conjunctions (such as "and")
- Prepositions (such as "with")
- The "to" infinitives

These words should, of course, be capitalized when they are at the beginning of the prompt or label.

- (OMS-75023) Errors, messages, questions, etc., that contain text that appears on the

form as labels or prompts should use the same capitalization conventions as the text on the form.

- Avoid abbreviations if possible. Some abbreviations have been approved for use in Oracle Applications and are listed in the section Abbreviations

If you have room, use the full (unabbreviated) version of the term. When using the full versions of these terms, it is not necessary to leave 30% extra for expansion as an approved abbreviation can be used upon translation if necessary.

- Every item must have Hint Text, a Label, a Prompt, or Tooltip Text associated with it (except special fields like current record indicators, which act as controls, not as fields). This is so that devices utilized by a challenged user can clearly communicate a label and value as focus moves.

Properties of Prompts and Titles

The following characteristics of prompts and titles apply:

- (OMS-75004) [*Translation*] Each prompt and title must have a minimum of 1" of space available horizontally (including the text itself) in which to expand when translated.

Only the following cases are exempt from this rule:

- Text for which exact translation lengths are known ahead of time.
- Prompts in folder blocks, which a user can alter and resize as needed. Such prompts only need to be sized properly for the base language they are developed in.
- (OMS-58012) All prompts and titles share the same font type - MS Sans Serif or Helvetica (either produces the desired result).
- (OMS-75006) All prompts have the following characteristics:
Font: 10 point, medium weight
Text color: Black when on a light background, or white when on a dark background
- (OMS-75007) All titles have the following characteristics:
Font: 10 point, bold weight
Text color: Black when on a light background, or white when on a dark background
- (OMS-75008) Prompts should only use Start, Center, or End for alignment settings.
- Display items may be used to show dynamic prompts and titles (OMS-74018). Guidelines for such use are as follows:

- *[Translation]* Size such a field so that when translated the text will not be clipped. Always allow as much space as possible but never less than the 30%/1" expansion rule minimum.
- The text must not be cleared when a user performs a "Clear Form" action.
- A display item acting as a prompt or title must have a height of 0.2," and be positioned 0.05" below a gridline (OMS-74022).
- Font and color must match standard prompt and title settings.
- A field with a display item acting as its prompt must have text supplied for either its Hint Text or a hidden Prompt (a Prompt with Display Style "Hidden").
- (OMS-75024) All fields require unique prompts to identify them, except for some display-only fields whose prompts are obvious. Examples are:
 - Adjacent Vendor Name and Vendor Number fields: the word Vendor can be omitted from the second field prompt.
 - Adjacent Item and Item Description fields: the displayed prompt for the second field can be omitted when the description field is display only:
 - Adjacent Amount and Currency fields can share the single displayed prompt of Amount if the currency type is defaulted:

For all fields that do not have a displayed prompt, text still must be available to uniquely identify the field. Either a hidden prompt or Hint Text must be supplied for each instance.
- Because a frame title is not linked to fields within the frame, all fields inside a frame must have Hint Text that combines the frame title and Prompt, in the format: .
- Fields that are part of a matrix layout, with prompts for the rows and columns, must have Hint Text that combines these prompts in the format: <row label>: <column label>.

Single-Record Block Prompts

- Prompts are positioned to the left of an item, with 0.1" between the rightmost character of the prompt and the start of the item (OMS-75009).
- Prompts should normally be drawn on a single line. In the case of a multi-line field, the prompt may occupy multiple lines but should still be a single boilerplate item (with a return between lines) and should not extend below the bottom of the field. Prompts may also be in multi-line format for T-lists. Multiple line prompts may also

be used for isolated fields that are not stacked immediately above or below another field.

- (OMS-75009) Specific settings are enumerated below:
 - Prompt Display Style: First Record
 - Justification: End
 - Attachment Edge: Start
 - Alignment: Center
 - Attachment Offset: 0.1"
 - Alignment Offset: 0.0"
- (OMS-75025) Prompts may be drawn above a field where items are part of a two-dimensional matrix.
- When a poplist is used as a prompt, the poplist should have a font weight of medium (OMS-74030). The background color must match the canvas color.

(OMS-75026) Each prompt value in the list should have a trailing colon (for example, "Name: "). In addition, once a value is selected from the poplist, the cursor should automatically move to the field that the poplist identifies, and clear the previous data if it is no longer applicable.

Multi-Record Block Prompts

- (OMS-75109) Prompts are positioned above the first record of each item.
- Prompts for text items are start, center, or end-aligned similarly to the data in their corresponding text fields, except in the case of long prompts that use connecting lines (OMS-75011) (discussed in the next section).
- Prompts for poplists are always left-aligned (OMS-75011).
- Prompts for check boxes are centered above the box, except in the case of prompts that use connecting lines (discussed later) (OMS-75011). Do not use the "label" property for check boxes in a multi-record block.

Long Prompts

(OMS-75027) There are several options when there is not enough space to fit the length of the prompt or the length of the prompt plus its translation allowance. Four options are discussed below, in order of most to least preferred:

- When prompts contain too much text to fit on one line, a multi-line boilerplate item should be used. Use a "return" character to advance to the next line of text, and divide the prompt into lines of approximately the same number of characters each.
- If a prompt is wider than its corresponding field, and a multi-line prompt cannot be used or still does not solve the problem, then leave a gap to the left or right of the field to accommodate the width of the prompt. Never widen a standard length field to fill this gap. If the field does not have a standard length, it should be widened to fill the gap, but the displayed length should never be made longer than the database length.
- Prompts may be connected to their fields with a line consisting of a frame of zero width.

For multi-record block prompts that must use a connecting line, align the prompt with the left side of the field and draw the connecting line from the leftmost character of the prompt to the leftmost character of the field (regardless of how the data in the field is justified) or to the center of the first character cell for a check box.

When the prompt cannot overflow to the right (for example, when the check box or field is on the right side of the block), align the prompt to the right side of the field or center of the check box and draw the connecting line from the rightmost character of the prompt to the rightmost character of the field, or to the center of a check box (in other words, exactly the mirror image).

In either case, the connecting line should be drawn in the center of the appropriate character cell and the origin of the prompt should be in the center of the same character cell (i.e. a half character cell (0.05") in from the appropriate edge of the field or in the center of a check box).

- *[Translation]* The prompt of a field may overlap an adjacent field if the maximum translated length of each prompt would still leave a 1" gap between them. This technique can only be used when a left-aligned and right-aligned field are adjacent, and one of them is substantially longer than the other.

Specific Settings

The following enumerates some of the specific settings for multi-record block prompts, including prompts over check boxes and prompts that use connecting lines.

- (OMS-75010) All prompts for multi-record blocks have the following vertical placement settings:
 - Display Style: First Record
 - Attachment Edge: Top
 - Attachment Offset : 0.0"

- (OMS-75011) Horizontal placement is determined as follows:

For left-justified data elements, including lists:

- Justification: Start
- Alignment: Start
- Alignment Offset: 0.05"

For center-justified data, including check boxes (except when using connecting lines with check boxes):

- Justification: Center
- Alignment: Center
- Alignment Offset: 0.0"

For right-justified data:

- Justification: End
- Alignment: End
- Alignment Offset: 0.05"

Note: The 0.05" offset of the prompt from the right or left edge of the field is done to account for the bevel width.

Identical Blocks

(OMS-75028) Occasionally two multi-record blocks are shown in the same window, one above the other, where the only difference is the set of rows they retrieve. In this case, you do not need to replicate the prompts for the lower block.

Conventions

Button Labels

Button labels should always be short and succinct. The following table contains examples of good and bad button labels:

Good	Bad
Lines	Enter Order Lines
Print	Run Print Report
Margins	Set Document Margins
Apply Notes	Automatically Apply Notes

Ellipses

(OMS-75012) Ellipsis points (...) are used at the end of a menu entry or button label in the following cases:

- The entry or button opens a modal window (for example, the menu entry "About Oracle Applications...").
- The user needs to provide further information about the action in another window (modal or not) before the action can be completed.

Button examples with ellipses:

- Copy to...
- Find Exceptions...

(OMS-75029) Ellipsis points should not be used at the end of a menu entry or button label if:

- the button is used to open a non-modal window that does not require further information before the current action can be completed.
- the button or menu entry opens a confirmation window because it invokes a potentially destructive or irreversible action, but does not require additional information to carry out its function.

Percent Fields

- (OMS-75013) Do not include the word "Percent" in the prompt - always use the symbol "%".

In any case where the percent sign is used, do not enclose it in parentheses or any other delimiters.

