Oracle ® Applications
User Interface Standards

RELEASE 11

March 1998
Contents

Preface ................................................. i
  User Interface Goals ........................ ii
  Runtime Environment .................... ii
  Conventions .................................. iii
  Implementing These Standards .......... iii

Chapter 1  General Design and Layout .......... 1 – 1
  Requirements Analysis ..................... 1 – 2
  Task Flow Analysis .......................... 1 – 2
  Designing for Portability ................. 1 – 3
  Elements of the Interface ............... 1 – 5
  Information Presentation Problems .... 1 – 10
    The User’s Perception of Information . 1 – 10
    Summary versus Detail .................. 1 – 10
    Avoiding Information Clutter .......... 1 – 10
    Information Visibility ................. 1 – 11
  Information Presentation Models ...... 1 – 12
    Regions .................................... 1 – 12
    Single–Record Formats .................... 1 – 13
    Multi–Record Formats .................... 1 – 13
    Hybrid Formats ............................ 1 – 14
    Window and Block Relations .......... 1 – 15
    Dynamic Layouts ......................... 1 – 16
    Wizards .................................. 1 – 17
Other Design Considerations .................................... 1 – 18
General Layout Rules .............................................. 1 – 18
Context ............................................................... 1 – 19
Retrieving Records .................................................. 1 – 20
Indicating Attributes ............................................... 1 – 21
Navigation and Function Invocation ............................. 1 – 22

Chapter 2

Global Settings and Objects ......................................... 2 – 1
Visual Attributes ..................................................... 2 – 2
Property Classes ..................................................... 2 – 4
Libraries ............................................................... 2 – 5
Pulldown Menus ....................................................... 2 – 6
  Action ............................................................. 2 – 6
  Edit ................................................................. 2 – 7
  Query ............................................................... 2 – 8
  Go ................................................................. 2 – 8
  Folder .............................................................. 2 – 9
  Special ............................................................ 2 – 10
  Help ............................................................... 2 – 10
Toolbar ................................................................. 2 – 12
Console ............................................................... 2 – 14
Calendar .............................................................. 2 – 15
Folder Objects ......................................................... 2 – 16

Chapter 3

Container Objects ..................................................... 3 – 1
Modules ............................................................... 3 – 2
Windows ............................................................. 3 – 4
  General Look and Feel ........................................ 3 – 4
  Non–Modal Windows ........................................... 3 – 6
  Find Windows ................................................... 3 – 11
  Modal Windows ............................................... 3 – 15
  Semi–Modal Windows ......................................... 3 – 17
Canvasses ............................................................. 3 – 19
  Content Canvasses ............................................ 3 – 19
  Stacked Canvasses ............................................ 3 – 19
Blocks ................................................................. 3 – 22
  General Block Rules ......................................... 3 – 22
  Context Blocks .................................................. 3 – 24
  Dialog Blocks .................................................... 3 – 25
This manual contains the user interface (UI) standards followed by the Oracle Applications development staff. It describes the UI for the Oracle Applications products and how to apply this UI to the design of an application built using Oracle Forms version 4.5.

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 the guidelines.
User Interface Goals

The Oracle Applications user interface is designed to meet the needs of a trained, dedicated user. It seeks to maximize the following aspects of usability:

Productivity

The application must make the end-user more productive than the prior system they were using. It should employ simple, easy-to-grasp presentations of information, familiar to the intended user of each screen.

Ease of Learning

The application must be easy to learn. The products must be consistent, familiar, and predictable. There should be clear and obvious methods for novice users combined with quick and powerful methods for experts.

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 each of the following environments:

- Client environment
  - Monitor: Color, with screen resolution of at least 800x600 pixels
  - Microsoft Windows operating system
  - A Web-Browser, such as Netscape Communicator
- Application Server environment
  - Microsoft Windows (MS Windows)
  - UNIX
Conventions

**Translation Icon**

For ease of use, this book uses the globe icon to indicate which standards are necessary specifically for translation purposes. Designers who are creating applications that do not need to run in other countries can use this icon to identify standards they may choose to ignore for their applications. It may be a good idea, however, to at least read those standards in case the application might need to be 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, but when followed often result in a better user interface.

Throughout this book, each ISO standard is indicated by a number next to the text describing the standard. When the number (preceded by the letters 'OMS') is shown in the margin 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 the Developer/2000 toolset and the client libraries provided by Oracle Application Object Library (AOL) is provided in the *Oracle Applications Developer’s Guide*. Accordingly, there are frequent references to the *Oracle Applications Developer’s Guide* throughout this manual.

Much of the AOL code designed to support these standards assumes that they were followed without exception. In particular, any
standards that describe specific geometric, cosmetic, or behavioral aspects that map to properties available in Oracle Forms, should never be violated if AOL is being used.
User interface development should not begin until a thorough analysis of the typical tasks and business flows that your product must satisfy is complete. This chapter briefly discusses techniques for acquiring that information. The remainder of the chapter discusses translation issues and fundamental data presentation problems that must be addressed and some techniques available to resolve them.

The following topics are covered:

- Requirements analysis
- Elements of the interface, including a brief overview of interface elements such as windows, menus, fields, LOV’s, etc.
- 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, querying records, and more
Requirements Analysis

It is important to design your product user interface based on feedback from qualified users. For example, you should design a journal entry screen based on feedback from the general ledger clerks who will be using it all day long. This seemingly simple concept is often overlooked, and UI designs instead are based on the intuition of a developer and ‘tests’ done on the developer sitting in the next office. Only the actual user of the product is qualified to comment if the UI is appropriate. Be sure to sample a wide variety of users; feedback from users at a single customer site will be biased toward their specific implementation of the task. Whenever possible get feedback based on actual ‘hands-on’ testing, rather than just paper prototypes; there is simply no substitute for observing the user work with the product and studying what challenges they encounter.

Task Flow Analysis

For every task that a product is designed to support, a flowchart of the steps necessary to perform the task should be built. Only by identifying all the aspects of a particular task will opportunities for optimization be revealed. Once the desired flow of a task is documented, you can start analyzing the following issues to lead you to the proper UI 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? Will the screen be used by high-speed data entry clerks, in which case using widgets that require a mouse would be inappropriate?
- What level of training do you anticipate the user having on this particular task and screen? Does the screen need to be optimized for first-time or infrequent use, 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?
Designing for Portability

Restrictions imposed by supporting different environment and translation requirements must be evaluated early in the design phase. Any decision to deviate from these restrictions must be considered carefully.

Web browser environment

Oracle Forms Web Cartridge currently does not support all window and mouse triggers such as "MOUSE_ENTER" and "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 the user will not see if they are running in a web browser.

Screen Characteristics

The maximum allowed window size is 7.8” (width) x 5.0” (height) (OMS–73015). This maximum size comes from the requirement to operate with 800x600 screen resolutions.

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.

Finally, to support users who are colorblind, 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).
Translation

So that a product runs in the proper language in all countries, all prompts, titles, messages, and data presented to the user, other than data they enter while using the product, must be translatable. All text must be presented to a user in their native language, except critical errors which a developer needs to debug the application (and should never be seen by a user).

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. For more information on expansion requirements, refer to Chapter 5.

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 separator character.

**Note:** OracleApplications currently requires that dates be entered in the format DD–MON–RR (for example, 01–JAN–98). This restriction will be lifted in a future release of the product.
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. Some of these elements are shown in Figure 1–1.

Windows

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 only act within that window.

Windows (See page 3–4)

Menus

A menu is a list of actions from which the user can choose. A menu is associated with each form, and displays either within each window, or within the container window. There are specific rules for which windows inherit the menu, and the behaviors for dynamically enabling
and disabling menu entries. Oracle Applications use a generic menu for all products to invoke standard functions. This menu also includes a ‘Special’ menu for product–specific functions that can be enabled and populated at runtime.

- Pulldown Menus (See page 2 – 6)

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 either its own window or within the container window.

- Toolbar (See page 2 – 12)

Canvasses

Canvasses are surfaces within windows on which objects are displayed. A window may show one or more canvasses, either one at a time or simultaneously. The choice of canvas type (stacked or content), and placement, is based on the information presentation model chosen.

- Canvasses (See page 3 – 19)

Blocks

A block is the physical implementation of an ’entity’. A block ’owns’ interface items, but is an interface item itself in that all items of a record in a block must be complete before the user can move out of that record. The items of a block may be placed on one or more canvasses, but usually a block is presented in a single window. Blocks may show one or more records of an entity at a time.

- Blocks (See page 3 – 22)

Regions

Regions are logical groupings of fields, usually within a block. There are specific cosmetics to indicate a region, and behaviors for ’Alternative Regions’, which allow more than one region to occupy the same window space.

- Regions (See page 3 – 44)
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 while 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.

Fields

Fields, also known as text items, allow a user to type alphanumeric characters. Often a field has a List of Values (LOV) associated with it to restrict the set of valid values.

List 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 popup window at the user’s request.

Buttons

Buttons are used to invoke actions. 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 whole window. There are specific guidelines for placement, size, and labelling.

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 clearly be thought of as having ‘true’ (checked) and ‘false’ (unchecked) states.
Display Items
A display item is used to show textual information, which the user cannot interact with in any way. It is appropriate for display-only information which requires no scrolling or querying. Additionally, it can be rendered to match the standards for prompts and titles, and thus can be used for dynamic labelling.

Lists
List items allow the user to select a value from a drop-down list, with a recommended maximum of 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 TLists commonly called a "scroll list".

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

Charts
Charts displayed by Oracle Graphics may be embedded within an Oracle Forms screen, either as a bitmap or 'live' object. A discussion of standards for chart items is planned for a later release of this document.

Option Groups
Like List items, option groups allow the selection of one value from a choice of several. Option groups must always have a value, and consume extensive real estate, especially in multi-record blocks.

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 left of the block (OMS–73078).

Multi-Record Blocks (See page 3 – 28)

Stacked Canvas scroll bars allow a portion of the screen to scroll within a specific viewport; all stacked canvasses that require scrolling use scroll bars (OMS–73267).

Stacked Canvasses (See page 3 – 19)
Regions that Scroll (See page 3 – 50)

Multiline text item scroll bars allow scrolling within a multiline text item; all multiline text items use a scrollbar. (OMS–74007).

Messages

Messages are shown in popup windows, to which the user must respond. Different styles of popups and required responses are discussed, as well as standards for verbiage they contain.

Errors (See page 5 – 19)
Warnings (See page 5 – 20)
Questions (See page 5 – 21)
Information (See page 5 – 22)

Editors

Oracle Forms supports the opening of a simple editor from any text item. The user may specify their own default editor for fields that need full word processing capabilities.

Editors (See page 4 – 26)

Descriptive and Key Flexfields

Oracle Applications uses 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 is customized at the customer site so that users may enter additional information which the product has not already provided a field.

Descriptive Flexfields (See page 4 – 27)
Key Flexfields (See page 4 – 30)
Information Presentation Problems

The fundamental problems to solve when designing a user interface are how to best present information to a user, and how to allow the user to access the information they need. This section discusses some high-level issues that must be answered while optimizing windows for a task, followed by models that attempt to resolve them. 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 desire to see many records of an entity simultaneously, and to see all of the detail about each record at once. 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.

Avoiding Information Clutter

If it is well organized, even large and complex sets of information can be presented in a way which allows a user easy access to the information they need. However, if information is not well organized, then it simply becomes clutter. Presenting an entity with a large number of attributes merely as a ‘sea’ of fields would force the user to
scan an entire screen to locate the piece of information they are interested in. Instead, carefully organize and group the data to allow a user to navigate the information more easily and naturally.

**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.
Information Presentation Models

This section contains various methods available for laying out windows. The models presented here are not meant to be an exclusive list though; if another layout is more appropriate for a task, then it should be used.

Regions

A region is a 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 groups of fields. Regions are usually displayed with rectangles or lines surrounding them and a title to make the grouping 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.

Regions (See page 3 – 44)

After an entity is divided into regions, if all regions cannot be displayed simultaneously we use a scheme called ‘Alternative Regions’ whereby two or more regions are displayed in the exact same space, one at a time. The user can jump to a specific region by selecting a value from a poplist, or [Tab] continuously through all fields of all regions. The poplist doubles as the region title. A set of fields, minimally the primary key fields, should be excluded from the alternative regions and remain frozen on the content canvas so the user can maintain context at all times.

When best to use Alternative Regions:

• 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 based on other product installations.

Alternative Regions (See page 3 – 51)

Scrolling of regions is not desirable since it may leave the user laboriously switching to, then scrolling through each region searching for a field, but is acceptable when absolutely necessary. A region should only scroll when the logical grouping of fields results in a field
Regions that Scroll (See page 3 – 50)

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

Regions in Single–Record Formats (See page 3 – 46)

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:

- The user must see multiple records to perform the transaction, because there is a relationship between the records
- The user must see summary attributes of many rows at the same time, typically to scan for information quickly
- Displaying the data in a record will easily fit on one line
- Displaying data in a multi–record format conveys that more than one record can be entered
- The user normally perceives of the entity in a multi–record format (e.g. the Lines of a Requisition)

Regions in Multi–Record Blocks (See page 3 – 49)
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 (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.
- For records which include multi-line text items, the multi-line items may be displayed in the overflow region.

Overflow Regions (See page 3 – 47)

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 Chapter 3.

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

When best to use:

- 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.

**Combination Blocks (See page 3 – 31)**

---

**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 where 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 horizontal line across the entire width of the content canvas may be used to identify each block (OMS-73070). The block title is displayed on this line but is only necessary if the content of the blocks is not obvious.

When the blocks cannot all be displayed at once, then ‘Alternative Blocks’ are employed. An extension of Alternative Regions, selecting a value from the poplist causes the appropriate block to display. Each of these blocks may, in turn, have alternative regions. When this occurs, the list contains all the regions for all the blocks, with regions pertaining to the same block prefixed by the block title (OMS-73203).
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

Based on the value of a particular profile, product installation, setup parameter, etc., the screen layout can be dynamically modified. 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 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 a customer to change which attributes are on a particular screen, as well as their order and size along with 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 that customization would substantially improve the usability of the screen.

Folder Blocks (See page 3 – 35)

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 or provided for the user.

Wizards should be available for all complex tasks, 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.
- **Previous**: Returns the user to the previous window in the wizard. This button appears on all but the first window.
- **Done**: 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 window within the wizard, if all of the steps are not required. Pressing Done should save the changes the user has entered, unless this would be considered an unexpected or undesirable behavior.
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.

- Don’t 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 inches high, and multiples of 0.1 inches wide. Two–dimensional widgets (multiline text items, TLists) are multiples of 0.25 inches high. Textual button heights differ depending on the platform but a minimum height of 0.3” must be reserved for them (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 multiline prompts.

- At least a one–character wide space should be left between the items on a canvas and the window frame (OMS–75008).

  - It is also preferable to leave at least a half row (.125”) at the top and bottom empty too, but this is not required. Exceptional cases exist for lines and rectangles used to denote blocks and regions (OMS–73007).
• Indicate things that are similar, and things that are different, when it is meaningful for the user. That is, do not create regions or use boilerplate lines to group items unless it improves the usability of the screen.

• Make data stand out, and 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 Chapter 4.

• When displaying Currency (i.e. 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).

Titles in Non-Modal Windows (See page 3 – 7)

• 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).

Context Blocks (See page 3 – 24)

• All multi-record blocks include a ‘Current Record Indicator’ field, to clearly point out the current record (OMS-73178).