- (OMS-75014) Single-record percent fields should have the prompt before the field and a percent sign as boilerplate immediately after the field (no gap).
In this case, Hint Text must be added to the field, like "Prompt %".
- (OMS-75015) Multi-record fields have a prompt above the field such as "Prompt %":
- If the only reasonable prompt is "Percent," then only use "%" after the field or above the column. This occurs when the qualifying word is part of the region title.
Hint Text for these fields must be "Quota To Date: Total" and "Quota To Date: %".
- (OMS-75030) The data in a percent field should always be right-aligned.
- Percent fields should specify range limits of 0 to 100 if appropriate.

Ranges

(OMS-75031) Use the terms "From" and "To" to identify fields involved in a range rather than "Start" and "End" or "Low" and "High."

Note: If there is a standard industry term that is more appropriate for your product, use it instead of the generic "From" and "To" terminology. You may alternatively place the range fields horizontally with a single prompt (such as "Hire Dates") in front of the "From" field and a dash between the fields as in "Hire Dates ____ - ____" instead of spelling out the prompts From and To.

For the second field, Hint Text must be provided, such as "Hire Dates: To."

Access Keys (Mnemonics)

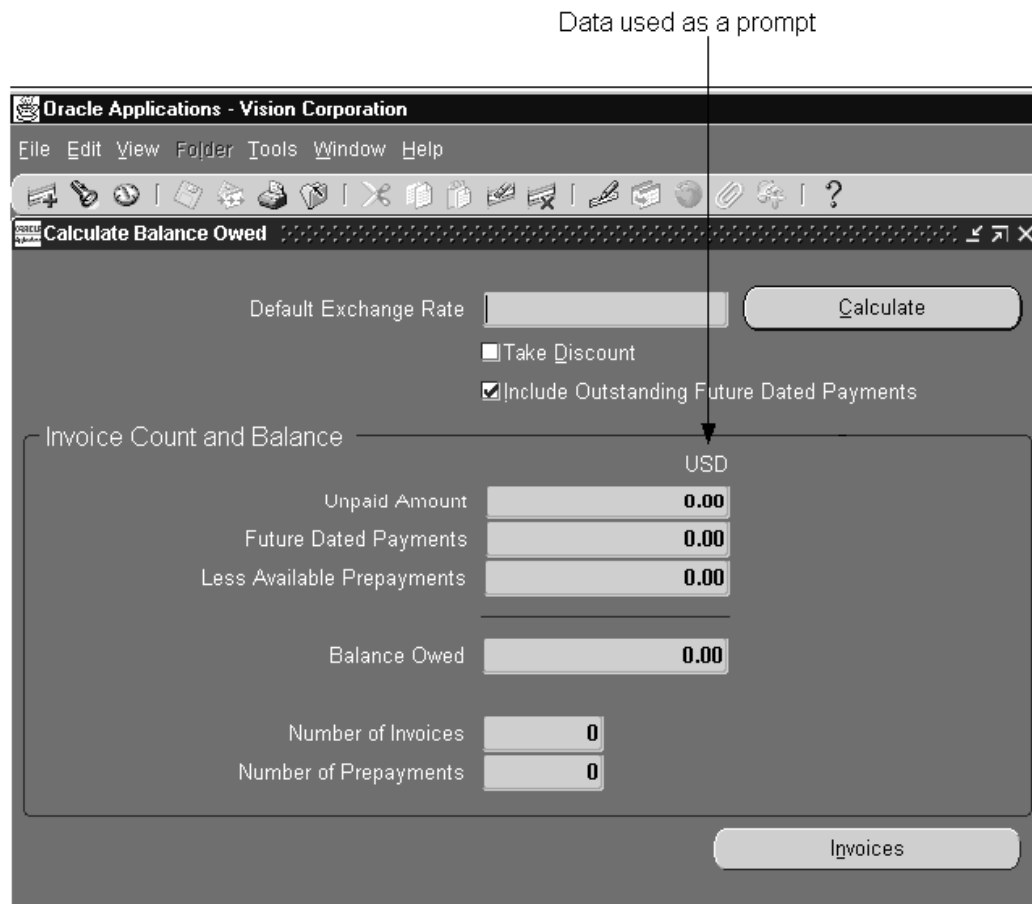
Controls that can be operated with direct keyboard access provide an access key (underlined mnemonic access character) to invoke them.

- (OMS-75016) Provide access keys on the following objects:
 - All textual buttons, except "OK" and "Cancel." However, these buttons must have access keys if OK is not the default, and if Window Close does not perform the Cancel action.
 - Option buttons and check boxes should have access keys. If the item is not navigable, then it must have an access key. Note that only a check box or option button in a single-record block can have an access key, because the label property is not used in multi-record blocks; therefore, in multirow blocks, these items must be navigable.

- (OMS-75032) In order of preference:
 - First letter of button or of key word.
 - First letter of the nonkey word.
 - Second letter of button label, such as C for Activate.
 - A strong letter of the label, such as X for Exit.
- Try not to underline letters with descenders such as y, j, q, or p because the underline and letter overlap (y, j, q or p).
- Follow common conventions where applicable.
- (OMS-75017) Access keys must be unique within a window, and must not conflict with the keys used by the pull-down menu (F, E, V, L, T, W and H) even if those menus are not accessible.
- Oracle Applications will automatically determine unique access keys within a window. The letter you supply for buttons and other widgets will be used unless it is already employed, in which case another letter will be selected at runtime.
- (OMS-75033) Use the following access keys for these common terms:
C for Clear, F for Find, C for Cancel, N for New, O for Open, and D for Done.

Using Data as Prompts

Occasionally the prompt for a field is a data value that is entered or derived elsewhere. For example, in an Accounts Payables screen, a comparison of values of balances can be made by specifying the exact rows of data to analyze, and the currency of interest. In this case, the currency code (such as USD) may be used as a prompt.



Errors

- Errors are messages that indicate serious problems which the user must acknowledge in order to continue processing.
- (OMS-75018) Error messages show a graphic depiction of an alarm bell, the message text, and the "OK" button.
Examples:
"Please enter a unique Territory Code."
"Field must be entered."
"You cannot delete this Customer because it is referenced on an Order."
"You do not have sufficient authority to approve this Order."
- (OMS-75034) Standard Forms function keys pressed at inappropriate times, where no explanation is necessary, should merely beep.

Warnings

- Warnings are messages in the form of a question that the user must respond to. Warnings allow the user to continue or terminate an action.
- Warnings should be direct and concise. For example, use "Delete this Order?" rather than "Do you really want to delete this Order?"
- (OMS-75019) Warning messages show a "yield" sign, the message text (ending with a question mark), and relevant buttons (such as "Yes" and "No" or "Delete" and "Cancel").
- Examples:
"Delete this Purchase Order?"
"Cancel this Order?" ("Yes" & "No" buttons)
"Copy all the Lines on this Invoice?"
"This customer is on Credit Hold. Continue the Order?"
- (OMS-75035) The standard generic delete message used in Oracle Applications is "Delete this record? (Cancel/Delete)" with Delete as the default button on the right. More specific delete messages are used where appropriate.

Questions

- Questions are used when the message is not intended to provide a way for a user to terminate an action. Questions allow for three-state decision making, such as "Yes," "No," and "Cancel." Such a message is displayed when the user must perform some other action before the desired action can continue.

(OMS-75020) Questions show a "question mark" icon, the message text (ending with a question mark), and the "Yes" and "No" buttons (or buttons specified by the developer).

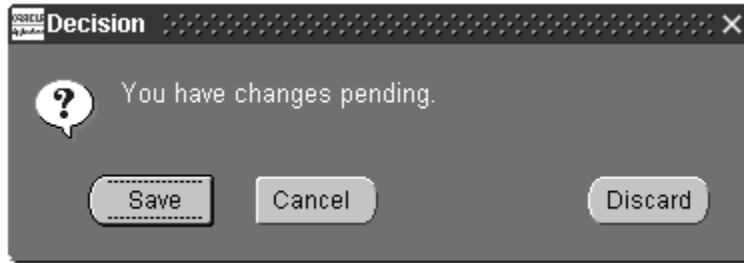
Example:

In response to clearing a block:

"Do you want to save your changes?" (buttons "Yes," "No," and "Cancel")

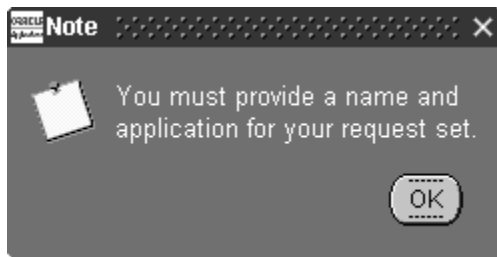
Note: The three-state message is used when the application is in a state where the desired action can be performed, but there is a prior step which the user most likely would want to perform. Selecting "No" in the case above will still cause the desired action to be performed. "Cancel" will result in the data not being cleared.

- Always provide specific button labels, rather than "Yes" and "No," when a clearer meaning can be expressed in one or a few words.



Information

- Information messages are messages that the user must acknowledge, but do not require the user to make any choice.
- (OMS-75021) Information messages show an "information" notepad, the message text, and the "OK" button.
- Examples:
 - "Line and Shipment Quantities currently do not match."
 - "There are items awaiting your attention."
 - "The Concurrent Request ID for this report is 123."
 - "There have been 3 unsuccessful login attempts since your last login."



Hints

- Hints are messages that are of very little consequence, or a "progress indicator" message, that never require any acknowledgement.
- (OMS-75036) Hints appear on the status bar only.
- Examples:

"Working..."

"Processed Order line 12 of 37..."

"At first record."

"At last block."

Messages

This section gives guidelines for writing messages.

- Make messages short and avoid redundancy.
- Do not use a Cause/Action format.
- Provide as much detail as the user needs to fix the problem, but no more than necessary.
- Use short but complete sentences. Use proper grammar, punctuation and capitalization.
- Avoid ambiguous words. Try to use words having only one meaning. Avoid words with data processing connotations unless you are referring to a specific application function.
- Say "please" wherever possible. When a message contains instructions, use please.
- Use vocabulary consistent with forms boilerplate. Refer to a field by its correct name. If a field is labeled "Sales Representative," don't use the message "Please enter a different salesperson."
- Address the user as "you." Talk to the user, not about the user. Users prefer friendly messages that address them directly and make them feel they control the application. "You" is also more concise and forceful than "The user...."
- Avoid nonspecific future tense. Use future tense only when your message refers to a specific time or event in the future. In other cases, "will" is usually ambiguous. For example, "Please select an invoice to cancel" is correct; whereas, "Please select an invoice that you will cancel" is incorrect.
- Use active voice when possible.
- Avoid accusatory messages. Do not insinuate that the user is at fault. Do not mention a user's mistake unless it pertains to the problem's solution.
- Use imperative voice. For example, the message "Enter a commission plan" is better than "You can enter a commission plan."

- Avoid conditionals. Use positive, imperative statements over statements containing conditionals.
- Use "can" to indicate either capability or permission. Auxiliaries such as "could," "may," and "might" are ambiguous and imply more uncertainty than "can." Limit the range of uncertainty by using "can," which always implies capability or permission but does not imply chance or luck. For example, the message "You cannot delete a printed release" is preferable to "You may not delete a printed release."
- Use only idiomatic abbreviations that match those used in your application's forms. If the forms that use a message do not abbreviate a term, do not abbreviate that term in a message.
- Try to avoid messages with multiple possibilities, such as "Value is invalid or already exists." This requires the user to figure out which message applies to the error.
- Use message numbers if there is any reasonable chance the user will need to refer to the message when communicating with Technical Support. Do not use them for simple problems like "Invalid Date."

Runtime Behaviors

Overview

This chapter covers the following topics:

- Validation Models, page 7-2
- Saving Changes, page 7-3
- Navigation within Forms, page 7-4
- Navigation to Forms -- Using the "Navigator", page 7-8
- Linked Forms, page 7-10
- Disabled Functions, page 7-11
- Irreversible Actions, page 7-13
- Running Totals, page 7-13
- Field Ranges (From/To Field Pairs), page 7-14
- Currency, page 7-16
- Multiple-Record Selection, page 7-18
- Highlighting Information, page 7-19
- Long-Running Processes, page 7-21
- Ordering of Displayed Records, page 7-21
- About This Record..., page 7-22

- About Oracle Applications..., page 7-22

Validation Models

- (OMS-76001) All Applications screens validate input on a per-field basis. Explicitly coded validation rules must issue a brief but meaningful message when the rules are violated. Oracle Forms itself will issue messages for certain validation failures, such as missing required fields, or incorrect date or number formats.
- (OMS-60055) In certain circumstances it is not possible to validate all input at field level, such as:
 - Performance of heads-down entry screens would be dramatically compromised.
 - Cross-validation rules exist between fields, and cannot be evaluated until the entire record is entered (such as Field Ranges).

In these exceptional cases, validation is performed when the user attempts to leave the record (by navigation, saving, etc.). Should validation fail, specific error messages indicating the invalid fields, and corrective actions to be taken, must be issued. The cursor should be moved to the first of the invalid fields, if possible.

- Some validation is replicated at both field level and commit time, such as uniqueness and referential integrity checks. Oracle Forms itself can trap several of these error conditions automatically.
- (OMS-76002) Fields which cannot currently be validated, because they require some other field(s) to be entered first, prevent input until the other fields(s) are populated and validated.
- (OMS-76003) No data entry can be performed in a detail block until a master record has been entered or queried.
- (OMS-76004) Default values for a field appear immediately upon navigating to a new record. As data is entered in a record, if a "master" field value is changed, "dependent" fields are immediately repopulated, either with "null" or with a default value if one exists.

Related Topics

Errors, page 6-12

Disabled Functions, page 7-11

Saving Changes

The user explicitly saves changes by invoking "File--> Save" from the menu, or by pressing the "Save" button on the toolbar.

Note: When the user selects a form to open from the Navigator, it is invoked in a separate database session and commit cycle. Thus, the Save action only applies to the current form from which it is applied.

Other methods of saving changes are described in the remainder of this section.

Save and Proceed

This function allows a user to save changes on the current set of records, then place the form in a mode ready to start the next transaction. Depending on the form, this may cause any of the following after the data itself is saved:

- The form returns to the exact state it was when first invoked.
- Navigation proceeds to the next master record currently queried.
- A Find window appears if there are no more records left and if the Find window appears when the form is first entered.

A "Save and Proceed" textual button may be coded in modal windows that need the functionality.

Next Step

This function is enabled when a form is opened by using the Processes tab in the Navigator. When selected, it saves any changes in the current form, closes it, and advances the current workflow process to the next sequential node.

Implicit Saves

Several types of actions require that data be saved to the database in order for the actions to be performed, or the action logically saves the changes automatically.

- (OMS-76047) The user should be prompted to save changes if any are pending when an action either requires data to be saved before it can be processed, or does a save itself. If the user decides not to save, then the action is terminated.
- An action may save automatically before or after it is completed if it is reasonable to assume the user expects that behavior. For example, if the user invokes the "Update Order Pricing" action, then it is reasonable to assume that they want to save that transaction upon completion. Furthermore, such an action may mimic the "Save and

Proceed" function if that is the expected, and most useful, behavior of that action.

- An application may save automatically as the user moves between records, but this should be avoided unless it is reasonable (that is, the user would be annoyed by having to perform the Save themselves in complex master-detail forms).

Buttons that perform the "Save" action

(OMS-76048) Often it is unusual to think of "saving" certain types of data, such as the parameters entered to run a report. In those cases, provide a button which replicates the save action, but is labeled consistently with the intent of the form. The "Save" entries on the menu and toolbar would, of course, perform the exact same function as the button.

Examples are:

- "Submit" button on the "Submit Requests" form
- "Import" button on the "Import Requisitions" form
- "Transfer" button on the "Transfer Items" form

Navigation Within Forms

Visual Indicators

The strongest visual indicator of navigation sequence is simply the layout. Follow a left-to-right, top-to-bottom scheme, except where the information is typically presented in a specific format (such as Addresses), and the navigation sequence would still be predictable.

Other indicators regarding navigation are as follows:

- (OMS-76042) Text items which currently are not editable are shown with a gray background.
- (OMS-76044) Buttons, check boxes, and option buttons that are disabled (thus nonnavigable) are dimmed.

Navigation with the Keyboard

- (OMS-76005) Keyboard navigation allows the user to move focus to all editable items, and invoke all functions without mouse intervention.
- (OMS-76006) In some cases there may be noneditable fields that are skipped in the Tab sequence. For example, if there is an item and an item description field, such that the item field is enterable but the description field is not, then the description

field should not be included in the [Tab] sequence. However, users should be able to [Shift][Tab] (backward tab) to the description field from the next item in the navigation sequence if they want to scroll the description or get help. If a field which the user may need to scroll comes after all enterable fields, it must be made navigable.

- (OMS-76049) In query-only blocks, all items are navigable.
- Map item navigation to record and/or block navigation when necessary in order to visit all the blocks drawn in a single window (which the user perceives as a single object.)

For example, a Purchase Order consists of the Header, Lines, and Shipments blocks. The Header and Lines are drawn in one window; the Shipments in a different window. Pressing [Tab] may move the cursor between the Header and Lines blocks, but never moves between the Lines and Shipments blocks, because they are in different windows.

- (OMS-76007) A user can [Tab] past a required field if it is null (it will be trapped when they try to leave the record).
- (OMS-76008) Behavior of [Up arrow] and [Down arrow]
 - When [Down arrow] causes a new record to be instantiated, the cursor moves to the first sequenced item on that record (this is native Oracle Forms behavior).
 - When [Down arrow] or [Up arrow] would cause the cursor to move to an item that is disabled on the target record, the cursor moves to the first sequenced item on that record (this is native Oracle Forms behavior).
- (OMS-76009) Behavior of [Tab] vs. [Return]
 - [Tab] is used to move between fields. [Return] accepts the default button if there is one and does nothing otherwise.