Current Record Indicator (See page 3 – 28)
Retrieving Records

Oracle Applications provide three methods to locate previously entered data. When using 'Query Enter', 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 'Query 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 'Query Find' with either a Find window or a Row LOV as described below.

**Query Find**

When the user invokes Query 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 auto-query. Query 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 auto-query
   
     or
   
   - For allowing the user to find a subset of records in a detail block

   or

   - 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 a List of Values after the Find Window if necessary) before being displayed.

   Find windows (See page 3 – 11)

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

   When a “Row LOV” should be used (OMS–73557):
   - For a single-row block display
or

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

List of Values (LOV) (See page 4 – 23)

**Query Enter**

Query mode is the native Oracle Forms mechanism for querying data, and allows complex queries with Query-By-Example (QBE). All blocks that allow the user to retrieve data support Query Enter. Query Enter 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 query mode to run a command.

**Note:** LOVs are not always used in Query Mode. However, the LOV icon and the <List> lamp will still be displayed as if they were active when navigating through the fields. Trying to invoke an LOV in query mode will produce an error message unless, the LOV was specially coded and enabled for this mode.

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

**Direct Entry**

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 ‘Query Find’ or ‘Query Enter’. Instead, direct entry into the primary key fields of the master can be used to uniquely specify the master record.

Find Blocks (See page 3 – 39)

**Indicating Attributes**

The following characteristics of fields, records, and blocks are indicated:
• Text items that are always display-only (in which the user can never type) are displayed without a bevel and with a slightly smaller height than other fields. Examples are context fields, most overflow region fields, and total fields (OMS–74010).

• Text items that may ever allow the user to type in values are displayed with a bevel (OMS–74009).

• Text or List items that are currently not applicable, such as a ‘dependent’ field that can only be validated when a ‘master’ field has a value, are displayed with the same color as the canvas (grey); all other fields always have a white background (OMS–74014, OMS–74130).

OMS–71019: Fields that respond to the List of Values function show the List lamp in the console, and have the List of Values entries enabled in the menu and toolbar.

OMS–71020: Text items always show the Edit lamp in the console, and have the Edit Field entries enabled in the menu and toolbar.

OMS–71021: Entries on the menu and toolbar are disabled if the function is not available for the current field, record, or block.

Navigation and Function Invocation

Several elements are used to facilitate movement among the screens and blocks of a product, as well 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.

Navigating to Forms — Using the “Navigator” (See page 6 – 10)

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 Special 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.

Buttons (See page 4 – 21)

Lists

Lists are used as control elements of alternative regions and blocks (in addition to being used as data elements). They allow the user to select a particular region, and double as the title of the region currently shown.

The choices associated with all alternative regions and blocks are also accessible with the ‘Block Menu’ function of Oracle Forms. Pressing this function will open a List of Values. This function is provided to enable keyboard–only usage of the products (OMS–73132).

Alternative Regions (See page 3 – 51)

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.

Behavior of Next/Previous Block (See page 6 – 7)

Drilldown

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 drilldown, the user can click on the current record indicator to perform this function (OMS–73080). In addition, fields that allow drilldown are
indicated by green underlined text and can be accessed by double-clicking on the field.

Current Record Indicator (See page 3 – 28)

[Tab]

[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 either:

- 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.

Navigation (See page 6 – 5)

Special Menu and Toolbar

Product-specific functions that apply to all or most windows of a form may be placed in the Special menu. These functions may also be replicated as textual buttons within windows to increase their visibility and accessibility. Furthermore, up to three functions may be added to the iconic toolbar, provided the functions apply to most of the windows of the form (OMS-72006).

Pulldown Menus (See page 2 – 6)
Toolbar (See page 2 – 12)
This section describes standard settings and objects that exist in every form, and for the most part are invariant between screens. 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).

The following topics are covered in this chapter:

- Visual Attributes
- Property Classes
- Libraries
- Pulldown Menus
- Toolbar
- Console
- Calendar
- Folder Objects
Visual Attributes

Applications Object Library includes pre-defined visual attributes, linked to a custom color palette.

Every form uses the same color palette. The specific values these standards require are listed below; however, the color palette contains a full spectrum of colors for use with images or product-specific needs.

<table>
<thead>
<tr>
<th>Color Name</th>
<th>Red(%)</th>
<th>Green(%)</th>
<th>Blue(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Black</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>White</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Canvas</td>
<td>75</td>
<td>75</td>
<td>75</td>
</tr>
<tr>
<td>Cyan</td>
<td>0</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Blue</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>Button</td>
<td>40</td>
<td>55</td>
<td>70</td>
</tr>
</tbody>
</table>

Table 2 – 1 (Page 1 of 1)

Visual Attributes are combinations of typefaces and colors that can be applied to many Oracle Forms objects. The visual attributes enforce the following colors and fonts:

<table>
<thead>
<tr>
<th>Object</th>
<th>MS Windows</th>
<th>Other Platforms</th>
</tr>
</thead>
</table>
| Prompts     | Black on Canvas  
MS Sans Serif 10 Med | Black on Canvas  
Helvetica 10 Med |
| Titles      | Black on Canvas  
MS Sans Serif 10 Bold | Black on Canvas  
Helvetica 10 Bold |
| Canvasses   | Black on Canvas  
N/A | Black on Canvas  
N/A |
| Check boxes | Black on Canvas  
MS Sans Serif 10 Med | Black on Canvas  
Helvetica 10 Med |
| Option Groups | Black on Canvas  
MS Sans Serif 10 Med | Black on Canvas  
Helvetica 10 Med |
| Textual Buttons | (Controlled by MS Windows)  
MS Sans Serif 10 Bold | White on Button  
Helvetica 10 Bold |
| Iconic Buttons | (Controlled by MS Windows)  
N/A | White on Button  
N/A |

Table 2 – 2 (Page 1 of 2)
Note: The look of a widget itself, such as a check box, is controlled by the particular GUI and/or its control settings. The colors and fonts specified here may only affect the textual component of the widget.

<table>
<thead>
<tr>
<th>Object</th>
<th>MS Windows</th>
<th>Other Platforms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Lists</td>
<td>Black on White MS Sans Serif 10 Bold</td>
<td>Black on White Helvetica 10 Bold</td>
</tr>
<tr>
<td>Control Lists</td>
<td>White on Blue MS Sans Serif 10 Bold</td>
<td>White on Button Helvetica 10 Bold</td>
</tr>
<tr>
<td>Fields</td>
<td>Black on White MS Sans Serif 10 Bold</td>
<td>Black on White Helvetica 10 Bold</td>
</tr>
<tr>
<td>Disabled Fields</td>
<td>Black on Canvas MS Sans Serif 10 Med</td>
<td>Black on Canvas Helvetica 10 Med</td>
</tr>
<tr>
<td>Selected Records</td>
<td>Black on Cyan MS Sans Serif 10 Bold</td>
<td>Black on Cyan Helvetica 10 Bold</td>
</tr>
<tr>
<td>Record Indicator</td>
<td>Blue on Blue N/A</td>
<td>Button on Button N/A</td>
</tr>
</tbody>
</table>

Table 2 – 2  (Page 2 of 2)
Property Classes

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

- All enterable fields are 0.25” high, and have a lowered bevel (OMS–74009).
- All display-only fields are 0.21” high, and have no bevel (OMS–74010).

For more information on Property Classes, see the Oracle Applications Developer’s Guide.
Libraries

Libraries contain reusable client-side code. 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 Applications Object Library are:

<table>
<thead>
<tr>
<th>Library Name</th>
<th>When Used?</th>
<th>Library Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>FNDSQP</td>
<td>Always</td>
<td>Routines for message dictionary, profiles, flexfields, and various other utilities</td>
</tr>
<tr>
<td>APPCORE</td>
<td>Always</td>
<td>Packages specifically designed to implement these standards, and various other utilities</td>
</tr>
<tr>
<td>APPDAYPK</td>
<td>For any form that contains dates</td>
<td>Packages to implement the calendar, used on most date fields</td>
</tr>
<tr>
<td>CUSTOM</td>
<td>For writing Zooms or other custom code</td>
<td>'NULL' routines until customized</td>
</tr>
</tbody>
</table>

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 discuss the use of these libraries in detail.

Oracle Applications Developer's Guide
Pulldown Menus

The menu allows the user to invoke ‘standard’ Oracle Forms functions, such as ‘Clear Record’ and ‘List of Values’. It also contains features unique to Oracle Applications, such as Attachments. The menu may also contain product–specific entries under ‘Special’. Most entries on the menu are disabled and enabled automatically based on the current context (OMS–71021).

OMS–72003: The menu is a pulldown style and is visible at all times. Where possible, standard GUI accelerator keys are available for menu entries (such as Cut, Copy, and Paste). Most entries also have equivalent Oracle Forms keys, determined by key mappings in Oracle Terminal.

OMS–72004: The top level of the menu is as follows:

The pulldown choices for each menu item are listed below. There is a brief explanation given with each menu choice.

### Action

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Navigate To...</strong></td>
<td>Shows the Navigator window.</td>
</tr>
<tr>
<td><strong>Zoom</strong></td>
<td>Invokes customer–defined Zooms.</td>
</tr>
<tr>
<td><strong>Save</strong></td>
<td>Saves any pending changes.</td>
</tr>
<tr>
<td><strong>Save and Proceed</strong></td>
<td>Saves any pending changes, and returns the form to a state where the next transaction can be started.</td>
</tr>
<tr>
<td><strong>Place on Navigator</strong></td>
<td>Saves the current record onto the Navigator for quick access later.</td>
</tr>
<tr>
<td><strong>Print...</strong></td>
<td>Prints the current screen that the cursor is in. In some cases it may print a report associated with the current data.</td>
</tr>
<tr>
<td><strong>Refresh</strong></td>
<td>Repaints the entire screen.</td>
</tr>
<tr>
<td><strong>Close Window</strong></td>
<td>Closes the current window.</td>
</tr>
<tr>
<td><strong>Close Form</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Exit Oracle Applications</strong></td>
<td></td>
</tr>
</tbody>
</table>
Close Form  Closes all windows of the current form.
Exit Oracle Applications  Exits the application.

**Edit**

The Edit pulldown, shown here, contains the following choices:

- **Cut**  Cuts current selection to the clipboard.
- **Copy**  Copies current selection to the clipboard.
- **Paste**  Pastes from the clipboard.
- **Clear Field**  Clears the current field.
- **Duplicate Field Above**  Copies the value from the prior row.
- **List of Values...**  Invokes the LOV window.
- **Edit Field...**  Invokes the Editor. The editor to be used can be specified in an Oracle Forms configuration file. By default the Oracle Forms editor is invoked.
- **New Record**  Creates a new blank record.
- **Delete Record**  Deletes the current database record.
- **Clear Record**  Clears the current record, and does not ask for confirmation if pending changes might be lost.
- **Duplicate Record Above**  Copies the prior record to the current record.
- **Translations...**  Invokes the Translations window.
- **Attachments...**  Invokes the Attachments window.
- **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).
- **Clear Block**  Clears all records of the current block.
- **Clear Form**  Clears all pending changes in the current form.
Query

The Query pulldown, shown here, contains the following choices:

### Find...
- Shows the Find window or Row LOV in order to retrieve records.

### Find All
- Retrieves all records.

### Enter
- Invokes Query–By–Example mode.

### Run
- Retrieves all records, or if in Query–By–Example mode runs the currently entered query.

### Cancel
- Exits Query–By–Example mode.

### Show Last Criteria
- Recovers the previous criteria while in Query–By–Example mode.

### Count Matching Records
- Counts the number of records that will be retrieved using Query–By–Example mode.

### Get More Records
- Fetches the next set of records that match the current query.

Go

The Go pulldown, shown here, contains the following choices:

### Next Field
- Moves the cursor to the next field.

### Previous Field
- Moves the cursor to the previous field.

### Next Record
- Moves the cursor to the next record.

### Previous Record
- Moves the cursor to the previous record.

### First Record
- Moves the cursor to the first record.

### Last Record
- Moves the cursor to the last record.

### Next Block
- Moves the cursor to the next block.

### Previous Block
- Moves the cursor to the previous block.
Summary/Detail  Switches between summary and detail views of a combination block.

Folder

The Folder pulldown, shown here, contains the following choices:

New...  Creates a new folder.
Open... Allows selection of a previously saved folder.
Save   Saves changes of the current folder definition.
Save As... Same as Save, but allows entry of a name to copy folders.
Delete... Allows deletion of a saved folder.
Show Field... Allows selection of a field to display 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... Allows the current prompt to be changed.
Autosize All Adjusts all column widths based on the data they contain.
Show Order By Allows specification of the sort order.
View Query... Displays the current query restrictions.
Reset Query Erases the current query restrictions.
Folder Tools Displays the folder tool palette, which contains the most commonly used folder functions.
Note: The Folder pulldown menu is only enabled when the cursor is in a folder block. Specific entries within the menu may be disabled based on your site profile option values or rules established by a developer.

Special

The Special pulldown menu may contain up to fifteen product-specific entries. It is not enabled except for those products that choose to make use of it. For more information on using the Special menu to link forms, refer to Chapter 6.

Help

The Help pulldown, shown here, contains the following choices:

- **Window Help**
  - Displays the help text 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.

- **Display Database Error...**
  - Displays the last database error.

- **Tools**
  - Displays the Tools submenu shown below.

- **View My Requests**
  - Shows the current users completed Concurrent Requests.

- **About This Record...**
  - Displays information about the current record.
Displays information about the current form and environment.

The Tools submenu provides access to the diagnostic tools. A user would normally only invoke these functions at the request of their System Administrator or other Support personal.

**Examine**
Displays the underlying Oracle Forms block and item (displayed as *Field*) corresponding to a field on the form.

**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 V4.5 Debugger, provided the form you are currently in was started in debug mode (e.g., starting the form from the command line on a UNIX platform, you would specify Debug=YES).

**Properties**
Brings up the Properties submenu shown below.

**Custom Code**
Brings up the Custom Code submenu shown below.

The Properties submenu provides diagnostic information for Items and Folders.

**Items**
Displays the properties of the current item.

**Folder**
Writes diagnostic information about the current folder to the working directory.

The Custom Code submenu is an option group, allowing you to change the mode of the custom library used by Zoom.

**Normal**
The custom library is used. This is the default.

**Off**
Turns the custom library off, disabling Zooms and other customizations made in the CUSTOM library.
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 automatically attaches to either the container window or the root window. For a picture that includes the toolbar, refer to Figure 1–1.

The toolbar consists of the following, arranged horizontally:

- **Navigate To...**—Shows the Navigator window.
- **Zoom**—Invokes customer–defined Zoom.
- **Save**—Saves any pending changes.
- **Save and Proceed**—Saves any pending changes, returns the form to a state where the next transaction can be started.
- **Print**—Prints the current screen that the cursor is in. In some cases it may print a report associated with the current data.
- **Clear Form**—Clears all pending changes in the current form.
- **Summary/Detail**—Switches between summary and detail views of a combination block.
- **Find**—Shows the Find window or Row LOV in order to retrieve records.
- **New Record**—Creates a new blank record.
- **Delete Record**—Deletes the current database record.
- **Clear Record**—Clears the current record, and does not ask for conformation if pending changes might be lost.
- **Folder Tools**—Displays the folder tool palette.
- **Translations**—Invokes the Translations window.
Attachments—Invokes the Attachments window. If one or more attachments already exist the icon changes to a paper clip on a piece of paper.