Navigation with the Mouse

- (OMS-73002) Allow the input focus to move to any field at any time, provided that it is meaningful in the current context, and can be validated.

To disable items, Oracle Applications apply a combination of UPDATEABLE, INSERT_ALLOWED, and NAVIGABLE properties, rather than utilizing the DISABLED property because of the side effects this property has, especially in multi-row blocks. This combination allows the cursor to move to a "disabled" item if it is clicked on, but prevents the user from changing the value. For a complete discussion of the reasons for this, see the *Oracle Applications Developer's Guide*.

- (OMS-76010) A disabled field may still accept input focus, but it prevents typing.
- (OMS-76011) A user cannot click out of a field that contains invalid data. However, a user can click out of a required field if it is null (it will be trapped when they try to leave the record).
- (OMS-76012) A user cannot click out of a window that is modal.

Keyboard Navigation to Control Elements

- (OMS-76013) Navigation should proceed to all control elements, except the following:
 - Query Control poplists
 - Buttons below a multi-record block
 - Buttons that apply only when the cursor is on a particular field
 - Tab labels
 - "Clear" buttons

All such widgets must have a keyboard equivalent to activate them, such as Access keys and the tab control list.

Behavior of Next Block

- Next Block is the mechanism for moving "down" the hierarchical blocks of a complex object.
- (OMS-76014) When no child block exists for the current block, Next Block issues the message "At last block."
- (OMS-76015) When only one child block exists for the current block, Next Block moves the input focus to that child, possibly opening another window.
- (OMS-76016) When more than one child block exists, then only one block becomes the "next" block, decided as follows:
 - There is an obvious "dominant" child, or
 - The child blocks have a sequence, and the "first" of these is the "next" block. Such a sequence is indicated by the button layout to access the blocks, with the "first" block button positioned as the rightmost action button.
- Next Block must support movement between sibling blocks when those blocks are

displayed in the same window to facilitate keyboard-only use.

- (OMS-76017) A user can visit a child block when the master record is empty, although no data entry or queries will be allowed.
- (OMS-76018) Next Block from a modal window always issues the message "At last block."
- (OMS-76019) Next Block from the Summary window of a combination block moves to a child block, not the Detail window.
- (OMS-76020) Next Block from a Find window moves to the appropriate results window and performs the indicated query.

Behavior of Previous Block

- Previous Block is the mechanism for moving "up" through the hierarchical blocks of a complex object.
- Always move the input focus to the master block, possibly de-iconifying the parent window.
- (OMS-76021) From the first block of the form, issue the message "At first block."
- (OMS-76022) Previous Block from a modal window always issues the message "At first block."
- (OMS-76023) Previous Block from a Find window moves to the appropriate Results window without performing the indicated query.

Forward Navigation from the last item of a Block

- (OMS-58527) In a multi-record block, pressing [Tab] while on the last item of the record will move to the first item of the next record.
- (OMS-58140) In a single-record block, the behavior is dependent on where child blocks are rendered:
 - If any child block is rendered in the same window, then pressing [Tab] while on the last item of the record will move to the first item of the "current" record in the child block.
 - If no child block is rendered in the same window, then pressing [Tab] will move to the first field of the current record of the same block.

Previous Navigation from the first item of a Block

- (OMS-58141) Behavior is dictated by the Forward navigation requirement. No concession is made to optimize the behavior of "previous" navigation.

Tab or Region Navigation

- (OMS-76026) When the cursor is about to enter a tab region area the target field is the first item of the first tab region, which is not necessarily the region currently displayed. This implies that navigation into that area may alter the region currently displayed. This is consistent with the model for keyboard navigation, where [Tab] visits every item within a block sequentially.
- (OMS-76027) Within a tab region, [Tab] moves the cursor to the next sequenced item, regardless of whether it is within the currently shown region or not.
- (OMS-76028) Within a tab region, [Down arrow] and [Up arrow] navigation preserves the currently displayed region, unless it is inappropriate for the new record, in which case the first tab region is displayed, and the cursor moves to the first item of the record.
- (OMS-76029) Tabbing or selecting a choice from the tab control poplist moves the cursor to the first item of the selected region. However, if the cursor is currently in a different block, the appropriate region is merely shown and the cursor focus does not change.

Related Topics

Tab Regions, page 4-43

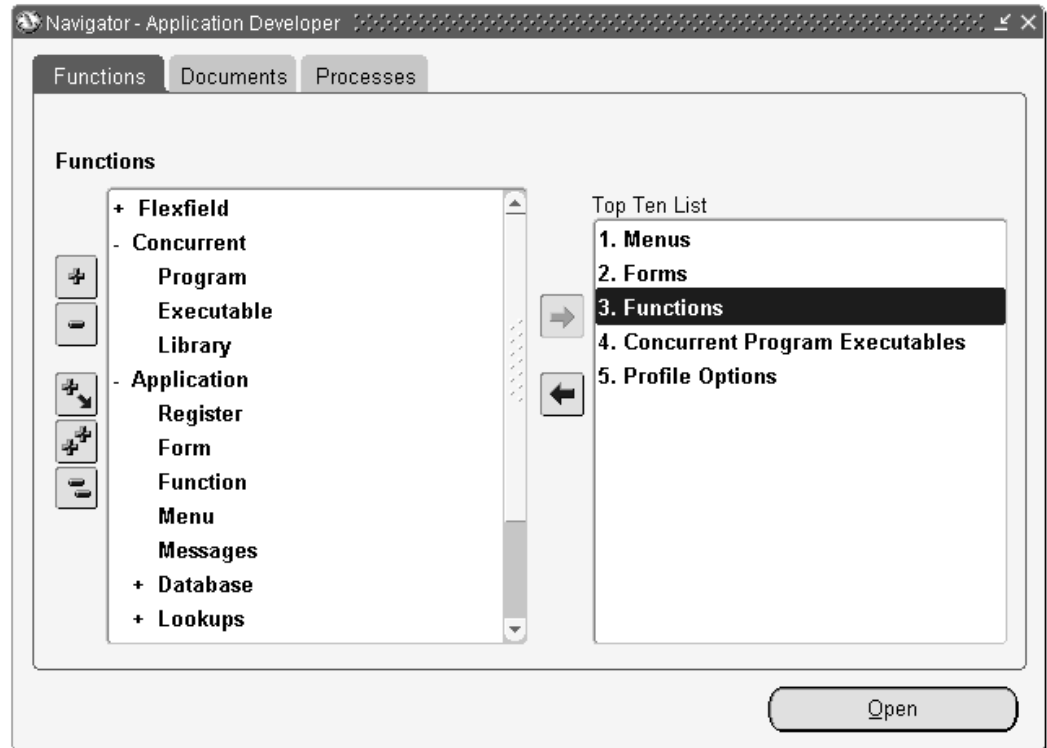
The Navigator

The Navigator (or "Navigate Window") is the means for opening another form while in an application. The Navigate window is always available during a session.

The Navigate Window

If the Navigate window is hidden behind another window or is minimized, choose View--> Show Navigator to bring it to the front.

The following figure is an example of a Navigate window that appears after signing on to Oracle Applications and choosing a responsibility. Users use this window to navigate to forms. The Navigate window is always present during a session of Oracle Applications and displays the name of the current responsibility as part of its window title.



Refer to the *Oracle Applications Developer's Guide* for information regarding functions and responsibilities. The three modes of the Navigator are briefly discussed in this chapter.

Functions

The Functions tab is the primary navigation mechanism. It enables the user to select a particular function and then open that form.

Each user can place up to 10 functions in the Top 10 list for rapid access.

Documents

The Documents tab shows records that were stored with the "Place on Navigator" menu option. This feature is only enabled in specific product forms.

Processes

The Processes tab shows workflow processes to which the current user has access, as well as running instances of them.

A user can launch a process, which then guides them through a specific transaction established by a System Administrator at their company. If a form is opened by using the Processes flow, selecting Next Step from the File menu advances the process to its next sequential step.

Linked Forms

Open Form

The Open Form feature of Oracle Forms allows two or more forms to run simultaneously, but independently of each other. The Navigator is the "launching pad" from which a user opens the screens associated with the current responsibility, and each form opened runs independently.

Specific behaviors of screens in this environment are as follows:

- "File -> Save" only saves the changes in the current form.
- "File -> Close Form" closes only the current form, and asks to save changes if any are pending.
- "File -> Exit Oracle Applications" closes all currently open forms, asking to save changes on a form-by-form basis if any changes are pending.