List of Values—Invokes the LOV window for the current field.

Edit Field...—Invokes the Editor.

Window Help—display help for the current window.

Additionally, up to three buttons, to the right of the Help button, are available for product-specific functions. These functions would also be replicated on the 'Special' menu. For example, when the Navigate window is active, the following additional button appears to the right of the Help button:

Switch Responsibility—Displays the Responsibilities window.

For more information on the Toolbar, see the Oracle Applications Developer’s Guide.
Console

The console contains the Oracle Forms message and status line. The console is attached to the same window as the toolbar.
Calendar

The Calendar (shown in Figure 2 – 2) 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, as well as the Folder tools palette. These objects appear to the user automatically during normal operation of a folder block.

Folder Blocks (See page 3 – 35)
This section describes the standard properties and behaviors for Modules, Windows, Canvasses, 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
- Windows, including modal and non-modal windows
- Canvasses, including content and stacked canvasses
- Blocks
- Regions
**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. ‘Real’ coordinates are used, because they are based on a logical, not physical, measure (OMS–71002). The following picture shows the relevant designer settings:

![Diagram of a designer interface showing module properties](image)

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 and MS Sans Serif 10 point)
• The height is derived from the number of pixels necessary to fully display Kanji characters (14 pixels excluding the text item bevel in SVGA).

OMS–73900: The following picture shows the correct Designer ruler settings, which must be set for every canvas being developed.

![Designer ruler settings](image)

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, GUI 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.

[Validation Models (See page 0–2)]
Windows

Windows inherit the proper look and feel of the GUI on which they are running, such as characteristics of the frame, title bar fonts, 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:

- inherit their color settings from the environment.
- do not use bevels around the edge of the window.
- leave a gap at the top and bottom of the window, except for buttons and coordination check boxes and several other exceptional cases.
- leave the left and right edge character cell columns of a window blank, except for region lines and block boundary lines.

Icons

Icons that are shown when a window is minimized are determined in the following manner:

- The Toolbar, Navigator, Attachments, 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

OMS–73012: Each window in a form must be titled uniquely so that iconified names and entries in the Windows menu are significant.

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

Scroll Bars

OMS–73013: No scroll bars are attached to the window (although scroll bars may appear inside the window, attached to canvasses, blocks, or items).

Toolbars

OMS–73014: No windows use toolbars, except the ROOT_WINDOW or container window which controls the Applications toolbar.

Window Style

On platforms with a container window, most application windows appear inside the container window except certain dialogs and the Folder tool palette.

Size

Guidelines and limits for window size are as follows:

OMS–73015: • The maximum window size is 7.8” (width) x 5.0” (height), to support an 800x600 screen resolution.

OMS–73016: • The minimum size of a window is 2” x 2”.

OMS–73017: • A window may be drawn any size between the maximum and minimum, with the following recommendations:
  
  Make windows only as large as is necessary.

  Make non–modal windows within a form approximately the same size.

  Make modal windows smaller than non–modal windows that invoke them.

Position

The following points pertain to window position:

OMS–73018: • A window must be fully visible when it is first opened.
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.

• A child or detail window opens in relation to the parent window that invoked it, in one of the following styles:
  - 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 its left edge, but offset down by 0.3”.
  - CENTER: Window opens centered relative to another window.

The following picture illustrates Summary/Detail positioning, as well as Cascading and Tiling of windows.
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).

**OMS– 73023:**
- 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, or until it is closed.

**OMS– 73024:**
- 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.

### Title

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

**OMS– 58501:**
- Most windows are titled with the name of the object shown in the window.

  EXAMPLES: Sales Orders, Journal Entries, QuickCodes, Periods, Menus, Reports

**OMS– 58500:**
- 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'.

  EXAMPLES: Transfer Items, Post Journals, AutoCreate Quotes, Import Receipts
If the window is display only, use the format ‘View <Noun>‘.

EXAMPLES: View Requests, View Receipts

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.

OMS– 58503:
- 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– 73027:
- 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>)’

    EXAMPLES: Transfer Items (WR1)
              Post Journals (AR1)
              AutoCreate Quotes (AR5)
              Import Receipts (HQ)

OMS– 73028:
- If the product uses Datetrack, and the Datetrack date is different from the current date, then the title is of the format ‘<window title> (<Org> <Date>)’

    EXAMPLE: Hire Person (SF1 12–JUN–1992)

OMS– 58505:
- Titles of all but the First Window of a form should also display context of the ‘topmost’ master record.
  - Context is shown in the format ‘<window title> – <context>‘. Multiple context values may be displayed as in Example 1 where the window title ‘Journal Lines’ is followed by two context values ‘NOV93’ and ‘123’. 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]’.

    EXAMPLE 1: Journal Lines – NOV93, 123
    EXAMPLE 2: Assignments (OR1 12–JUN–1992) – John Doe
    EXAMPLE 3: Benefits – John Doe
    EXAMPLE 4: Purchase Order Lines (ABC) – [New]

The following picture shows an example of a window title hierarchy.
OMS–73030:
- 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–73930:
- 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–73031:
- Whenever a master record is changed, or primary key information on the record is changed, context shown in window titles must be immediately updated.

**Note:** 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:

- **OMS– 58147:** A user closes a non-modal window with the native GUI close mechanism or ‘Action, Close Window’. Do not replicate this standard method by providing ‘Close’ or ‘Dismiss’ buttons in any non-modal window.

- **OMS– 58149:** 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– 58152:** If the cursor is in the window that the user requests to be closed, then it must be moved to a previous block that is not in that window before the window can be closed.

- **OMS– 58150:** When a user closes a master window, all of its details must be closed, as well as any associated Find windows.

- **OMS– 73036:** If both a Summary and Detail window exist (as with Combination Blocks), when they are both closed all details must be closed as well. Often this coincides with exiting the form.

- **OMS– 73037:** A detail window must close if a context change is made in a master window such that the objects shown in the detail window are no longer applicable.

- **OMS– 73038:** Closing the First Window of a form is the method to exit the form, in addition to ‘Action, Close Form’.

- **OMS– 73039:** Closing a parent window displays a message asking whether to save. Child windows do not require the message as 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 was not previously saved.

- **OMS– 73040:** No windows can be closed while in enter-query mode. Attempting to close a window in this mode elicits a ‘beep’.

### Resizing

Guidelines for resizing windows are as follows:
OMS– 73041:  
• 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– 73042:  
• Only in folders and exceptional cases do windows respond to resize events and reformat the items in the window, usually by displaying more or fewer columns.

OMS– 73043:  
• All non–modal windows should allow minimization (iconification).

OMS– 73044:  
• Only windows that respond to resizing (such as folders) allow maximization; all others do not.

• 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

OMS– 73045:  
All non–modal windows inherit the menu.

Find Windows

Find Windows are a type of window that enables the user to locate one or more records without having to invoke enter–query mode. 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.

OMS– 73645:  
A Find window should be used in the following situations:

• For multi–row blocks and Combination blocks that do not auto–query

• 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). For single–record blocks, though, the results of the search must be reduced to a single–record (using a List of Values after the Find Window if necessary) before being displayed.
Other blocks may instead use an LOV (called a “Row LOV”) that shows all possible records a user can query.

Query Find (See page 1–20)

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

Figure 3–5

For information on implementing Find Windows, see the Oracle Applications Developer’s Guide.

Find Windows
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.
- 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 QBE. Alternative Regions should be used when the number of criteria is very large.
- Find windows are non-modal.
- The window title should be of the format ‘Find <objects>’.
A Find window initially opens centered on its Results window, but retains its position if moved by the user. 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.

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 (i.e. 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 Close Window or the operations menu).

A Find window automatically closes when its corresponding Results window is closed.

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.

Buttons

Find windows have the following buttons:

Clear
Clears the Find window for entering new search criteria. Defaults should be re-applied.

New
Places the cursor on a new record in the transaction block. If the form allows entry of more than one entity or it just makes the action 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).

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.

Find
Performs the query with the current criteria. This is the default button for the window.
Search Behaviors

OMS– 73052:  
- Next Block performs the same function as the Find button.
- [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.

OMS– 73053:  
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– 73054:  
- Multiple sets of search criteria (records) are allowed in Find windows.

OMS– 73554:  
Use Go–Next Record or Go–Previous Record to either display a previously entered search or enter a new search. This allows users to easily make slight modifications to a prior search and run it again.

OMS– 73055:  
- Despite the existence of a Find window, the block being queried should also allow QBE.

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

- Most fields within a Find window should validate, but with minimal cross-field validation.

OMS– 73555:  
- 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– 73556:  
- If no records are located after pressing ‘Find’, the cursor focus remains in the Find window.

Row LOV Windows

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

OMS– 73557:  
A Row LOV should be used in the following situations:

- 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 auto–queries. In general, the types of multi–record or combination blocks that auto–query include detail blocks, and master blocks where the number of records is small.

Figure 3 – 6 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.

List of Values (LOV) (See page 4 – 23)

Modal Windows

Modal windows force the user to work within a single window, then either accept or cancel the changes they have made. They should only be used when a user is required to enter specific information to complete an action.
For information on implementing modal windows, see the Oracle Applications Developer’s Guide.

Modal Windows
Oracle Applications Developer’s Guide

**Position**

OMS– 60053: Modal windows are always opened centered on the screen or centered relative to some other window. They are re-centered each time they are opened.

**Title**

OMS– 73558: Modal window titles should be closely related to the labels of the widgets that open them.

**Menus and Toolbar**

OMS– 73057: Modal windows with one or more text fields on them should allow access to the menu and toolbar, so the user can access the following:

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

OMS– 73058: If the Modal window allows multiple records, then the following should also be enabled:

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

OMS– 73059: All inapplicable menu and toolbar items should be disabled. Modal windows without text fields should not allow menu or toolbar access.
Functions that are disabled should issue a 'beep' when the user invokes them.

**Closing**

*OMS– 73060:*

Modal windows are not closeable with the native GUI window close mechanism. They are closed only in response to the user pressing a button that concludes his work in that window. Typical buttons and how they are used are as follows:

- **OK or Done**
  Closes the window. In some cases, it may perform a save as well. 'OK' is used if the user cannot access the menu, and 'Done' is used when the menu is accessible.

- **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. This should only be used when changes in the modal window will visibly change other windows.

**Implicit Saves (See page 0 – 3)**

*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'.

**Size**

*OMS– 73061:*

Modal windows do not allow resize, maximization, or minimization.

---

**Semi–Modal Windows**

Sometimes a user may not return to a calling window until an action is completed or cancelled, but needs to be able to move among multiple windows for doing that action. In those cases, “Semi–Modal” windows are used.

*OMS– 73561:*

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 cancelling the action (typically with Done or Cancel buttons). Clicking on the calling window results in a beep, and the user returns to where they were.
Semi-modal windows are not a native Oracle Forms feature. See the *Oracle Applications Developer’s Guide* for further information.
Canvasses

Canvasses are the surfaces on which all interface items are placed. In general, all canvasses have the following properties:

- No bevels are used around the edge of the canvas.
- They use the Canvas color (gray).

This section describes aspects of canvasses that depend on whether the canvas is a content canvas or a stacked canvas.

For information on implementing canvasses, see the *Oracle Applications Developer’s Guide*.

Content Canvasses

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

**Display Characteristics**

Content canvasses have the following display characteristics:

- All content canvasses are marked to Display immediately.
- Content canvasses do not raise on entry.

**Size**

The size should be the same as the window it will be shown in, even though the size automatically adjusts at runtime.

Stacked Canvasses

One or more stacked canvasses may be displayed in front of the content canvas of a particular window. If needed, a stacked canvas may fully occupy a window.

Alternative Regions (See page 3–51)
For information on implementing stacked canvasses, see the *Oracle Applications Developer's Guide*.

Stacked Canvasses

*Oracle Applications Developer's Guide*

### Display Characteristics

Stacked canvasses 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 Displayed.
- Stacked canvasses always raise on entry.

### Size

OMS– 73567:  
The Size should be set to the exact size necessary to contain all of the items on the canvas.

### 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 view, and at most three to five fields (OMS–73122).

### Sequence

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

If a stacked canvas may scroll, then a scroll bar must be enabled. Also, if any canvas requires a scroll bar in Alternative Regions, all of the stacked canvasses corresponding to that Alternative Region should then have a scroll bar. Canvasses 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.

The following picture illustrates horizontal scrolling by showing the fields that one must normally scroll to the right to see.
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:
  - 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.
The title is drawn as in the following picture. This picture shows the origin and alignment of the title, as well as the correct distance from the left edge of the block banner.

A horizontal line is drawn across the entire width of the content canvas to identify the block. Draw this line on the row immediately above the block, as follows:

- Thickness: 2 points
- Bevel: Inset
- Line Color: Black

In front of this horizontal line, the title is displayed as follows:

- Background color: Canvas (gray)
- Horizontal Origin: Left
- Vertical Origin: Center
- Horizontal Position of Origin: snapped to a grid line, 2 character cells to the right of the left edge of the canvas.
- Vertical Position of Origin: snapped midway between grid lines
- Alignment: Start
Note: These settings apply specifically to titles that are displayed with boilerplate. Appropriate settings must be applied when the title is a display item or poplist.

OMS– 73571: The title text includes one leading and one trailing space, to prevent the underlying line from directly butting up to the text.

OMS– 73572: • The widget that displays the title may be any of the following:
  – Boilerplate (for static block titles)
  – Display Item, designed to look like boilerplate (for dynamic block titles)
  – Poplist (for Alternative Regions and/or Blocks)

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.

Figure 3 – 9

![Context Blocks Diagram]
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).
- Context fields are placed in a single-record format, with the highest master context located first.
- 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.
- 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.
- 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.

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'.

  Modal Windows (See page 3 – 15)

- Only in exceptional cases should a Dialog block allow multiple records.

Enabled Functions

Most standard Oracle Forms functions do not apply in a dialog block. However, some functions may be enabled under certain conditions.
See the Menu and Toolbar section of the Modal Windows essay for a description of conditions and a list of functions that can be enabled.

Modal Windows – Menu and Toolbar (See page 3 – 16)

Navigation

OMS– 73074: 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 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– 73075: • [Tab] must be restricted to fields within that window.

OMS– 73076: • Navigation Style of the block must remain on the current record.

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

OMS– 73174: • 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).

OMS– 73175: • If the block contains multiple regions, the tab sequence should move between the items of a region before moving to another region within the block.

OMS– 73175: • Prompts should be placed to the left of the field (OMS–75009).

Single–Record Block Prompts (See page 5 – 7)
• Fields should be vertically stacked unless space permits gaps between rows. For more information on item spacing, see Chapter 4 (OMS–74009, OMS–74010, OMS–74020).

  Text Items (See page 4 – 3)
  Display Items (See page 4 – 9)

• Gaps can be used to group logically related fields. Visually grouping fields by using gaps can be used as an alternative to a region box 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.

### Querying

**OMS– 73075:**

• In cases where there is only one record possible, Query Enter, Query Run, Query Find, and Query Find All should all automatically bring up that one record. Query Enter and Query Find should issue a 'beep' in this situation to indicate that their behavior is different from normal.

**OMS– 73076:**

• In cases where the user must use the Find window to query records, Query Enter, Query Run, Query Find, and Query Find All should all automatically bring up the Find window. Query Enter, Query Run, and Query Find All should issue a 'beep' in this situation to indicate that their behavior is different from normal.