Developer-Defined Links

A developer may tie two or more forms together with the Open Form feature, when a related inquiry or entry form exists for the current form, and context can be passed. For example, it is very convenient to be able to open the Customers form directly from the Sales Orders form, in order to allow entry of a new customer while creating a new order, or to view details of a customer that has already been defined. These forms should be made available on the Tools or Special menu.

If there is a weak link between two forms, or one that would be used infrequently, do not provide a mechanism on the first form to directly invoke the second; the user can open that form easily from the Navigator instead.

Zoom

The Zoom feature allows customers to create links between Oracle Application forms and forms they code. Oracle Applications ship with no Zooms defined, and the Zoom entry on the View menu is disabled. A customer may add their own code that enables this menu entry on a per-form or per-block basis. When the user selects this entry, the customer's code can execute any logic required, including opening another form and passing context to it. For more information, see the section on Zoom in the *Oracle Applications Developer's Guide*.

Call Form

Call Form is a Form Builder feature that allows one form to invoke another. While the second form is active, the first form is "suspended." Because of the modal nature of this

invocation, and other technical problems, Oracle Application products can not use Call Form.

Disabled Functions

Functions that are never applicable during a session

Functions may not be available to a user for a variety of reasons:

- Function Security for their responsibility excludes it. For example, a particular responsibility may allow viewing of Sales Orders, but not approval of them.
- Based on the setup of a product, the function is not available. For example, "Check Funds" only applies when encumbrance accounting has been turned on.
- Based on the existence of a product. For example, if Oracle Personnel is installed, the employee screens within Oracle Purchasing must not allow certain actions.

These functions can be disabled as follows (in decreasing order of preference):

- (OMS-76030) Widgets should be hidden. This is achieved most easily by positioning such items last in the block or region. Occasionally items surrounding these items can be resized or moved to "fill the gaps." Buttons that might be hidden should be on the left side of the window.
- Such items should be placed in their own region or window if possible, and should never appear as a valid choice in any navigation list.
- If the item cannot be hidden for technical reasons, then it should be disabled at form startup. This is considered an exception, however.

In extreme cases, an entire form may not be accessible. If this cannot be determined before opening the form, then when the form is run it should immediately show an error message, which exits the form when acknowledged.

Items that are disabled and enabled during a session

Certain items need to change their ability to accept user input while the form is running as context changes:

- (OMS-76031) To enforce the field-validation model, dependent fields must be disabled until their master fields are validated. For example, on a Sales Order you can only enter a Customer Site after a Customer Name has been entered and validated. Thus, the Site field is disabled until Name is valid.
- A field may become inapplicable as data is changed. For example, on an Order Line when a service is ordered the Amount field is enabled and the Quantity field is

disabled. When goods are ordered, the Quantity field is enabled and the Amount field is disabled.

Disable as follows:

- In general do not hide items that are not currently applicable, then display them when they become applicable. There are times, however, either due to the complexity created by having all possible buttons and fields shown, or due to lack of space for all possible buttons and fields to be shown at once, where it is acceptable to break this rule. The only place this rule must always be adhered to is in pull-down menus.
- Avoid dynamically disabling buttons except when the state is reasonably fast to determine programmatically, and users will generally understand why it is disabled. Instead, allow a user to press the button, and issue a message stating exactly why the function cannot be invoked.

Examples:

- Only text editors of type "OSM" can have tokens defined.
The button that allows access to the tokens window is disabled for all records that are not of editor type OSM.
- A document can be approved only if it is in a certain state.
The "Approve" button is always enabled, and if pressed, a lengthy process to determine the state of the document is executed. The approval process is either performed, or a message is presented to the user indicating exactly why the process cannot be done.
If the state of a button is dependent on entering a value in a text field, the button should not be dynamically disabled, since this requires the user to figure out that they must tab out of the field to enable the button.

"Inappropriate" Functions

(OMS-76032) Standard Oracle Forms functions that do not apply in a particular situation are disabled. Examples of such functions are:

- Attempting to move to the next record of a single-record dialog block
- Invoking Edit Field on anything but a Text Item
- Attempting to create a new record in a display-only block

Such functions are disabled as follows:

- Oracle Forms automatically prevents their invocation, such as Edit Field on a check box.

- Functions within Oracle Forms that can be declaratively disabled, such as querying or inserting new records, will automatically issue a message similar to "This function is not allowed here."
- Functions such as Next Block or Down that can be invoked with the keyboard and cannot be declaratively disabled should issue a "beep" when pressed.
- (OMS-58168) Menu and toolbar entries are disabled (dimmed) when inappropriate for the current block, record, and field.
- If it is not clear why a particular function cannot be invoked, display a message explicitly stating the reason, rather than issuing a "beep," or relying on the Oracle Forms message.

Do not map a function to a "best guess" alternative. For example, do not map Next Record in a dialog block to be the same as Next Item.

Irreversible Actions

(OMS-76033) Any function that may cause irreversible data changes must prompt the user for confirmation. For example, the action of deleting a record might use the following confirmation: "Delete this Purchase Order? (Delete/Cancel)." More specific delete messages are used where appropriate.

The delete action, invoked by selecting Delete from the Edit pull-down menu or by selecting the Delete button on the toolbar, is the standard way for the user to permanently remove records, whether the actual processing behind the scenes involves 1 or 100 records and whether it will be done online or in a batch job. A button with a more specific label (such as "Purge Journal") may be included if it clarifies the action, but this is always done in addition to enabling the standard invocation mechanisms (the menu and toolbar).

The default button within the confirmation window should be the one that confirms the action. Only in cases where the action (if accidentally taken) has the potential to be exceptionally destructive should the default be to cancel the action. The confirmation may explain that it will be deleted in a job later.

Running Totals

- (OMS-76034) Totals show the cumulative values for all rows, not just the subset of rows currently queried or displayed.
- (OMS-76035) Fields that serve as running totals of detail records are usually drawn in the master block.
- (OMS-76036) In single-record blocks, when a running total field shows a total for multiple fields that are all visible, the fields should be laid out with a one-character

vertical gap below the multiple fields and above the total field, with a zero-height frame (horizontal line) in the middle of the gap.

The decimal places of the Total field must align with those of the fields being totaled.

- (OMS-76037) In multi-record blocks, totals may be indicated in two ways:
 1. If a master block exists, as a field within the master block.
 2. As a separate field below the column that is being totaled.The decimal places of the field must align with those of the column.
- *[Translation]* The field must be long enough to display the expected maximum length of the total in any currency.

Field Ranges (From/To Field Pairs)

A pair of fields used to represent a range has specific cosmetics and behaviors.

Behaviors

- Ranges are validated only at record level. When an error occurs, such as the From value is greater than the To value, an error message is shown indicating the fields in error, such as, "The range of Hire Dates entered is not valid."

In Find windows, this validation occurs when the user presses the Find button or does a Next Block. If the From or To values are invalid, a message is displayed and the cursor is moved to the From field in the Find window.
- The only field-level validation that should occur is the validation of the data entered in the field itself. No cross-field validation with the other field of the pair is done at field level.
- For date ranges, the From field on a new record can default to the current date unless such a default would be incorrect most of the time or might have negative consequences. For example, defaulting the current date into a field that controls the first date of data to be purged would probably not be useful.
- In Find windows, when the focus moves into the To field, if it is blank and there is a value in the From field, copy the value from the From field to the To field. Outside the Find window context, the To value should not be defaulted unless it is most likely that the dates will be the same.
- Ranges are interpreted as listed in the following table:

From Value	To Value	Interpretation
null	null	all values
null	31-DEC-2007	all dates up to and including 31-DEC-2007
01-JAN-2007	null	all values including and after 01-JAN-2007
01-MAY-2007	01-MAY-2007	only values that exactly match 01-MAY-2007
01-MAY-2007	05-MAY-2007	all dates from 01-MAY-2007 to 05-MAY-2007 inclusive

Any time component that may be seen on the actual database column being queried should be ignored if the Find fields merely allow a date entry.

Single-Record Blocks

Range fields may be painted one of two ways: horizontally with a single prompt (such as "Hire Dates") in front of the "From" field and a dash between the fields as in "Hire Dates ____ - ____", or vertically stacked with the prompts From (the upper field) and To with a region frame and title that describes the range (such as "Hire Dates").



A horizontal range field with the prompt "Hire Dates" on the left, followed by a text input field, a dash separator, and another text input field.

In the example above, the prompt for the combined range fields is plural, and the separator between the fields is a dash (boilerplate, not a prompt) centered on three character cells between the two fields of the range. This is the recommended style. Note that the "To" field must have either a hidden prompt or Hint Text because it has no visible prompt of its own.

The style in the following example should only be used if a vertical orientation is aesthetically better or required for translation reasons.



A vertical range field with a region frame titled "Hire Dates". Inside the frame, the word "From" is positioned to the left of a text input field, and the word "To" is positioned to the left of a second text input field below it.