### Clearing

**OMS– 73077:**

• In cases where there is only one record possible, Clear Record, Clear Block, and Clear Form should all automatically query that one record. These functions should issue a 'beep' in this situation to indicate that their behavior is different from normal.
**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 Chapter 1.

[Multi–Record Formats (See page 1 – 13)]

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

[Multi–Record Blocks
*Oracle Applications Developer’s Guide*]

**Scroll Bar**

OMS– 73078: Multi–record blocks display a record scroll bar, as follows:
- position: 0.1”
- Scroll Bar Orientation: Vertical
- Scroll Bar Width: 0.2”
- Height: Same as items in the block
- Reverse Direction: False
- Canvas: The same canvas as the block content canvas.
- 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. Such multi–record blocks do include a current record indicator, however.

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: The indicator is 0.1” wide.
- Stacked between the scroll bar and the first column of the block with no blank space on either side of the indicator (see Figure 3 – 10).
- It is Blue on the current row, and canvas (gray) on all others.
- It has a lowered bevel.
- Clicking on it moves to the first item in the record.

Figure 3 – 10

- If the block supports a ‘drill–down’ capability, then the indicator has the same characteristics as above, except for the following: it is 0.2” wide and has a raised bevel.
- 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 auto–query 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 was 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 top.
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 Chapter 5 for more information on cases where a field prompt is wider than the field.

Multi–Record Block Prompts (See page 5 – 9)

If, for some reason, the window is wider than the multi–record block, then the multi–record block should be positioned on the left side of the window, rather than centered.

Querying

In cases where the maximum number of possible records is less than or equal to the number of rows shown in the window, Query Enter, Query Run, Query Find, and Query Find All should all automatically bring up all possible rows. Query Enter and Query Find should issue a 'beep' in this situation to indicate that their behavior is different from normal.

In cases where the user must use the Find window to query records, Query Enter, Query Run, Query Find, and Query Find All should all automatically bring up the Find window. Query Enter, Query Run, and Query Find All should issue a 'beep' in this situation to indicate that their behavior is different from normal.

Clearing

In cases where the maximum number of possible records is less than or equal to the number of rows shown in the window, Clear Block and Clear Form should automatically requery all possible rows. These functions should issue a 'beep' in this situation to indicate that their behavior is different from normal. The behavior of Clear Record should not be altered in this situation.

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 Chapter 1.

Combination Blocks (See page 1 – 14)

For information on implementing Combination Blocks, see the Oracle Applications Developer’s Guide.

Combination Blocks
Oracle Applications Developer’s Guide

Combination Block Windows
The Summary and Detail windows of a Combination block should behave the following way (see Figure 3 – 3):

- OMS– 73084: The Summary and Detail windows should have the same width.
- OMS– 73085: The Detail window is positioned just below the title bar of the Summary window, with the left edges aligned.
- OMS– 73086: 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– 73087: 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– 73088: 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.

Non–Modal Windows, Position (See page 3 – 6)

Behavior of Combination Blocks
The Summary and Detail blocks of a Combination block should behave in the following manner:

- OMS– 73089: The Toolbar and Menu contain a Summary/Detail command for the user to easily switch between the Summary and Detail windows. This command is enabled only for Combination blocks.
The Summary block has a current record indicator that supports drilldown, which moves the cursor to the Detail block for the current record.

Ideally, fields in either Summary or Detail format should allow update, but only allowing update in the Detail format is acceptable.

Selecting Find will open the Find window, and then will display the Summary view once rows are retrieved. The only exception to this occurs if 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.

Delete Record, Enter Query, and Execute Query, 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.

Changing to a different record in either block changes to that record in the other view as well.

Buttons

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

- **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. ‘Open’ is the rightmost button of the 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, and 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
- 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.

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.
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.

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

The Folder functions can be invoked from the Folder 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. The following lists the Folder functions (from the Folder menu) along with the corresponding Folder Tool palette button (if there is one).

**Note:** Some Folder menu items do not function when you are in Enter Query mode.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>Creates a new Folder. The user must enter a new, unique (per entity and user) Folder name.</td>
</tr>
<tr>
<td>Open</td>
<td>Loads a previously defined Folder. A user can select from a list of his own Folders and any public Folders for the current entity.</td>
</tr>
<tr>
<td>Save</td>
<td>Saves the current Folder. If it has never been named, then reverts to 'Save As' functionality.</td>
</tr>
<tr>
<td>Save As</td>
<td>Allows a user to save a Folder, specifying a name, whether it should auto-query upon loading, whether it should serve as the default for the user upon entry to this form, and whether other users can use the same definition. 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.</td>
</tr>
<tr>
<td>Delete</td>
<td>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.</td>
</tr>
<tr>
<td>Show Field</td>
<td>Opens an LOV displaying fields that can be shown (and are not currently). Selecting a value adds the field after the current cursor position.</td>
</tr>
<tr>
<td>Hide Field</td>
<td>Removes the current field. The cursor moves to the field sequenced after the field that was just cut.</td>
</tr>
<tr>
<td>Move Right</td>
<td>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.</td>
</tr>
</tbody>
</table>
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.

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.

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.

Show Order By  Allows the user to show or hide the Order By buttons. When they are shown, the user can set the ordering for the first three items currently shown that support ordering. Each column can be set to Ascending, Descending, or Unsorted. The settings are applied left-to-right when data is fetched and sorted.

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

OMS– 73095: 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– 73096: The Open Folder icon is positioned 0.1” from the left edge of the window.

OMS– 73097: The title is positioned 0.4” from the left edge of the window, on the same row as the Open Folder icon.

OMS– 73098: The Folder title is blank until the user creates a new Folder or opens an existing one.

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.
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 UI 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

A Find Block:

- is displayed in the same window and above the detail records.
- contains one or more fields to specify an already-existing primary key or multi-part primary key.
- has a single button labeled ‘Find’ and located on the right side of the block.
- is separated from its detail records by a block separator line with a title.

Blocks (See page 3 – 23)

- 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 block separator line above it with a title like “Find Object” where the object being found is identified in singular.

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.
- 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.

- 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.

- Previous Block from within a Find Block causes the message "At first block" to be displayed.

### 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. Alternative Blocks are essentially the same as Alternative Regions (also discussed in this chapter) with the difference being that the user switches blocks rather than regions.

- In Alternative Blocks, selecting a value from the poplist causes the appropriate block to display. Each of these blocks may, in turn, have Alternative Regions. When this occurs, the poplist contains all the regions for all the blocks, with regions pertaining to the same block prefixed by the block title in the format ‘block title: region title’.

#### Alternative Regions (See page 3 – 51)

### 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.

#### Master–Detail Relations

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’.
Coordination

Coordination between master and detail blocks should follow these standards:

OMS– 73104:

- 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.

Figure 3 – 14

A check box is drawn with no label, and is positioned as follows:

- If coordination applies to the whole window, the checkbox is placed on the top line of the window.

- If there is a block boundary line separating the detail block, and the coordination check box applies only to that block, then the coordination check box should be placed on the block boundary line in the rightmost three character cells as shown in Figure 3 – 9.

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

- If none of the above apply, a block banner is created to place the checkbox on.

OMS– 73105:

The coordination check box is related to the coordination behavior as follows:

- 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 window).
- 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 checkbox is changed by the user, a message describing the new coordination status is shown on the message line.

OMS– 73107:  
• 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– 73108:  
• When a detail block is in the same window as its master, and both blocks are visible simultaneously, they should usually be immediately coordinated.

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.

OMS– 73109:  
• 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.

Masterless Operations

OMS– 73110:  
A user cannot enter or query detail records until in the context of a master record.
Other Rules and Behaviors

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

OMS– 73650:

- The ‘topmost’ master block of a form does not auto-query unless
  - only a very small number of records will be returned
  - the query will be fast
  - most likely the user will operate on one or more of the queried records
  - Folder blocks may auto-query, but this is determined by each user as part of the Folder definition.

OMS– 73651:

- 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.
Regions

In general, all regions are denoted by a rectangle or line that separates the fields of the region from others in the block. There are several specific variations on placement and cosmetics depending on the usage of the region.

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

Title

- The title is drawn as in the following picture. This picture shows the origin and alignment of the title, the correct distance from the left edge of the region, and the leading and trailing space in the title text.

Figure 3 – 15

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

**OMS– 73111:**
- The title of the region is the name of the group of items it contains (usually a noun).
OMS–73750: • The title is required for most regions, except where the region contents have an obvious function.

OMS–73751: • The widget that displays the title may be any of the following:
  – Boilerplate (for static region titles)
  – Display Item, designed to look like boilerplate (for dynamic region titles)
  – Poplists (for Alternative Regions)
  – Check Boxes or Option Groups (for when an entire region may be applicable or inapplicable)

OMS–73112: • The title is positioned along the top of the region as follows:
  – Horizontal Origin: Left
  – Vertical Origin: Center
  – Horizontal Position of Origin: snapped to a grid line, 1.5 character cells to the right of the left edge of the region
  – Vertical Position of Origin: snapped midway between grid lines

OMS–73113: • The title is positioned in front of the rectangle or line.

OMS–73114: • The title text includes one leading and one trailing space.

OMS–73115: • The title text is always in boldface type.

**Cosmetics**

A region is bounded by either a rectangle (for a single-record format region) or a line (for a multi-record format region), displayed with the following characteristics:

  – Thickness: 2 points
  – Bevel: Inset
  – Line Color: Black

**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 regions or fields where the tab sequence may be unpredictable.

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

**Figure 3 – 16**

**Regions in Single–Record Formats**

**Appearance**

Regions in single–record blocks should have the following look:

**OMS– 73117:**
- A rectangle is drawn around the region, with its corners drawn in the center of character cells.
- The vertical sides of the rectangle may be displayed in the first and last column within the window.

**OMS– 73350:**
- Multiple regions may be drawn side–by–side, or top–to–bottom, but ideally should form rectangles when viewed together.
- If two regions 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. Figure 3 – 18 shows an overflow region as well as the [Tab] sequence when the items in the overflow region are navigable.

For general information on when overflow regions should be used, refer to Chapter 1.

Overflow Regions (See page 1 – 14)

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. It is strongly recommended that you leave a gap of at least one half row between the multi-record block and the overflow region.

- 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.
Cosmetics

The region may be drawn without a surrounding rectangle if it is clear that the fields pertain to the current record of the multi-record block, and no title is necessary.

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.
- 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 you are on 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. The tab sequence is illustrated in Figure 3–18.

Figure 3–18

![Diagram of overflow region navigation](image-url)
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.

Regions in Multi–Record Blocks

Appearance

Regions in multi–record blocks should have the following look:

- A horizontal line is drawn in the row immediately above the region, spanning the fields of the region. A region title is drawn in front of this line.

- When two or more regions are drawn contiguously, reduce the width of the line denoting the leftmost region by one character on the right side. The following picture illustrates this:
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. Also, unlike Alternative Regions, the user has no indication of what other information scrolling will reveal until they scroll.

OMS–73220: Scrolling of regions should only be employed under the following conditions:

- 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.

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

OMS–73121: The region must be its own stacked canvas, and must be presented entirely within any region cosmetics.
• 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 viewport.

• A region that scrolls must show a scroll bar.

• If any one of the multiple regions of an Alternative Region scrolls, then all of those stacked regions should have scroll bars. The scroll bar is visibly disabled for the regions where there is nothing to scroll to.

• Do not scroll the primary identification fields.

• Regions in multi-record blocks scroll horizontally. Regions in single-record blocks may scroll vertically, but this is discouraged unless absolutely necessary.

• In a multi-record block, regions that scroll are separated from fields on the content canvas by a one character gap.

• In a multi-record block, if there is only one scrolling region, the normal region cosmetics may be omitted.

### Alternative Regions

A block with multiple regions that cannot be displayed simultaneously uses a series of stacked canvasses to display each region, one at a time, within a single region boundary. These ‘stacked’ regions are called Alternative Regions.

For information on implementing Alternative Regions, see the Oracle Applications Developer’s Guide.

An Alternative Region (the multi-row case) is displayed as shown at the top of the following figure:
Alternative Regions have a poplist control element containing all possible regions for that block. Figure 3 – 20 shows an open Alternative Region poplist.

### Look of the Alternative Region Poplist

The look of an Alternative Region poplist is determined by the following points:

**OMS– 73128:**
- The poplist control element that lets you choose the visible region is located where the region title would normally be placed, and sized to fully display the longest region title.

**OMS– 73333:**
- In cases where, due to installation or other setup conditions, only one region can ever be seen despite the existence of several of them, the poplist should be hidden exposing static boilerplate for the region below it.

**OMS– 73334:**
- In cases where the Alternative Region poplist starts out with two or more valid choices but at some point gets reduced to only one valid choice, leave it as a poplist (rather than hiding the poplist and exposing boilerplate).

### Behavior of the Alternative Region Poplist

Alternative Regions poplists should behave according to the following standards:

**OMS– 73129:**
- The value displayed in the poplist always corresponds to the current region shown, even if the user does not use the poplist to reveal another region. For example, a user may access another
region through the fields of the block by using [Tab], in which case the poplist must be synchronized with the new displayed region.

**OMS– 73130:**
- When the user’s cursor is in the same block as the Alternative Region, selecting a region from the poplist moves the cursor to the first item of the region.

**OMS– 73131:**
- When the user’s cursor is in a different block from the Alternative Region, selecting a region from the poplist merely displays that region; the cursor context is not changed.

**OMS– 73444:**
- Only valid regions that the user has access to should be displayed in the poplist. However, if it may not be obvious to a user why a region is not available, the value should be in the poplist, and upon selecting it a message must be displayed indicating why the value cannot be chosen, and the value must be reverted to the prior one.

**OMS– 73132:**
- Block Menu invokes an LOV titled ‘Change Region’ that allows the user to select from the same list of choices as in the control poplist. This function is provided for keyboard-only use, because the control poplist is not accessible except via the mouse.

**OMS– 73133:**
- The Alternative Region poplist should have the Queryable attribute set to TRUE, so that it can be used while the block is in enter-query mode. It does not actually impact the query, however.

**Alternative Region Characteristics**

In addition to the poplist characteristics, Alternative Regions should have the following characteristics:

**OMS– 73134:**
- A set of fields, minimally the primary key fields, should be excluded from the Alternative Regions and remain frozen on the content canvas so the user can maintain context.

**OMS– 73135:**
- Each region shown within an Alternative Region area is on its own stacked canvas and contains only the fields of that region and their prompts (no bounding rectangle or title). Each stacked canvas has the following properties:
  - All regions in an Alternative region should be the same size. The size is determined by the minimum size required to display all of the fields in the largest region.
• The Viewport Size is the area immediately within the region boundary.

**OMS– 73136:**
- In a multi-record block, Alternative Regions are separated from fields on the content canvas by a one character gap.

**OMS– 73137:**
- If any one of the multiple regions of an ‘Alternative Region’ scrolls, then all of those stacked regions should have scroll bars. The scroll bar is visibly disabled for the regions where there is nothing to scroll to.
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
- Text Items
- Display Items
- PopLists
- T–Lists
- Option Groups
- Checkboxes
- Buttons
- List of Values (LOV)
- Descriptive Flexfields
- Key Flexfields
- Widgets to Avoid
General Properties

- Each object should be displayed consistently with the same widget, width, prompt, etc., 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’.

- Ideally, an item should be displayed the same across all releases of the product. For example, an item should not be a check box in the first release, and a poplist in the next.

- Items are always sequenced left–to–right, top–to–bottom, within a region, block, or window (OMS–71005).

For more information on general properties of widgets, see the Oracle Applications Developer’s Guide.
Text Items

Single–Row Text Items

Display Field
Enterable Field

No Bevel for display fields

Inset Bevel for entry fields

Multi–Row Text Items

Enterable Multi-Row Fields

Display Multi-Row Fields

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.
### Alignment Rules

<table>
<thead>
<tr>
<th>Values for Text Item</th>
<th>Conditions</th>
<th>Recommended Alignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numeric Values</td>
<td>Numbers that do not display consistent precision. For example: identification or phone numbers.</td>
<td>Start Aligned</td>
</tr>
<tr>
<td></td>
<td>Numbers with consistent precision such as quantities and prices.</td>
<td>Right Aligned *</td>
</tr>
<tr>
<td>Textual Values</td>
<td>All conditions</td>
<td>Start or Center Aligned</td>
</tr>
<tr>
<td>Date Values</td>
<td>All conditions</td>
<td>Start Aligned</td>
</tr>
</tbody>
</table>

Table 4 – 1 (Page 1 of 1)

**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

- **OMS– 70011:** A text item Numeric field must 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.

- **OMS– 74001:** A text item DATE field must be 1.2” wide. This allows 11 characters to be displayed plus one cell to display the bevel. No mask should be associated with this field.

- **OMS– 74002:** A text item TIME field must be 0.8” wide, and should not have a mask associated with it. The field can be 0.6” wide if seconds are not important.

- **OMS– 74003:** A text item as a DATETIME field must be 1.7” wide, and should not have a mask associated with it.

  **Note:** For DATE, TIME, and DATETIME fields, masks will be read from environment variables.

The following picture summarizes the standard lengths of DATE, TIME, and DATETIME fields.
A text item used as a percent field:

- May display percents with a floating decimal point, if extended precision is required.
- May display percents with a fixed decimal point with a precision of two. (preferred)

- Text items should allow mixed-case entry, unless there is a strong business need to enforce either upper or lower case.

- If a text item must show multiple lines, then always enable the scroll bar and set wordwrap to ‘Word’.

- A text item that must fully show between 1 and 10 characters should be sized one average character wider than the maximum contents width to guarantee that the contents will be fully visible.

Example: Suppose there is an uppercase only text item that must fully show three letters. The text item should then be sized to display four characters (0.04”). 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 also accounts for the fact that the field bevels take up a substantial amount of space.
Figure 4 – 2

Cosmetics

OMS– 74009:

• If the text item ever allows input, then it is displayed with the following values (shown above):
  
  Height: 0.25”
  Inter-row Spacing: 0.0”
  Bevel: Lowered
If the text item never allows input but does allow scrolling and/or querying, then it is displayed as follows (shown above):

- Height: 0.21"

- Width: In a multi-record block, the width should be 0.01" less than normal. For consecutive display items, this 0.01" gap may not show up due to rounding. In those cases, make the width of the field 0.02" less than normal.

- Inter-row Spacing: 0.04"

- Bevel: None

- Y position: 0.02" below the gridline

**Note**: Even if all the items in a multi-record block are display items or display-only text items (so that all would aligned), the Y position should still be lowered 0.02" to align with the current record indicator.

- 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.
Behaviors

- Text items with an LOV associated with them almost always employ 'Use LOV for Validation'. This allows a user to type a partial value into the field, and it will autoreduce against the list of valid values.

OMS–74012:
- Text item DATE and DATETIME fields must enable the List lamp. Invoking List on these fields opens the Calendar window.

OMS–74013:
- 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–74014:
- A text item that cannot currently be validated due to a dependency of another field, or cannot be currently entered for any other reason, is displayed with the Canvas (gray) background color, is not navigable, and prevents any typing in the field.

OMS–74015:
- When disabling a text item, make sure that any related fields (such a field that holds a description) are also disabled.

OMS–74016:
- 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).

- 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. 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).

OMS–74017:
- On inquiry forms, where all fields are display-only, all fields should be navigable.
Display Items

**Single–Row Display Items**

No Bevel for
display fields

| Display Field |

**Multi–Row Display Items**

| Display Multi-Row Fields |

**Usage**

**OMS– 74018:**

- Only use display items for fields that never require the user to interact with them in any way, including scrolling or querying. 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**

**OMS– 74019:**

- Display items used as boilerplate must have a width appropriate to entirely display their contents, bearing in mind expansion due to translation, or need to be sized for the current text length at runtime.

**Cosmetics**

**OMS– 74020:**

- When Display items hold data, they are displayed similarly to text items (see Figure 4 – 3):
Height: 0.21”

Width: In a multi-record block, the width of all display items which have another display item to their right should be reduced by .01” so the columns do not run into one another. For example, a column 2.5” wide should become 2.49” wide. This reduction should not be done in Folder blocks, however (the folder code spaces out columns automatically), or as a supplement to the borders of bevelled fields. This .01 gap may be occasionally lost due to rounding error. In those cases, make the width of the field .02” less than normal to achieve the gap.

Inter-row Spacing: 0.04”

Color: Black on White

Bevel: None

Y position: 0.02” below the gridline.

Note: Even if all the items in a multi-record block are display items or display-only text items (so that all would aligned), the Y position should still be lowered 0.02” to align with the current record indicator.

• When Display items are used as boilerplate, they are displayed as follows:
  Height: 0.2”
  Inter-row Spacing: 0.0”
  Color: Black on Canvas (gray)
  Bevel: None
  Y Position: 0.05” below the nearest gridline

• To support bidirectional languages such as Arabic which read right to left, use the following rules to set alignment properties:
  – 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**

**OMS– 74023:**
- Use poplists when only one value is applicable, and the list of choices is never expected to grow beyond 15.
- Poplists are used to control and indicate the current region shown in Alternative Blocks or Regions.
- In some cases, poplists are used as field prompts in single-record blocks.
- Do not use a dynamic poplist when possible values may become inactive. If a value is validated by either an effectivity date or 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.

**Rules**

**OMS– 74524:**
- An attribute modelled with a poplist in an entry form may be modelled with a text item in an inquiry-only form.

**OMS– 74024:**
- 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 and is the default if no default is specified.

**OMS– 74025:**
- 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.
• 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 should be 1” wide.

• A poplist is restricted to displaying 30 characters. To adhere to translation requirements, English text should not exceed 23 characters.

Cosmetics

• Poplists that hold data are displayed as follows:
  Height: 0.25”
  Color: Black on White

• Poplists that are used as control elements are displayed as follows:
  Height: 0.25”
  Color: White on Blue

Figure 4 – 4 shows a poplist used as a control element at the top.

• Poplists that are used as field prompts in single-record blocks are displayed as follows:
  Height: 0.25”
  Color: Black on White
  Font Weight: Medium
Behaviors

OMS– 74130:

- A poplist that cannot currently be validated due to a dependency of another field, or cannot be currently entered for any other reason, is displayed with the Canvas (gray) background color, is not navigable, and prevents any entry in the field.

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

Poplists

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.

Rules

- Always show at least five rows of data.
- T–lists should only be used in low–volume forms with extensive screen real estate available.
- An attribute modelled with a T–list in an entry form may be modelled with a text item in an inquiry–only form.
- In a Find window, also include (Any Value) in the list of valid choices. It should be the last value, but should be the default.
- In a Find window, include (No Value) in the list of valid choices if the T–list in the corresponding entry form is optional. If (No Value) is a choice, it should be placed immediately before the (Any Value) choice.
Option Groups

Usage

OMS– 74031:
- 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.

Good Example

<table>
<thead>
<tr>
<th>Sex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Male</td>
</tr>
<tr>
<td>Female</td>
</tr>
</tbody>
</table>

Bad Example

<table>
<thead>
<tr>
<th>Ship Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPS</td>
</tr>
<tr>
<td>FedEx</td>
</tr>
<tr>
<td>USPS</td>
</tr>
</tbody>
</table>

In the second example, the use of an option group is bad because it is inappropriate for customer–defined lookups (different customers will use different shipping methods).

Rules

OMS– 74032:
- Avoid using option groups for items that may ever appear in multi-record blocks.
- An option group must always have a default value.
- Use in place of a check box when the two states are not accurately modelled as yes/no.
• Can be used to set a ‘mode’ of a form, such as what type of information should be displayed.

• 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’.

OMS– 74232:
• An option button should have an access key when it is used to set a ‘mode’ of a form.

Access Keys (Mnemonics) (See page 5 – 16)

OMS– 74033:
• 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.

Cosmetics

OMS– 74034:
• Draw the buttons of an option group in their own region, where the name of the item is the region name, and the individual buttons are labelled elements within the region.

EXCEPTION: If the title of an option group is obvious, then the region title and boundary may be omitted. This should not be done unless necessary though, as the box helps to indicate that the options are mutually exclusive and visually different from check boxes.

OMS– 74234:
• Option buttons may be laid out vertically or horizontally.

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

O Choice – A.
O Choice – B.
O Choice – C.

Horizontal Layout requires less space and the keyboard access method is more intuitive.


Querying a row with ‘invalid’ option group data

OMS– 74235: For attributes modelled 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 labelled ‘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 if they change the value.

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.

<table>
<thead>
<tr>
<th>Good Examples</th>
<th>Bad Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>☑ Allow Override</td>
<td>☑ Male</td>
</tr>
<tr>
<td>☑ Receipt Required</td>
<td>☑ Root Menu</td>
</tr>
</tbody>
</table>

In the first bad example above, the use of the check box is contrived to represent the Male/Female choice as a question with a yes/no response. The second bad example demonstrates poor usage of a check box because the opposite of Root Menu is not obvious.

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

Rules

OMS– 74035:  
- A check box item is mandatory, and must always have a value (including a default value).

OMS– 74036:  
- In a Find window, an attribute normally modelled with a check box is modelled 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– 74037:  
- 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.
Cosmetics

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

  ![Check Box Example]

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

  ![Check Box Example]

  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.

OMS–74038:

- Check boxes in multi-record blocks are 0.2" wide, with no label (the prompt for the field is displayed with boilerplate above the column using a connector bar centered over the first character cell).

Entering at least one selection

OMS–74238:

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.

For information on implementing Check Boxes, see the *Oracle Applications Developer’s Guide*.

![Check Boxes]

*Oracle Applications Developer’s Guide*
Coordination Check Boxes

Coordination check boxes are used to control master-detail relations between blocks. See Chapter 3 for a complete discussion of their characteristics.

Master-Detail Characteristics (See page 3 – 40)
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 menu.
  – Try to separate functions at a logical break point when possible. For example, using buttons for all of a form’s actions and placing the navigation functions in the Special menu.
  – When possible use the same standard product-wide.

Rules

OMS– 74039:
• Try to use only one row of buttons per window.

OMS– 74039:
• 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– 74040:
• Place the default button in the lower right corner of a window.

OMS– 74041:
• 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-labelled buttons the same.

OMS– 74041:
• 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– 74042:
• 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 chapter on text for more information on expansion requirements for button labels.

General Properties (See page 5 – 2)
• 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

OMS–74043:
• All buttons have an Access key except 'OK' and 'Cancel' within dialog windows.

  Access Keys (Mnemonics) (See page 5 – 16)

Cosmetics

OMS–74044:
• Textual Buttons are displayed as follows:
  Height: 0.3”
  Color: Black or Canvas (gray)
• Iconic buttons within a screen are always 0.25” by 0.25” (toolbar iconic buttons, are 0.3” by 0.3”).

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

  Buttons
  Oracle Applications Developer’s Guide
List of Values (LOV)

Usage

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

Rules

OMS–74045:
- Position all LOVs with the upper left corner at (1”, 1”).

OMS–74046:
- Width: Maximum: 7.8” Minimum: 3” Any width between these is allowed. However, the following guidelines are recommended:
  - The field length in the LOV should at least match the length of the field in the window. If the field in the window uses a two line prompt then the field length of the LOV should be the total length of the prompt plus 30% for translation.
- Since the user cannot scroll values in the LOV, you may need to leave more space in the LOV than in the window.
- LOV window should be smaller than the window from which it was invoked so the LOV does not totally obscure it.

**OMS– 74047:**
- The title is the name of the object in the LOV, and is plural i.e., '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– 74247:**
- The prompt of the first column is related to, or matches identically, the prompt of the item that invoked it.

**OMS– 74248:**
- 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.

**Cosmetics**

**OMS– 74249:**
- All LOVs use the default visual attributes.

**Behaviors**

**OMS– 74250:**
- LOVs do not display automatically upon navigating to a field.

**OMS– 74451:**
- LOVs automatically select a row when the list of valid choices is reduced to one.

**OMS– 74452:**
- After selecting from an LOV, the cursor automatically moves to the next field except when:
  - the next field is in an alternative region not currently displayed.
  - in a Folder form
  - in mirror items of fields in Folders, like in Combination Blocks.
  - it is the last navigable field in a multi-row block with horizontal scrolling.

**OMS– 74048:**
- If an LOV may show more than 100 records, then the user must be prompted to reduce the list of valid values first.
- 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 SQL.
- The validation clause is too costly to evaluate in SQL.
- 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.

- In most forms, do not provide LOVs in enter-query mode. Never do so in simple setup forms; occasionally provide them in major transaction forms.

When providing an LOV in enter-query mode or a Find window, it should select all the primary key values from the foreign key table, restricted only by those records which could never be valid for the current field.

EXAMPLE: An LOV for vendors in the PO form in a Find window enter-query mode shows all vendors that could ever be placed on a PO, not just the set of vendors that currently are allowed to be placed on a PO.

Do not reuse the entry LOV in enter-query mode unless it provides the correct set of data for both modes.

For information on implementing List of Values, see the Oracle Applications Developer’s Guide.

| Lists of Values |
| Oracle Applications Developer’s Guide |
Editors

The Editor is invoked by pressing the Edit toolbar button, or by selecting Edit from the 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).

OMS–74350: On a DATE field, the Edit action opens the Calendar.

On a Flexfield, Edit opens the Flex window (OMS–74058).
Descriptive Flexfields

**Usage**

- All entities should provide a Descriptive Flexfield to allow customization.

**Cosmetics**

- Code a descriptive flexfield as a text item, displaying two characters (width of 0.2”).
- The prompt associated with the descriptive flexfield is ‘[ ]’. In a single-record block, draw the characters to the left and right of the field and centered in their respective character cells; in multi-record blocks, draw them as [space space] immediately above the column, centered horizontally and vertically. The following picture illustrates the placement of the prompt.

**EXCEPTION:** In multi-row folders position the display item that shows the brackets down 0.05” from the top similar to other column headings.
Figure 4–5

OMS–74052:

- The descriptive flexfield is located as the last item in each block, on the content canvas. When an alternative region exists, the descriptive flexfield is located after the region (not as an item within an alternative region).

EXCEPTION: Descriptive flexfields are not placed on the content canvas in folders.

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.

OMS–74252:

- If a descriptive flexfield is part of a row that is always display-only, then the descriptive flexfield should be drawn without a bevel (following the conventions for display-only text items).

Behaviors

OMS–74053:

- The descriptive flexfield Forms field should be navigable, but should not allow input. It adheres to all text item standards.

- Flexfield code will automatically show the concatenated values in the descriptive flexfield Forms field.
• The List lamp must not be enabled for descriptive flexfields.

• Descriptive flexfield uses a user-level profile that determines whether the flex window should pop open upon the user navigating into the field.

  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 (flex does this automatically):
  – 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 (flexfields do this automatically):
  – Any successful navigation out of the window moves the cursor to the next field.
  – Cancelling 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 themselves.