Note that in this example, each field must have Hint Text that combines the frame title and field prompt.

If multiple field ranges exist, and the fields are of different widths, try to arrange them with the From fields and the To fields stacked as shown in the following picture:

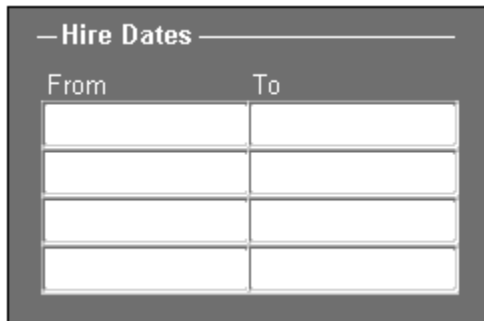


Hire Dates	<input type="text"/>	-	<input type="text"/>
Sequence	<input type="text"/>	-	<input type="text"/>

Note that each field must have Hint Text, such as "Hire Dates: From."

Multi-Record Blocks

In multi-record blocks, range fields are always adjacent fields with the prompts From and To, with a spanning region title for the range contents, as shown in the following example:



— Hire Dates —	
From	To
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

Currency

Currency Code

(OMS-76038) Display the currency code rather than the symbol associated with the currency.

(OMS-76045) Always put the currency code before amounts on forms, letting the value's general label (for example, "Amount") apply to both the currency code and the amount fields. Do this for both input and display-only cases.



Amount	USD	<input type="text" value="123.00"/>
--------	-----	-------------------------------------

A screenshot of a multi-row currency field. The field is titled "Amount" and contains two rows. Each row has a small box with "USD" and a larger text box containing the value "123.00".

Note: The currency field must still have either a hidden prompt or Hint Text.

Alternatively if all the amounts on the window are in the same currency, you can use a single separate field labeled "Currency" to indicate the currency type for amounts on that window.

The currency code can be left off completely (as shown below) if only functional currency is supported for that command.

A screenshot of a single-row currency field. The field is titled "Total" and contains a text box with the value "9972.75".

The prompt for the currency code field can be omitted in a multi-row display-only block.

Field Alignment

(OMS-76039) Currency codes are start-aligned.

(OMS-76040) Currency amounts are end-aligned.

Decimal character (Radix)

Currency amounts must be displayed with the appropriate decimal character. For example, the comma is the appropriate decimal character for German currency (9.123,45) whereas for American currency the appropriate decimal character would be the period (9,123.45).

Field Widths

The numeric field widths that are standard for certain Oracle Applications fields (typically 1.2" or 1.6") are sufficiently large to handle all necessary group separators, decimal characters, and bevels around the field.

Leave 0.4" for the currency field.

Negative Formatting

The format for negative numbers is determined by the profile option "Currency:Negative Format." The default identifier is a hyphen (-) preceding the currency amount, as in "-23." Other display possibilities are 23-, <23>, (23), and [23].

Note: Fields with negative values may also be color-coded if highlighting them is useful.

Validation Messages

If a user changes the currency code when its associated amount(s) are already entered (nonzero), the user should be warned with a message to verify that the amount is correct for the new currency. This is particularly important because in some cases precision may be truncated by the currency change.

If changing the currency code causes an overflow in the amount field(s) (the amount is larger than the acceptable number of digits), blank out the amount(s) and display a message telling the user what has been done. If the amounts cannot be blanked out (such as if the amounts show in saved child records), instead present an error when the currency code field is changed and force its correction or do the change only from a dialog box so the child records can be changed.

Places where extra precision is needed

There are a few places where in the input of prices, quantities, and costs the user may require extra precision beyond the normal precision displayed for a value. In those places allow the decimal point to float as needed and display the amount of precision the user enters. This will upset the numeric alignment in multi-row displays, so it should be done only where such extra precision input may be necessary.

Multiple-Record Selection

Multiple-record blocks may allow a user to take an action on several records at once. For instance, a screen designed to post Journals allows selection of any number of records, and then invoking the "Post" function operates on all the currently selected records.

Behaviors

- A selected record is displayed with a colored background. This includes all text items and poplists of the record.
- All menu items and action buttons which can act on multiple records, display the

number of records selected. For example, if four records are selected to be posted in the Post Journals form the Post button label would be displayed as "Post 4."

Note: Initially and anytime no records are explicitly selected the menu items and buttons display "1" as in, "Post 1" to show they will act on the current record.

- A single record is selected or deselected by holding down the Control key and clicking on the record.
- A continuous range of rows is selected by holding down the Shiftkey and clicking on the record. This selects all rows between and including the current record and the last record selected.
- All records can be selected by choosing "Select All" from the Edit menu.
- All records can be deselected by choosing "Deselect All" from the Edit menu.
- After the desired function is invoked for the selected records, their appearance onscreen should be altered (typically a status field is updated, or the record is cleared entirely if it can no longer be acted on). If the records are not cleared they should remain highlighted.
- A confirmation window should always appear when there is any uncertainty as to what records may be affected or when it is irreversible or destructive, for example, "Delete 3 records?"

Highlighting Information

In certain cases, it may be desirable to make a value or even an entire row stand out to the user. The following are methods for making data stand out from the information around it:

- Required fields on the current record automatically render with a yellow background.
- While in enter-query mode, queryable fields are drawn with a light blue background.
- A particularly important row may be highlighted by changing the background color of all fields.
- A value may be rendered in color to indicate a certain attribute. For example, red is often used for errors and negative values. Be mindful of using color when modifying text as certain colors are commonly used for specific attributes. Distinct

explanations or descriptions may be required.

Important: (OMS-76046) In order to support users who are color blind, any use of color coding must be used to augment only. For example, a financials total may be rendered in red if negative, but should also be drawn such that it is readable without the aid of color (such as with a hyphen preceding the value).

- In the case where a group of two or more enterable fields are related such that filling in one field results in automatic calculation and display of the values in the other fields, it is acceptable to display the calculated value(s) in regular weight (non-bold) font. For example, a discount on a price may be represented by three fields: discount (absolute amount), percent discount, or negotiated price. Any one of the three fields may be entered and appropriate values are then calculated and displayed in the other two fields. It may be necessary (for example, in case the list price changes) to know which value was entered and which values were subsequently calculated. For this reason, the calculated values should be displayed in regular weight font.

Printing

The "Print..." action from the File menu may invoke either of the following responses:

- If there are no reports associated with the task at hand (the task being worked on in the current form or window), then the standard print dialog appears allowing the user to print the current window.
- If there are one or more reports associated with the task at hand and which require minimal (if any) extra parameters to run, then a dialog appears that allows the user to select a particular report, or to invoke the standard print dialog to print the window. If a selected report requires additional parameters that cannot be defaulted, then either additional fields may be presented directly in the dialog, or upon selection of the report, the Submit Requests form may be opened.

Note: This dialog should not list all reports related to the entity being worked on, but rather just the reports directly related to the task at hand.

The option to print just the current screen is always last in the choices, and is the default.

Examples:

- Several reports exist for the current form, and all require multiple parameters that cannot be automatically defaulted:

Selecting "Print..." will bring up a dialog box containing an option group with choices "Reports" and "Print Screen" and buttons "OK" and "Cancel."

Selecting "Reports" will open the SRS form; selecting "Print Screen" will open the platform-specific print dialog.

- One report exists that needs a single additional parameter to be run, and several more complex reports also exist:

Selecting "Print.." will bring up a dialog box containing an option group with choices "Accrued Receipts," "Other Reports" and "Print Screen" and buttons "OK" and "Cancel."

Long-Running Processes

Long-running processes should be handled in the following ways:

- (OMS-76041) Any process which may take more than two seconds to run should change the pointer to a busy indicator (typically the stopwatch or hourglass).
- Any process which may take a substantial amount of time (such as 2 or 3 minutes) should show a progress indicator which shows percent completion, or if the total processing time is not known, it should display the current state of the processing.

Ordering of Displayed Records

This section gives recommendations for allowing a user to specify the order (ascending or descending, for example) of queried records. In general, these ideas are intended to go into a form's Find window, if it has one.

- Use a region titled "Order By" in the lower left of the Find window. This region should have an option group for specifying the order by criteria. This is the preferred method of specifying order by criteria.
- Provide a poplist with "Order By" choices in the Find window. Use this method when the first method is inadequate.
- For the most complicated cases, use an LOV of "order by" choices or an "Order By" region with several poplists (labeled to their left with 1., 2., 3., and so forth, and to their right with ascending/descending icons that can be toggled) for specifying the ordering conditions.

Requerying Records After Changing Order By

In the cases where "Order By" is specified directly on the Results window (not in the Find window), changing the "Order By" should bring up a message "Reorder records

now?" with choices Yes and No (with Yes as the default). If the user chooses Yes, the previous query is run so that the user can see the reordered data.

Record History...

A user chooses "Help -> Record History" in order to see the information automatically supplied by the Applications Object Library "WHO" columns, such as:

- Created By
- Creation Date
- Table Name
- Updated By
- Update Date

These fields are not shown on the form unless they are critical attributes of the entity.

For information on maintaining WHO information, see the *Oracle Applications Developer's Guide*.

About Oracle Applications...

To see basic information about the product being run select "Help -> About Oracle Applications..." from the pull-down menu bar. The About Oracle Applications window provides details about the version of the Oracle Applications components, login information, and information about the current form and environment.

Definitions

Definitions

This appendix contains definitions of certain terms used in this reference manual.

Action Buttons	Product-specific functions or navigation that the user can invoke. "Post," "Encumber," and "Distributions" are examples of actions.
Block	The representation of an entity on a form. For example, the Vendor block consists of the attributes pertaining to the Vendor entity.
Canvas	The surface on which interface items and prompts are physically drawn. Canvases are displayed in a window.
Context	The information necessary to uniquely identify an instance of an entity. For example, when viewing a Sales Order line, a user should also see the Sales Order Number in order to retain context.
Dialog Box	A window used to enter information needed to complete a specific action. The user must interact with this window before proceeding.
Editable Field	A field that allows the user to type information. A field may dynamically change from editable to noneditable as the user operates the form. Nonenterable fields include context, overflow regions, session-disabled, dependent fields, and fields on queried records that cannot be changed.
Folder	A type of block that allows users to customize the screen

and query criteria.

Field	An interface element that displays information to the user and/or accepts input from the user. Text items, check boxes, and poplists are examples of fields. Also known as "Widget" or "Item."
Gateway	A hybrid format consisting of a Summary, Detail, and Find windows. The summary block is almost always a Folder block. From the gateway the user can perform inquiries or limited updates. For example, the Discrete Jobs Gateway in Work In Process (WIP) allows the user to find Discrete Jobs, inquire on details and perform limited updates to them.
Master-Detail	A relation between two entities that indicates a hierarchy of information. For example, a Sales Order consists of a Header entity and a Line entity; the Header is the master of the Line, and the Line is the detail of the Header.
Modal	A state where the user must supply specific information before continuing operation of the application. A dialog box is modal, implying that the user can only operate on that window, and must indicate when they are done (usually by pressing "OK," "Done," or "Cancel" buttons).
Origin	The point of a widget that corresponds to the (X,Y) position that locates it.
Organization	The current value of the organization, set of books, business group, etc., that the user is running.
Overflow Region	A single-record, display-only set of fields that display attributes related to the current record of a multi-record block.
Prompt	A label that uniquely identifies an item. "Salesperson" and "Item Description" are examples of prompts.
Record Scroll Bar	A scroll bar that controls the set of records displayed.
Region	A set of related items within an entity. "Currency Information" is a region of the Purchase Order Header entity, consisting of the Rate, Type, and Date fields.
Region Scroll Bar	A scroll bar drawn within a region that controls the visible portion of a scrolling region.

Save	The process by which data is committed to the database. Explicit saves are initiated by the user (for example, choosing Save from the File menu, or pressing the Save button). Implicit saves are those inherent in the nature of a function (for example, approving a document saves the transaction).
Siblings	A relation between entities indicating that two or more entities have the same master entity. There may or may not be a relation between the sibling entities themselves. A Sales Order may have both Lines and Credits, but there is not necessarily a relation between the Lines and Credits entities themselves.
Toolbar	A series of iconic buttons that replicate common actions on the Menu, such as Save and Print.
Window	A screen in a graphical user interface (GUI) environment. A window is a frame enclosing a surface, on which elements are painted.
Workbench	A workbench is a more powerful form of gateway. Typically users could perform a significant portion of their job functions from the workbench and keep the workbench open all day. For example, the Orders Workbench allows the users to find Orders and Returns and enter, update, approve, cancel, and place them on hold.

Approved Terminology

Overview

This appendix contains abbreviations of certain terms, product names, and acronyms approved for use as boilerplate. In addition, a list of common terms with known translation lengths is provided.

Abbreviations

If possible avoid using abbreviations. If you must abbreviate always leave as much room as possible within the window layout. The following tables include the minimum required length for each of the approved abbreviations.

In addition, always leave extra space when combining words as there may be connector words required in other languages. For example, Item Number cannot be only 5+4 tenths of an inch as in some languages it will translate to Number of Item.

If you must abbreviate a word, always use the abbreviations listed in this appendix. Never abbreviate if sufficient space exists in English to use the full word, except the symbol "%" which should always be used in prompts rather than spelling out the word "percent". Make sure you leave at least enough space for the minimum length listed and the translators will abbreviate in other languages if necessary. If you have multiple uses of a term and some must be abbreviated, you should still fully spell out those with enough room for the full word. Do not abbreviate every usage in pursuit of consistency.

Applications Product Names

The following table lists abbreviations you can use for some of the Oracle Applications product names and the minimum length you must allow for translation.

Product Name	Abbreviation	Min. Length
Accounts Payable	AP	.4"
Accounts Receivable	AR	.4"
Capacity Requirements Planning	CRP	.6"
Engineering	Eng	.6"
Fixed Assets	FA	.5"
General Ledger	GL	.4"
Human Resources	HR	.4"
Material Demand Scheduling	MDS	.6"
Manufacturing	Mfg	.4"
Master Production Scheduling	MPS	.6"
Material Requirements Planning	MRP	.6"
Order Entry	OE	.4"
Project Accounting	PA	.6"
Purchasing	PO	.4"
Rough Cut Capacity Planning	RCCP	.6"
Work In Process	WIP	.4"

Term Abbreviations

The following table lists abbreviations you can use for specific terms and the minimum length you must allow for translation.

Complete Term	Abbreviation	Min. Length
Percent (Always use % in prompts)	%	.1"
Account	Acct	.5"
Cumulative Qty this model year	CUM	.5"
Currency	Curr	.6"
Customer	Cust	.7"
Depreciation	Depr	.7"
Department	Dept	.7"
Description	Desc	.6"
Expenditure	Expnd	.7"
Extension	Ext	.5"
ID / Identification	ID	.4"
Inter-organization	Interorg	.9"
Maximum	Max	.4"
Minimum	Min	.4"
Number	Num	.4"
Operation	Op	.5"
Organization	Org	.5"
Quantity	Qty	.5"
Reclassification	Reclass	.9"

Complete Term	Abbreviation	Min. Length
Reference	Ref	.5"
Release	Rel	.6"
Resource	Res	.7"
Revision	Rev	.4"
Sequence	Seq	.4"
Shipment Documents	Ship Docs	.10"
Shipment Set	Ship Set	.10"
Specification	Spec	.7"
Subinventory	Subinv	.8"
Transaction	Trans	.5"
Unit of Measure	UOM	.4"
Warehouse	Whse	.8"

Terms With No Abbreviation

The terms in the following table do not have an approved abbreviation; however, the maximum length required for translation is less than the rule equivalent (term plus 30%, minimum of 1").

Prompts with Known Translation Lengths	Translation Length
Company	.7"
Item	.5"
Line	.5"

Prompts with Known Translation Lengths	Translation Length
Original	.8"
Probability	.11"
Rank	.5"
Revaluation	.11"
Task	.6"
Time	.5"

Acronyms

The following table lists acronyms you can use for specific terms and the minimum length you must allow for translation.

Complete Term	Approved Acronym	Min. Length
Assemble To Order	ATO	.7"
Available to Promise	ATP	.4"
Construction-In-Process	CIP	.4"
Cost of Goods Sold	COGS	.5"
Engineering Change Order	ECO	.5"
Free On Board	FOB	.9"
General Services Admin	GSA	.4"
Investment Tax Credit	ITC	.4"
Return Material Auth	RMA	.5"
Return to Supplier	RTV	.4"

Complete Term	Approved Acronym	Min. Length
Year-to-Date	YTD	.4"

Index

A

- Abbreviations, B-1
- Accessibility, 2-19, 5-2
- Access keys, 6-10
- Alternative blocks, 4-35
- Americans with Disabilities Act
 - Accessibility, 2-19
- APPCORE, 3-3
- APPDAYPK, 3-3
- APPFLDR, 3-3
- Approved terminology, B-1
- Attributes
 - display-only fields, 2-17
 - LOV fields, 2-17
 - non-applicable items, 2-17
 - visual, 3-1

B

- Blocks, 4-18
 - alternative, 4-35
 - combination, 4-26
 - context, 4-19
 - definition, 2-4
 - dialog, 4-21
 - find, 4-33
 - folder, 4-29
 - multi-record, 4-23
 - navigation, 2-18
 - prompts, 6-5, 6-6
 - relation to windows, 2-11
 - single-record, 4-21