For information on implementing Descriptive Flexfields, see the Oracle Applications Developer’s Guide.
Key Flexfields

Usage

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

Behaviors

OMS– 74055:
- All key flexfields use the direct-entry method.

OMS– 74056:
- Only key flexfields with Validation set to Full should enable the List lamp.

OMS– 74057:
- Invoking a List of Values in a flexfield allows for entry of a combination-level List of Values.

OMS– 74058:
- Invoking Edit in a flexfield expands the field to allow entry of values in individual segments.

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

Cosmetics

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

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

Key Flexfields
Oracle Applications Developer’s Guide
Widgets to Avoid

The following widgets available in Oracle Forms should be avoided due to either usability problems or cross-platform issues.

**Combo Boxes**

A combo box is a hybrid of a text item and a list. It allows a user to select a value from the list, or type a new entry directly into the field, with no validation against the list. Because of the inconsistency this creates with text items that have LOVs associated with them and poplist items, we do not allow the use of combo boxes in Oracle Applications.

**OLE and VBX controls**

Because these controls only operate on the MS Windows operating system, if you are developing a product that must run on other platforms, including a web browser, you cannot rely on these widgets. The same reasoning applies to the use of OLE automation as well.
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
- Properties of Prompts and Titles
- Single–Record Block Prompts
- Multi–Record Block Prompts
- Conventions
- Access Keys (Mnemonics)
- Using Data as Prompts
- Errors
- Warnings
- Questions
- Information
- Hints
- Messages
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:

- 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 Appendix B. This 30% expansion rule applies to the number of ‘average’ characters, not inches. The 30% rule applies to different types of text as follows:

<table>
<thead>
<tr>
<th>Text type</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prompt on left of field</td>
<td>leave 30% expansion room to left of prompt</td>
</tr>
<tr>
<td>Prompt above left justified field or list</td>
<td>leave 30% expansion room to right of prompt</td>
</tr>
<tr>
<td>Prompt above right-justified field</td>
<td>leave 30% expansion room to left of prompt</td>
</tr>
<tr>
<td>Prompt above center-justified field or check box</td>
<td>leave 30% expansion room, divided equally to the left and right of the prompt (i.e. 15% on each side)</td>
</tr>
<tr>
<td>Buttons</td>
<td>size to allow 30% expansion of the label + 0.2”</td>
</tr>
<tr>
<td>Check boxes in single-record blocks</td>
<td>size to allow 30% expansion of the label + 0.3”</td>
</tr>
<tr>
<td>Poplists</td>
<td>size to allow 30% expansion + 0.5”</td>
</tr>
<tr>
<td>Option Groups</td>
<td>size every button the same, such that the widest label can expand 30%</td>
</tr>
<tr>
<td>Region Titles</td>
<td>size underlying rectangle or line such that title can expand to the right by 30%</td>
</tr>
<tr>
<td>Display-only fields</td>
<td>size 30% wider than needed in English, except for fixed-width fields (date, time, etc.)</td>
</tr>
</tbody>
</table>

Table 5–1 (Page 1 of 1)
OMS– 75002:

- 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 contents width 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).

  **Note:** The two prior standards are commonly referred to as the ‘30%/1” rule. It is important to always remember that these are the minimum spacing requirements; whenever possible leave additional space for translation.

OMS– 75003:

- 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. However, some abbreviations have been approved for use in Oracle Applications and are listed in Appendix B.

  If you have room, use the full (non-abbreviated) 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.
Properties of Prompts and Titles

Prompts are labels for fields, check boxes, lists, etc. Titles are labels for blocks and regions. The following rules and characteristics apply to each:

**OMS– 75004:**
- 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 pt., medium weight
  - Edge: Clear
  - Fill: Clear
  - Text color: Black

**OMS– 75007:**
- All titles have the following characteristics:
  - Font: 10 pt., bold weight
  - Edge: Clear
  - Fill: Canvas (gray)
  - Text color: Black

**OMS– 75008:**
- Prompts and Titles should only use Start, Center, and End for alignment settings.
- Display items may be used to show dynamic prompts and titles (OMS–74018). Guidelines for such use are as follows:
  - 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.
OMS–75024:

- 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).

OMS–75024:

- All fields require unique prompts to identify them, except for some display-only fields whose prompts are ‘obvious’. Examples are:
  - Vendor Name and Vendor Number:

    | Vendor Name | AT&T | Number | 00012 |
    |--------------|------|--------|-------|

  - Item and Item Description fields when the description is display only:

    | Item  | Ball Bearing Assembly |
    |-------|-----------------------|

  - Amount and Currency if the currency type is defaulted:

    | Amount | USD | 123.00 |
    |--------|-----|-------|
Single-Record Block Prompts

- Prompts are positioned to the left of an item, with one character cell 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 illustrated in the following picture and enumerated below:

Figure 5 – 1

- Horizontal Origin: Right
- Vertical Origin: Center
- Horizontal Position of the Origin on the Grid: 0.1” to the left of the start of the associated item
- Vertical Position of the Origin on the Grid: Centered between the gridlines. Multi-line prompts must be positioned so that the first line of text is positioned centered on the first row that the associated item occupies.
- Alignment: End
• 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 be black on white with a font weight of Medium (OMS–74030).

OMS– 75026:  
Each prompt value in the list should have a trailing colon (e.g. “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 below).

• 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 below) (OMS–75011). Do not use the 'label' property for check boxes in a multi–record block.

• A prompt which pertains to several fields is, in effect, a region. It may be drawn following the exact same standards as a region in a multi–record block. See Table 5 – 1, on page 5 – 2.

Long Prompts

OMS– 75027: There are several options for a prompt that is too long to fit on one line. The preferred option is to put the prompt onto multiple lines. If that does not work (such as in a case where one word of the prompt is still too long), gaps can be left between fields to accommodate the long prompt, or raise the prompt and use a connecting line to the field. If the other options are unacceptable, the prompt can be allowed to overlap an adjacent field under certain conditions. All four options are discussed below.

• When prompts contain too much text to fit on one line, a multi–line boilerplate item should be used (except in folders). 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. Note that the upper line of these prompts will not align with the grid when the prompts are vertically snapped to the bottom as required.

• 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 connecting lines, drawn as follows:
  
  Color: Black
  Thickness: 0 points
  Inset: None

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 horizontal snap point of the prompt should be in 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).

• 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 picture illustrates some of the specific settings for multi-record block prompts, including prompts over check boxes and prompts that use connecting lines.

In Folder forms, prompts cannot be multiline or use connector lines, but do not require the 30% additional space for translation or character width variations (OMS–75004). However, the default column width must be large enough to ensure that the label is not truncated in the base language the form is coded in.
• All prompts for multi-record blocks have the following vertical placement settings:
  Origin: Bottom
  Position of Origin: on the gridline

• Horizontal placement is determined as follows:
  For left-justified data elements, including lists:
    Origin: Left
    Position of Origin: 0.05” to the right of the left edge of the associated field
    Alignment: Start
  For center-justified data, including check boxes (except when using connecting lines with check boxes):
    Origin: Center
    Position of Origin: centered on the width of the item
    Alignment: Center
  For right-justified data:
    Origin: Right
    Position of Origin: 0.05” to the left of the right edge of the associated field
    Alignment: End
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. However, for consistency the 0.05” offset applies to all prompts in multi-record blocks, regardless of whether the items have a bevel or not.

Identical Blocks

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:

<table>
<thead>
<tr>
<th>Good</th>
<th>Bad</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lines</td>
<td>Enter Order Lines</td>
</tr>
<tr>
<td>Print</td>
<td>Run Print Report</td>
</tr>
<tr>
<td>Margins</td>
<td>Set Document Margins</td>
</tr>
<tr>
<td>Apply Notes</td>
<td>Automatically Apply Notes</td>
</tr>
</tbody>
</table>

Table 5–2 (Page 1 of 1)

Ellipses

OMS–75012: Ellipses (...) are used at the end of a menu entry or button label if

- the entry or button opens a modal window (for example, the menu entry “About Oracle Applications…”).
  or

- the user needs to provide further information about the action in another window (modal or not) before the action can be completed. Two examples are the menu entries “Query Find…” and “Navigate To…”.

Button examples:

OMS–75029: Ellipses 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 ‘Percent’ in the prompt – always use ‘%’.

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

OMS– 75014: • Single-record:

Prompt %

OMS– 75015: • Multi-record:

Prompt %

• If the only reasonable prompt would only be ‘Percent’, then only use ‘%’ after the field or above the column. This occurs when the qualifying word is part of the region title.

Example: A total field in a region called ‘Quota To Date’ could be shown as follows:

<table>
<thead>
<tr>
<th>Quota To Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
</tr>
<tr>
<td>%</td>
</tr>
</tbody>
</table>

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 or like the graphic representation pictured below which replaces "Hire Dates From ____ To____":

Hire Dates _______ - _______
Access Keys (Mnemonics)

Controls that need to be operated with direct keyboard access provide an Access Key (underlined mnemonic access character) to invoke them.

**OMS– 75016:**
- Always provide access keys on the following objects:
  - All textual buttons, except 'OK' and 'Cancel' ('OK' is done by Return and 'Cancel' is done by Escape).
  - Option buttons or check boxes that set a 'mode'. 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.

**OMS– 75032:**
- In order of preference:
  - First letter of button or of key word.
  - First letter of the non–key word
  - Second letter of button label
  - A strong letter of the label like.

**OMS– 75017:**
- Try not to underline letters with descenders such as y, j, q, or p as the underline and letter overlap. (y, j, q, or p)
- Follow common conventions where applicable.
- Access keys must be unique within a window, and must not conflict with the keys used by the top level of the menu (A, E, F, G, H, Q, S and W) even if those menus are not used.
OMS-75033:

- Use the following access keys for these common terms:

  - Clear
  - Done
  - Find
  - Lines
  - New
  - Open
  - Post
  - View
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 two currencies of interest. The results of this analysis is two columns of information, whose most logical prompts are the two currencies being analyzed.

![Image](image.png)
Errors

- Errors are messages that indicate serious errors or problems which the user must acknowledge, and normally cause processing to be halted.

OMS– 75018:

- Error messages show a ‘stop’ sign, 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.

Figure 5 – 4

APP-00666: Passwords must be at least 5 characters long.
Warnings

- Warnings are messages in the form of a question that the user must respond to, and allow the user to continue or abort 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 a warning is inappropriate, i.e., when the message is not intended to provide a way for a user to abort 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?' ('Yes', 'No', & '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.
- Information messages show an ‘information’ sign, 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.’

Figure 5–7

You failed to enter a correct username and password in the required number of tries. Contact your system administrator for help.
Exiting the signon screen.

OK
Hints

- Hints are messages that are of very little consequence, or a ‘progress indicator’ message, that never require any acknowledgement.
- Hints appear on the Forms message line 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 labelled “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 their 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’.
This chapter covers the following topics:

- Validation Models
- Saving Changes
- Navigation within Forms
- Navigation to Forms — Using the “Navigator”
- Linked Forms
- Disabled Functions
- Irreversible Actions
- Running Totals
- Field Ranges (From/To Field Pairs)
- Currency
- Multiple–Record Selection
- Highlighting Information
- Printing
- Long–running Processes
- Ordering of Displayed Records
- About This Record...
- About Oracle Applications...
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. See the “Errors” section for more information. 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 in a WAN environment.
  – 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 offending fields, and corrective actions to be taken, must be issued. The cursor should be moved to the first of the offending 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, are disabled until the other fields(s) are populated and validated.

Disabled Functions (See page 6 – 15)

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 a default value if one exists.
Saving Changes

The user explicitly saves changes by invoking ‘Action, 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 with OPEN_FORM 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, or
- 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 (even if the menu and toolbar can be accessed).

Implicit Saves

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

- 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 aborted.

- 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**

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 labelled 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:

- ‘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 and Lists which currently are non-enterable are shown with a Canvas (gray) colored background.
- OMS– 76043: Text Items that are always display-only (thus are not part of the navigation sequence) are rendered without a bevel.
- OMS– 76044: Buttons and option buttons that are disabled (thus non-navigable) are dimmed.

Navigation with the Keyboard

- OMS– 76005: Keyboard navigation allows the user to visit all items on an object that the user can interact with without mouse intervention.
- OMS– 76006: In some cases there may be non-enterable fields that users may need to visit without using the mouse. 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 inquiry-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.)

EXAMPLE: A PO 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.
• A user can [Tab] past a required field if it is null (it will be trapped when they try to leave the record).

• 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).

• 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

• 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 (OMS–73002). Stated another way, disable fields that cannot currently be validated, and prevent the cursor from moving to them.

Oracle Applications disables most items by applying a combination of UPDATEABLE, INSERT_ALLOWED, and NAVIGABLE properties, and named visual attributes, rather than utilizing the DISABLED property. 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.

• A disabled field still accepts input focus, but prevents typing.

• 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).

• A user cannot click out of a window that is modal.

### Keyboard Navigation to Control Elements

• Navigation should proceed to all control elements, except the following:
  - Alternative region poplists
- Buttons below a multi-record block
- Buttons that apply only when the cursor is on a particular field
- Clear buttons

**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 is never used to move between sibling blocks.

  **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.
• From the first block of the form, issue the message ‘At first block.’.

• Previous Block from a modal window always issues the message ‘At first block.’.

• 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

• 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.

• 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

• Behavior is dictated by the Forward navigation requirement. No concession is made to optimize the behavior of ‘previous’ navigation.

Alternative Region Navigation

• When the cursor is about to enter an alternative region area the target field is the first item of the first alternative 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.

• Within an alternative region, [Tab] moves the cursor to the next sequenced item, regardless of whether it is within the currently shown region or not.

• Within an alternative 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
alternative region is displayed, and the cursor moves to the first item of the record.

- Selecting a choice from the Alternative region control list 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.

- Pressing Block Menu invokes an LOV that duplicates the values and behavior of the Alternative region control list for the current block, or all blocks in the case of an Alternative Block scheme.
Navigation to Forms — Using 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. Refer to the Oracle Applications User’s Guide for more information about the Navigator.

The Navigate Window

- If the Navigate window is hidden behind another window or is minimized, choose from the Action menu, Navigate To....

Figure 6 – 1 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.

- Define Function names with unique first letters, if this can be done without using contrived words. This allows the user to select a menu item by typing the first letter of the menu item.

Note: Items in the navigation list should not begin with a number, as that would interfere with the "top ten list" described below.
• Organize by job function and activities (such as ‘Setup’, ‘Reports’, or ‘Working with P.O.s’) rather than grouped by implementation (such as ‘Requests’, ‘Inquiries’, etc.)

Refer to the Oracle Applications Developer’s Guide for information regarding functions and responsibilities.

The Navigation List

• The forms that a user can navigate to are displayed in a navigation list on the left-hand side of the Navigate window. The navigation list is organized much like the hierarchy of a file system, where a user can expand items that begin with a plus sign (+) to further sub–levels until finding the form of interest. Sub–levels appear indented below the items from which they are expanded. Items that are expanded are preceded by a minus sign (–). An item can expand no further when it displays neither a plus or minus sign.

To expand or collapse menu items the user can either double–click on the appropriate menu item or use the following buttons:

- Expands the currently selected menu item
- Collapses the selected menu item
- Expands all the sub–levels of the currently selected item
- Expands all the sub–levels of all expandable items in the navigation list
- Collapses all currently expanded items in the navigation list

• If there are forms that are used frequently, a user can add them to the top ten list located on the right–hand side of the Navigate window. The top ten list displays forms numerically so you can choose them instantly without having to search for them in the navigation list. A user can add a maximum of ten forms to the top ten list and can create a different top ten list for each responsibility that they have access to.

Note: The top ten list is unique for each responsibility and user sign–on combination.
• Above the navigation list, there are two fields that help identify a selection. The top field displays the name of the item currently selected, while the bottom field displays a description of that item.

  **Note:** The description of the item that you use should match the name used for the top ten list.

• At the bottom of the Navigate window, there is a check box labelled ‘Close Other Forms’ that lets a user choose between being able to have several forms at the same time or just one form at a time. Checking ‘Close Other Forms’ helps reduce system requirements by closing existing windows when the user opens a new window from the Navigator.
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:

• ‘Action, Save’ only saves the changes in the current form.
• ‘Action, Close Form’ closes only the current form, and asks to save changes if any are pending.
• ‘Action, 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 Enter Customers form directly from the Enter 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 Special menu, as follows:

• If all the inquiries that exist for a form can fit within the limit of fifteen entries on the Special menu (along with other features that might be placed there), list each as a separate entry.
• If too many inquiries exist to fit in the Special menu, create a single entry (‘Go To...’) that brings up an LOV or other modal dialog of the available inquiries.

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 Action 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 an Oracle Forms 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. For more information, see the section on Function Security in the Oracle Applications Developer’s Guide.

- 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):

- Widgets should be hidden. This is most easily achieved 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 at all 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:

- 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 pulldown 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:**

1. 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'.

2. 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**

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 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 on a check box.
• Functions within Oracle Forms that can be declaratively disabled, such as querying or inserting new records, will cause Oracle Forms to 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 issue a ‘beep’ when pressed.
• Menu and toolbar entries are disabled (dimmed) when inappropriate for the current block, record, and field.
• If it may not be 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

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 Record’ action 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 if that clarifies.
Running Totals

• Totals show the cumulative values for all rows, not just the subset of rows currently queried or displayed.

• Fields that serve as running totals of detail records are usually drawn in the master block.

• In single-record blocks, when a running total field shows a total for multiple fields which are all visible, the fields should be laid out as:

<table>
<thead>
<tr>
<th>Subtotal1</th>
<th>40.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subtotal2</td>
<td>30.25</td>
</tr>
<tr>
<td>Total</td>
<td>70.25</td>
</tr>
</tbody>
</table>

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

• 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. Below the column that is being totalled:

<table>
<thead>
<tr>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.00</td>
</tr>
<tr>
<td>30.25</td>
</tr>
<tr>
<td>43.25</td>
</tr>
<tr>
<td>Total</td>
</tr>
</tbody>
</table>

The decimal places of the field must align with those of the ‘column’. The line is not drawn as in the single-record case because items may be scrolled out of view, so the values shown may not add up to the total shown.

• 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 to ensure that the value entered in the field itself is sensible; 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 the dates being the same is the most likely case.

- Ranges are interpreted as follows:

<table>
<thead>
<tr>
<th>From Value</th>
<th>To Value</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>null</td>
<td>null</td>
<td>all values</td>
</tr>
<tr>
<td>null</td>
<td>31–DEC–1995</td>
<td>all dates up to and including 31–DEC–1995</td>
</tr>
<tr>
<td>01–JAN–1995</td>
<td>null</td>
<td>all values including and after 01–JAN–1995</td>
</tr>
</tbody>
</table>

Table 6 – 1 (Page 1 of 2)
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:

```
Hire Dates
```

In the Example above, the prompt is plural, and the separator between the fields is a dash centered on three character cells. This is the recommended style.

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

```
Hire Dates
```

If multiple field ranges exist, and the fields are of different widths, try to arrange them as follows:

```
Hire Dates
```

```
Sequence
```
Multi-Record Blocks

In multi-record blocks, range fields are always painted as in the following example:

<table>
<thead>
<tr>
<th>Hire Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
</tr>
<tr>
<td>To</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
**Currency**

**Currency Code**

- Display the currency code rather than the symbol associated with the currency.

OMS– 76038:
- Always put the currency code before amounts on forms, letting the value’s general label (“Amount” in the example below) apply to both the currency code and the amount. Do this for both input and display–only cases.

<table>
<thead>
<tr>
<th>Amount</th>
<th>USD</th>
<th>123.30</th>
</tr>
</thead>
</table>

- Alternatively if all the amounts on the window are in the same currency, you can use a single separate field labelled “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.

| Total | 9972.75 |

- The prompt for the currency code field can be omitted in a multi–row display–only block. In that case, the currency code field should not have 0.01” subtracted from its length, this makes the fields appear to run together (no gap between them by design).

<table>
<thead>
<tr>
<th>Amount</th>
<th>USD</th>
<th>123.30</th>
</tr>
</thead>
</table>

**Field Alignment**

- Currency codes are left–aligned.

OMS– 76039:
- Currency amounts are right–aligned.
Decimal character (Radex)

- Currency amounts should be displayed with the appropriate decimal character. For example, the period is the appropriate decimal character for German currency (123.456) whereas for American currency the appropriate decimal character would be the comma (123,456).

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 the bevels around the field.
- Leave 0.4” for the currency field and 0.1” between the currency and amount.

Positive and 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”. The other display possibilities are angle brackets, as in “<23>”, or a trailing hyphen, as in “23–”.

Validation Messages

- If a user changes the currency code when its associated amount(s) are already entered (non-zero), the user should be warned with a message to verify that the amount is correct for the new currency. This is particularly important as 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, do not mask the input. Instead allow the decimal point to float as needed and display the amount of precision the user enters. This will lose the numeric alignment in multi-row displays, so it should be only done 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 Cyan 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.
- Multiple discontinuous records are selected by using control–click.
- A continuous range of rows is selected by holding down the Shift key 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 ‘Edit, Select All’ from the menu.
- All records can be deselected by choosing ‘Edit, Deselect All’ from the 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.
- Additionally, 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.

- A particularly important row may be highlighted with the Cyan background. In general, limit highlighting to a single row.

- A value may be rendered in color to indicate a certain attribute. Red is used for errors and negative values. For example, a financials total may be rendered in red if negative or when showing inventory quantities, a negative quantity on hold at a certain date may be shown in red.

**Attention:** In order to support users who are color blind, any use of color coding must only be used to augment. 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 negative sign 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 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 (Standard Report Submission) 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:

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

Selecting ‘Print...’ brings 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’ opens the platform–specific print dialog.

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

Selecting ‘Print...’ brings 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

Any process which may take more than two seconds to run should change the pointer to a busy indicator (typically the stopwatch or hourglass).

If possible, provide a ‘progress bar’ for a user to see how progress is being made on the requested action by estimating the % done and displaying the percentage while the process is running. Alternatively, display hints on the message line (that do not require acknowledgment) that specifically state the progress, such as ‘Processed 1 of 17 records...’.

If possible, provide the user a way to abort a long-running process.

Window Level Utilities

Oracle Applications Developer’s Guide
Ordering of Displayed Records

This section gives recommendations for specifying the order (ascending or descending, for example) of queried records. Note that except for the title ‘Order By’, the following are recommendations and not strict standards within Oracle Applications. In general, these ideas are intended to go into a form’s Find window, if it has one. (These suggestions are not intended for the Find windows of Folder forms, however, since ordering for Folders can be specified separately).

- 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., etc. and to their right with ascending/descending icons that can be toggled) for specifying primary, secondary, etc. 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.
About This Record...

A user chooses 'Help, About This Record' in order to see the information automatically supplied by the AOL 'WHO' columns, such as:

- Created By
- Created Date
- Last Updated By
- Last Update Date
- Last Update Login

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...

A user chooses ‘Help, About Oracle Applications...’ to see basic information about the product being run. The About Oracle Applications window provides details about the version of the Oracle Applications components, login information, and information about the current form.

Figure 6 – 4

![About Oracle Applications Window](image)
This appendix contains definitions of certain terms used in this reference manual.
<table>
<thead>
<tr>
<th><strong>Action Buttons</strong></th>
<th>Product–specific functions or navigation that the user can invoke. ‘Post’, ‘Encumber’ and ‘Distributions’ are examples of actions.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Block</strong></td>
<td>The representation of an entity on a form. For example, the Vendor block consists of the attributes pertaining to the Vendor entity.</td>
</tr>
<tr>
<td><strong>Canvas</strong></td>
<td>The surface on which interface items and prompts are physically drawn. Canvasses are displayed in a window.</td>
</tr>
<tr>
<td><strong>Command Button</strong></td>
<td>A textual button with a standard, well–specified behavior, such as ‘OK’, ‘Cancel’, and ‘Done’.</td>
</tr>
<tr>
<td><strong>Context</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Control Element</strong></td>
<td>An element used to control behavior of the Application, such as buttons to open windows or initiate an action, or a poplist to change the region currently shown.</td>
</tr>
<tr>
<td><strong>Dialog Box</strong></td>
<td>A window used to enter information needed to complete a specific action. The user must interact with this window before proceeding.</td>
</tr>
<tr>
<td><strong>Enterable Field</strong></td>
<td>A field that allows the user to type information. A field may dynamically change from enterable to non–enterable as the user operates the form. Non–enterable fields include context, overflow regions, session–disabled, and dependent fields.</td>
</tr>
<tr>
<td><strong>Entity</strong></td>
<td>A unit or natural grouping of elements. ‘Assignments’ and ‘Sales Order Lines’ are examples of entities. An entity may be comprised of several blocks, such as ‘Sales Reps’, ‘Quotas’, and ‘Territories’ (the three things taken together are viewed logically as one, even though they are modelled as three blocks and three tables).</td>
</tr>
<tr>
<td><strong>Folder</strong></td>
<td>A type of block that allows users to customize the screen and query criteria.</td>
</tr>
<tr>
<td><strong>Field</strong></td>
<td>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’.</td>
</tr>
</tbody>
</table>
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 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).

Navigation
The process by which the user moves the input focus between different elements of a form, or to other forms.

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 (e.g. choosing Save from the Action menu, or by
pressing a Save button). Implicit saves are those inherent in the nature of a function (e.g. 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.

**Snap Point**
The point of a widget which corresponds to the (X,Y) position that locates it.

**Toolbar**
A series of iconic buttons that perform generic actions, such as List and Save.

**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 a user could perform a significant portion of their job function 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.
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 % which should always be used in prompts rather than spelling out 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, still fully spell out those with enough room for the full word. Do not abbreviate every usage in pursuit of consistency.

Applications Product Names

<table>
<thead>
<tr>
<th>Product Name Abbreviations</th>
<th>Min. Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP</td>
<td>Accounts Payable</td>
</tr>
<tr>
<td>AR</td>
<td>Accounts Receivable</td>
</tr>
<tr>
<td>CRP</td>
<td>Capacity Requirements Planning</td>
</tr>
<tr>
<td>Eng</td>
<td>Engineering</td>
</tr>
<tr>
<td>FA</td>
<td>Fixed Assets</td>
</tr>
<tr>
<td>GL</td>
<td>General Ledger</td>
</tr>
<tr>
<td>HR</td>
<td>Human Resources</td>
</tr>
<tr>
<td>MDS</td>
<td>Material Demand Scheduling</td>
</tr>
<tr>
<td>Mfg</td>
<td>Manufacturing</td>
</tr>
<tr>
<td>MPS</td>
<td>Master Production Scheduling</td>
</tr>
<tr>
<td>MRP</td>
<td>Material Requirements Planning</td>
</tr>
<tr>
<td>OE</td>
<td>Order Entry</td>
</tr>
<tr>
<td>Product Name Abbreviations</td>
<td>Min. Length</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>PA Project Accounting</td>
<td>.6”</td>
</tr>
<tr>
<td>PO Purchasing</td>
<td>.4”</td>
</tr>
<tr>
<td>RCCP Rough Cut Capacity Planning</td>
<td>.6”</td>
</tr>
<tr>
<td>WIP Work In Process</td>
<td>.4”</td>
</tr>
</tbody>
</table>

| Term Abbreviations |

<table>
<thead>
<tr>
<th>Approved Abbreviations</th>
<th>Min. Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>% Percent (Always use % in prompts)</td>
<td>.1”</td>
</tr>
<tr>
<td>Acct Account</td>
<td>.5”</td>
</tr>
<tr>
<td>CUM Cumulative Qty this model year</td>
<td>.5”</td>
</tr>
<tr>
<td>Curr Currency</td>
<td>.6”</td>
</tr>
<tr>
<td>Cust Customer</td>
<td>.7”</td>
</tr>
<tr>
<td>Depr Depreciation</td>
<td>.7”</td>
</tr>
<tr>
<td>Dept Department</td>
<td>.7”</td>
</tr>
<tr>
<td>Desc Description</td>
<td>.6”</td>
</tr>
<tr>
<td>Expnd Expenditure</td>
<td>.7”</td>
</tr>
<tr>
<td>Ext Extension</td>
<td>.5”</td>
</tr>
<tr>
<td>ID ID / Identification</td>
<td>.4”</td>
</tr>
<tr>
<td>Interorg Inter-organization</td>
<td>.9”</td>
</tr>
<tr>
<td>Max Maximum</td>
<td>.4”</td>
</tr>
<tr>
<td>Min Minimum</td>
<td>.4”</td>
</tr>
<tr>
<td>Num Number</td>
<td>.4”</td>
</tr>
<tr>
<td>Op Operation</td>
<td>.5”</td>
</tr>
<tr>
<td>Org Organization</td>
<td>.5”</td>
</tr>
<tr>
<td>Qty Quantity</td>
<td>.5”</td>
</tr>
<tr>
<td>Reclass Reclassification</td>
<td>.9”</td>
</tr>
<tr>
<td>Ref Reference</td>
<td>.5”</td>
</tr>
<tr>
<td>Rel Release</td>
<td>.6”</td>
</tr>
</tbody>
</table>
Approved Abbreviations | Min. Length
---|---
Res | Resource .7”
Rev | Revision .4”
Seq | Sequence .4”
Ship Docs | Shipment Documents .10”
Ship Set | Shipment Set .10”
Spec | Specification .7”
Subinv | Subinventory .8”
Trans | Transaction .5”
UOM | Unit of Measure .4”
Whse | Warehouse .8”

Terms With No Abbreviation

The following terms 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”).