- title, 4-18
- Boilerplate, 6-1
- Buttons, 5-14
 - labeling conventions, 6-8
 - placement, 4-5
 - usage, 2-17

C

- Calculated fields, 7-19
- Calendar, 3-11
- Call Form, 7-10
- Canvases, 4-17
 - content, 4-17
 - definition, 2-4
 - stacked, 4-17
- Check boxes, 5-12
 - definition, 2-5
- Combination blocks, 4-26
 - appearance, 4-26
 - behavior, 4-26
 - buttons, 4-27
 - definition and usage, 2-10
 - gateways, 4-27
 - windows, 4-26
- Committing changes, 7-3
- Container objects, 4-1
- Content canvases, 4-17
- Context, indication of, 2-15
- Context blocks, 4-19
- Copy, 3-5
- Currency, 7-16
- Current record indicator, 2-15, 4-23

Cut, 3-5

D

Definitions, A-1
Descriptive flexfields, 5-19
Design analysis, 2-2
Detail format, 4-26
Dialog blocks, 4-21
Disabled functions, 7-11
Disabled users
 Accessibility, 2-19
Display items, 5-5
Drill-down, 2-18
Dynamic layouts, 2-11
 folders, 2-12
 profiles, 2-12
Dynamic window titles, 4-7

E

Edit menu, 3-5
Editors, 5-19
Ellipses, 6-9
Error messages, 6-12
Expansion requirements, 6-2

F

Field ranges, 7-14
Fields, 2-5
File menu, 3-4
Find blocks, 4-33
Find windows, 4-10
Flexfields
 descriptive, 5-19
 key, 5-21
FNDSQF, 3-3
Folder blocks, 4-29
 cosmetics, 4-32
 developer, 4-33
 functions, 4-29
Folder menu, 3-6
Folders, 2-12
 objects, 3-12
Forms, 4-1
Function invocation, 2-17
Functions

disabled, 7-11

G

Gateways, 4-27
Global settings and objects, 3-1

H

Headers, spanning, 4-42
Help
 About Oracle Applications, 7-22
 About This Record, 7-22
 menu, 3-8
Highlighting, 7-19
Hint messages, 6-14
Hybrid formats, 2-10
 combination blocks, 2-10
 overflow regions, 2-10

I

Images, 2-5
Information
 Oracle Applications, 7-22
 record, 7-22
Information messages, 6-14
Information presentation models, 2-8
 hybrid formats, 2-10
 multi-record formats, 2-9
 regions, 2-8
 single-record formats, 2-9
Inquiry forms, 7-10
Interface, elements of, 2-3
 blocks, 2-4
 buttons, 2-5
 canvases, 2-4
 check boxes, 2-5
 flexfields, descriptive and key, 2-6
 images, 2-5
 list of values (LOV), 2-5
 lists, 2-5
 master-detail relations, 2-4
 menus, 2-3
 messages, 2-6
 radio groups, 2-5
 regions, 2-4
 scroll bars, 2-6

- status bar, 2-6
- text items, 2-5
- toolbar, 2-4
- windows, 2-3

Irreversible actions, 7-13

Items

- display, 5-5
- text, 5-2

K

Key flexfields, 5-21

Keys, 6-10

L

Layout

- general rules, 2-14

Libraries, 3-3

Linked forms, 7-10

List of Values (LOV), 5-17

- definition, 2-5

Lists, 2-5

- poplists, 5-7
- T-lists, 5-9

Long-running processes, 7-21

M

Master-detail relationships, 4-35

- coordination, 4-35
- general behavior, 2-4
- other behaviors, 4-37
- titles, 4-35

Menus, 2-3, 3-3

- customizing, 2-19
- Edit, 3-5
- File, 3-4
- Folder, 3-6
- Help, 3-8
- Oracle Applications main, 3-3
- Popup, 3-9
- Reports and Actions, 3-7
- Tools, 3-7
- View, 3-5
- Window, 3-7

Messages, 2-6, 6-15

Mnemonics, 6-10

Modal windows, 4-14

- closing, 4-15
- Dialog blocks, 4-21
- menus and toolbar, 4-15
- position, 4-14
- size, 4-16
- title, 4-14

Models

- validation, 7-2

Modules, 4-1

- display attributes, 4-1
- functional attributes, 4-3

Movement, 2-17, 7-4

Multi-record blocks, 4-23

- current record indicator, 4-23
- layout, 4-25
- prompts, 6-6
- querying, 4-26
- scrollbar, 4-23
- selecting multiple records, 7-18

Multi-record formats, 2-9

- prompts, 6-6

Mutually calculated fields, 7-19

N

Navigate window, 7-8

Navigation, 2-17, 7-4

- block, 7-6
- control elements, 7-6
- keyboard, 7-4
- mouse, 7-5
- tabbed region, 7-8

Navigator, 7-8

Non-modal windows, 4-6

- closing, 4-9
- menus, 4-10
- position, 4-6
- resizing, 4-9
- title, 4-7

Numeric

- text items, 5-2

O

Open Form, 7-10

Option groups, 5-9

Oracle Applications

- information, 7-22
- Order by, 7-21
- Ordering, 7-21
- Overflow regions, 4-40
 - cosmetics, 4-40
 - navigation, 4-41
 - position, 4-40
 - usage, 2-10

P

- Paste, 3-5
- Percent fields, 6-9
- Poplists, 5-7
- Popup menu, 3-9
- Portability, 2-2
 - application server environment, 2-2
 - screen characteristics, 2-3
 - translation, 2-3
 - web browser environment, 2-2
- Position
 - modal windows, 4-14
 - non-modal windows, 4-6
 - windows, 4-5
- Printing, 7-20
- Processes, long-running, 7-21
- Profiles, 2-12
- Prompts
 - multi-record blocks, 6-6
 - properties, 6-4
 - single-record blocks, 6-5
 - use of data as, 6-11
- Property classes, 3-2

Q

- Query by Example (QBE), 2-16
- Querying, 2-15
- Query Poplists, 2-18
- Question text, 6-13

R

- Radio groups
 - description, 2-5
 - option groups, 5-9
- Ranges, 7-14
- Record indicator, 2-15, 4-23

- Records
 - information, 7-22
 - querying, 2-15
 - selecting, 7-18
- Regions, 4-37
 - definition, 2-4
 - definition and usage, 2-8
 - multi-record blocks, 4-42
 - navigation, 4-38
 - overflow, 4-40
 - scrolling, 4-42
 - single-record formats, 4-39
 - tab, 4-43
 - title, 4-38
- Reports and Actions menus, 3-7
- Requirements analysis
 - portability, 2-2
- Resizing modal windows, 4-16
- Resizing windows, 4-9
- Row LOV
 - windows, 4-13
- Running totals, 7-13
- Runtime behaviors, 7-1

S

- Saving changes, 7-3
- Scroll bars, 2-6
- Scrolling regions, 4-42
- Selecting multiple records, 7-18
- Semi-modal
 - windows, 4-16
- Single-record blocks, 4-21
 - clearing, 4-22
- Single-record formats, 4-39
 - prompts, 6-5
 - usage, 2-9
- Spanning headers, 4-42
- Special menu, 2-19
- Stacked canvases, 4-17
- Status Bar, 3-10
- Summary/detail formats, 4-26

T

- Tab Controls, 2-18
- Tab key`, 2-18
- Tab regions, 4-43

- use of, 2-9
- Terminology
 - approved, B-1
- Text
 - buttons, 6-8
 - conventions, 6-8
 - general properties, 6-2
 - percent fields, 6-9
 - prompts, 6-4
 - titles, 6-4
 - translation requirements, 6-2
- Text items, 5-2
- Titles, 6-4
 - block, 4-18
 - modal windows, 4-14
 - non-modal windows, 4-7
 - region, 4-38
- T-lists, 5-9
- Toolbar, 2-19, 3-9
 - customizing, 2-19
 - definition, 2-4
- Tools menu, 3-7
- Translation, 1-2, 2-3

U

- User interface goals, 1-1
- User interface standards
 - introduction, 1-1

V

- Validation, 7-2
- Validation models, 7-2
- View Find, 2-15
- View menu, 3-5
- Visual attributes, 3-1

W

- Warning messages, 6-13
- Widgets, 5-1
- Windows, 4-4
 - button placement, 4-5
 - closing, 4-9
 - definition, 2-3
 - find, 4-10
 - look and feel, 4-4

- menus, 4-10
- modal, 4-14
- non-modal, 4-6
- position, 4-5
- relation to blocks, 2-11
- resizing, 4-9
- row LOV, 4-13
- semi-modal, 4-16
- size, 4-5
- titles, 4-7
- Window size, limits, 2-3
- Windows menu, 3-7
- Wizards, 2-12
- Workbench, 4-28

Z

- Zoom, 7-10