<table>
<thead>
<tr>
<th>Prompts with Known Translation Lengths</th>
<th>Translation Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company</td>
<td>.7”</td>
</tr>
<tr>
<td>Item</td>
<td>.5”</td>
</tr>
<tr>
<td>Line</td>
<td>.5”</td>
</tr>
<tr>
<td>Original</td>
<td>.8”</td>
</tr>
<tr>
<td>Probability</td>
<td>.11”</td>
</tr>
<tr>
<td>Rank</td>
<td>.5”</td>
</tr>
<tr>
<td>Revaluation</td>
<td>.11”</td>
</tr>
<tr>
<td>Task</td>
<td>.6”</td>
</tr>
<tr>
<td>Time</td>
<td>.5”</td>
</tr>
</tbody>
</table>
## Acronyms

<table>
<thead>
<tr>
<th>Approved Acronyms</th>
<th>Min. Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATO: Assemble To Order</td>
<td>.7”</td>
</tr>
<tr>
<td>ATP: Available to Promise</td>
<td>.4”</td>
</tr>
<tr>
<td>CIP: Construction–In–Process</td>
<td>.4”</td>
</tr>
<tr>
<td>COGS: Cost of Goods Sold</td>
<td>.5”</td>
</tr>
<tr>
<td>ECO: Engineering Change Order</td>
<td>.5”</td>
</tr>
<tr>
<td>FOB: Free On Board</td>
<td>.9”</td>
</tr>
<tr>
<td>GSA: General Services Admin</td>
<td>.4”</td>
</tr>
<tr>
<td>ITC: Investment Tax Credit</td>
<td>.4”</td>
</tr>
<tr>
<td>RMA: Return Material Auth</td>
<td>.5”</td>
</tr>
<tr>
<td>RTV: Return to Supplier</td>
<td>.4”</td>
</tr>
<tr>
<td>YTD: Year–to–Date</td>
<td>.4”</td>
</tr>
</tbody>
</table>
## Index

### ISO–Numbers

| 58012, 5 – 5 | 71021, 1 – 22 |
| 58140, 6 – 8 | 71105, 1 – 20 |
| 58141, 6 – 8 | 72001, 2 – 2 |
| 58147, 3 – 10 | 72002, 2 – 2 |
| 58149, 3 – 10 | 72003, 2 – 6 |
| 58150, 3 – 10 | 72004, 2 – 6 |
| 58152, 3 – 10 | 72005, 2 – 12 |
| 58168, 6 – 17 | 72006, 2 – 13 |
| 58500, 3 – 7 | 73001, 3 – 2 |
| 58501, 3 – 7 | 73002, 3 – 3 |
| 58503, 3 – 8 | 73003, 3 – 3 |
| 58505, 3 – 8 | 73004, 3 – 3 |
| 58527, 6 – 8 | 73005, 3 – 4 |
| 60053, 3 – 16 | 73006, 3 – 4 |
| 60055, 6 – 2 | 73007, 3 – 4 |
| 70011, 4 – 4 | 73008, 3 – 4 |
| 71002, 1 – 3 | 73009, 3 – 4 |
| 71005, 1 – 18 | 73010, 3 – 4 |
| 71019, 1 – 22 | 73011, 3 – 4 |
| 71020, 1 – 22 | 73012, 3 – 5 |
| 73013, 3 – 5 |
73014, 3 – 5
73015, 3 – 5
73016, 3 – 5
73017, 3 – 5
73018, 3 – 5
73019, 3 – 6
73020, 3 – 6
73021, 3 – 6
73022, 3 – 6
73023, 3 – 7
73024, 3 – 7
73027, 3 – 8
73028, 3 – 8
73030, 3 – 9
73031, 3 – 9
73036, 3 – 10
73037, 3 – 10
73038, 3 – 10
73039, 3 – 10
73040, 3 – 10
73041, 3 – 11
73042, 3 – 11
73043, 3 – 11
73044, 3 – 11
73045, 3 – 11
73046, 3 – 12
73047, 3 – 12
73048, 3 – 12
73049, 3 – 13
73050, 3 – 13
73052, 3 – 14
73053, 3 – 14
73054, 3 – 14
73055, 3 – 14
73057, 3 – 16
73058, 3 – 16
73059, 3 – 16
73060, 3 – 17
73061, 3 – 17
73062, 3 – 19
73063, 3 – 19
73064, 3 – 19
73065, 3 – 19
73066, 3 – 20
73067, 3 – 20
73068, 3 – 22
73069, 3 – 22
73070, 3 – 23
73071, 3 – 23
73074, 3 – 26
73075, 3 – 27
73076, 3 – 27
73077, 3 – 27
73078, 3 – 28
73079, 3 – 28
73080, 3 – 29
73081, 3 – 31
73082, 3 – 31
73083, 3 – 31
73084, 3 – 32
73085, 3 – 32
73086, 3 – 32
73087, 3 – 32
73088, 3 – 32
73089, 3 – 32
73090, 3 – 33
73091, 3 – 33
<table>
<thead>
<tr>
<th>Index</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>73092</td>
<td>3 – 33</td>
</tr>
<tr>
<td>73093</td>
<td>3 – 34</td>
</tr>
<tr>
<td>73094</td>
<td>3 – 34</td>
</tr>
<tr>
<td>73095</td>
<td>3 – 38</td>
</tr>
<tr>
<td>73096</td>
<td>3 – 38</td>
</tr>
<tr>
<td>73097</td>
<td>3 – 38</td>
</tr>
<tr>
<td>73098</td>
<td>3 – 38</td>
</tr>
<tr>
<td>73099</td>
<td>3 – 39</td>
</tr>
<tr>
<td>73100</td>
<td>3 – 39</td>
</tr>
<tr>
<td>73101</td>
<td>3 – 39</td>
</tr>
<tr>
<td>73102</td>
<td>3 – 39</td>
</tr>
<tr>
<td>73103</td>
<td>3 – 39</td>
</tr>
<tr>
<td>73104</td>
<td>3 – 41</td>
</tr>
<tr>
<td>73105</td>
<td>3 – 41</td>
</tr>
<tr>
<td>73106</td>
<td>3 – 41</td>
</tr>
<tr>
<td>73107</td>
<td>3 – 42</td>
</tr>
<tr>
<td>73108</td>
<td>3 – 42</td>
</tr>
<tr>
<td>73109</td>
<td>3 – 42</td>
</tr>
<tr>
<td>73110</td>
<td>3 – 42</td>
</tr>
<tr>
<td>73111</td>
<td>3 – 44</td>
</tr>
<tr>
<td>73112</td>
<td>3 – 45</td>
</tr>
<tr>
<td>73113</td>
<td>3 – 45</td>
</tr>
<tr>
<td>73114</td>
<td>3 – 45</td>
</tr>
<tr>
<td>73115</td>
<td>3 – 45</td>
</tr>
<tr>
<td>73116</td>
<td>3 – 45</td>
</tr>
<tr>
<td>73117</td>
<td>3 – 46</td>
</tr>
<tr>
<td>73118</td>
<td>3 – 47</td>
</tr>
<tr>
<td>73119</td>
<td>3 – 49</td>
</tr>
<tr>
<td>73120</td>
<td>3 – 49</td>
</tr>
<tr>
<td>73121</td>
<td>3 – 50</td>
</tr>
<tr>
<td>73122</td>
<td>3 – 51</td>
</tr>
<tr>
<td>73123</td>
<td>3 – 51</td>
</tr>
<tr>
<td>73124</td>
<td>3 – 51</td>
</tr>
<tr>
<td>73125</td>
<td>3 – 51</td>
</tr>
<tr>
<td>73126</td>
<td>3 – 51</td>
</tr>
<tr>
<td>73127</td>
<td>3 – 51</td>
</tr>
<tr>
<td>73128</td>
<td>3 – 52</td>
</tr>
<tr>
<td>73129</td>
<td>3 – 52</td>
</tr>
<tr>
<td>73130</td>
<td>3 – 53</td>
</tr>
<tr>
<td>73131</td>
<td>3 – 53</td>
</tr>
<tr>
<td>73132</td>
<td>3 – 53</td>
</tr>
<tr>
<td>73133</td>
<td>3 – 53</td>
</tr>
<tr>
<td>73134</td>
<td>3 – 53</td>
</tr>
<tr>
<td>73135</td>
<td>3 – 53</td>
</tr>
<tr>
<td>73136</td>
<td>3 – 54</td>
</tr>
<tr>
<td>73137</td>
<td>3 – 54</td>
</tr>
<tr>
<td>73149</td>
<td>3 – 13</td>
</tr>
<tr>
<td>73174</td>
<td>3 – 26</td>
</tr>
<tr>
<td>73175</td>
<td>3 – 26</td>
</tr>
<tr>
<td>73176</td>
<td>3 – 27</td>
</tr>
<tr>
<td>73178</td>
<td>3 – 28</td>
</tr>
<tr>
<td>73180</td>
<td>3 – 30</td>
</tr>
<tr>
<td>73203</td>
<td>3 – 40</td>
</tr>
<tr>
<td>73218</td>
<td>3 – 48</td>
</tr>
<tr>
<td>73220</td>
<td>3 – 50</td>
</tr>
<tr>
<td>73244</td>
<td>3 – 51</td>
</tr>
<tr>
<td>73250</td>
<td>3 – 49</td>
</tr>
<tr>
<td>73267</td>
<td>3 – 21</td>
</tr>
<tr>
<td>73280</td>
<td>3 – 31</td>
</tr>
<tr>
<td>73318</td>
<td>3 – 48</td>
</tr>
<tr>
<td>73333</td>
<td>3 – 52</td>
</tr>
<tr>
<td>73334</td>
<td>3 – 52</td>
</tr>
<tr>
<td>73350</td>
<td>3 – 46</td>
</tr>
<tr>
<td>73444</td>
<td>3 – 53</td>
</tr>
<tr>
<td>73546</td>
<td>3 – 12</td>
</tr>
<tr>
<td>73548</td>
<td>3 – 13</td>
</tr>
<tr>
<td>73550</td>
<td>3 – 39</td>
</tr>
</tbody>
</table>
73551, 3 – 39
73552, 3 – 40
73553, 3 – 40
73554, 3 – 14
73555, 3 – 14
73556, 3 – 14
73557, 3 – 14
73558, 3 – 16
73559, 3 – 40
73561, 3 – 17
73565, 3 – 19
73567, 3 – 20
73571, 3 – 24
73572, 3 – 24
73580, 3 – 29
73590, 3 – 33
73592, 3 – 33
73645, 3 – 11
73650, 3 – 43
73651, 3 – 43
73672, 3 – 25
73673, 3 – 25
73674, 3 – 25
73675, 3 – 25
73676, 3 – 25
73690, 3 – 33
73750, 3 – 45
73751, 3 – 45
73773, 3 – 25
73774, 3 – 25
73818, 3 – 47
73819, 3 – 48
73874, 3 – 26
73875, 3 – 26
73876, 3 – 26
73900, 3 – 3
73930, 3 – 9
74001, 4 – 4
74002, 4 – 4
74003, 4 – 4
74004, 4 – 5
74005, 4 – 5
74006, 4 – 5
74007, 4 – 5
74008, 4 – 5
74009, 4 – 6
74010, 4 – 7
74011, 4 – 7
74012, 4 – 8
74013, 4 – 8
74014, 4 – 8
74015, 4 – 8
74016, 4 – 8
74017, 4 – 8
74018, 4 – 9
74019, 4 – 9
74020, 4 – 9
74021, 4 – 10
74022, 4 – 10
74023, 4 – 11
74024, 4 – 11
74025, 4 – 11
74026, 4 – 12
74027, 4 – 12
74028, 4 – 12
74029, 4 – 12
74030, 4 – 12

Index – 4   Oracle Applications User Interface Standards
| 74031, 4 – 15 | 74134, 4 – 14 |
| 74032, 4 – 15 | 74232, 4 – 16 |
| 74033, 4 – 16 | 74234, 4 – 16 |
| 74034, 4 – 16 | 74235, 4 – 16 |
| 74035, 4 – 18 | 74238, 4 – 19 |
| 74036, 4 – 18 | 74239, 4 – 21 |
| 74037, 4 – 18 | 74247, 4 – 24 |
| 74038, 4 – 19 | 74248, 4 – 24 |
| 74039, 4 – 21 | 74249, 4 – 24 |
| 74040, 4 – 21 | 74250, 4 – 24 |
| 74041, 4 – 21 | 74252, 4 – 28 |
| 74042, 4 – 21 | 74348, 4 – 25 |
| 74043, 4 – 22 | 74349, 4 – 25 |
| 74044, 4 – 22 | 74350, 4 – 26 |
| 74045, 4 – 23 | 74451, 4 – 24 |
| 74046, 4 – 23 | 74452, 4 – 24 |
| 74047, 4 – 23 | 74523, 4 – 11 |
| 74048, 4 – 24 | 74524, 4 – 11 |
| 74049, 4 – 24 | 75001, 5 – 2 |
| 74050, 4 – 27 | 75002, 5 – 3 |
| 74051, 4 – 27 | 75003, 5 – 3 |
| 74052, 4 – 28 | 75004, 5 – 5 |
| 74053, 4 – 28 | 75006, 5 – 5 |
| 74054, 4 – 29 | 75007, 5 – 5 |
| 74055, 4 – 30 | 75008, 5 – 5 |
| 74056, 4 – 30 | 75009, 5 – 7 |
| 74057, 4 – 30 | 75010, 5 – 11 |
| 74058, 4 – 30 | 75011, 5 – 11 |
| 74059, 4 – 30 | 75012, 5 – 13 |
| 74130, 4 – 13 | 75013, 5 – 14 |
| 74131, 4 – 14 | 75014, 5 – 14 |
| 74132, 4 – 14 | 75015, 5 – 14 |
| 74133, 4 – 14 | 75016, 5 – 16 |
| 75017, 5 – 16 |
75018, 5 – 19
75019, 5 – 20
75020, 5 – 21
75021, 5 – 22
75022, 5 – 3
75023, 5 – 3
75024, 5 – 6
75025, 5 – 8
75026, 5 – 8
75027, 5 – 9
75028, 5 – 12
75029, 5 – 13
75030, 5 – 14
75031, 5 – 14
75032, 5 – 16
75033, 5 – 17
75034, 5 – 19
75035, 5 – 20
75036, 5 – 23
75109, 5 – 9
76001, 6 – 2
76002, 6 – 2
76003, 6 – 2
76004, 6 – 2
76005, 6 – 5
76006, 6 – 5
76007, 6 – 6
76008, 6 – 6
76009, 6 – 6
76010, 6 – 6
76011, 6 – 6
76012, 6 – 6
76013, 6 – 6
76014, 6 – 7
76015, 6 – 7
76016, 6 – 7
76017, 6 – 7
76018, 6 – 7
76019, 6 – 7
76020, 6 – 7
76021, 6 – 8
76022, 6 – 8
76023, 6 – 8
76024, 6 – 8
76025, 6 – 9
76026, 6 – 9
76027, 6 – 9
76028, 6 – 9
76029, 6 – 9
76030, 6 – 15
76031, 6 – 15
76032, 6 – 16
76033, 6 – 16
76034, 6 – 18
76035, 6 – 18
76036, 6 – 19
76037, 6 – 19
76038, 6 – 23
76039, 6 – 23
76040, 6 – 23
76041, 6 – 30
76042, 6 – 5
76043, 6 – 5
76044, 6 – 5
76045, 6 – 23
76046, 6 – 27
76047, 6 – 3
76048, 6 – 4
76049, 6 – 5
Letter Index

A
abbreviations, B – 2
Access keys, 5 – 16
Action menu, 2 – 6
Alternative blocks, 3 – 40
Alternative regions, 3 – 51
   characteristics, 3 – 53
   poplist, 3 – 52
   use of, 1 – 12
APP CORE, 2 – 5
APP DAY PK, 2 – 5
APP FLD, 2 – 5
Attributes
   display-only fields, 1 – 22
   fonts, 2 – 2
   input fields, 1 – 22
   LOV fields, 1 – 22
   non-applicable items, 1 – 22
   visual, 2 – 2

B
Blocks, 3 – 22
   alternative, 3 – 40
   combination, 3 – 31
   context, 3 – 24
   definition, 1 – 6
   dialog, 3 – 25
   Find, 3 – 39
   folder, 3 – 35
   multi-record, 3 – 28
   navigation, 1 – 23
   prompts, 5 – 7, 5 – 9
   relation to windows, 1 – 15
   single-record, 3 – 26
   title, 3 – 22
boilerplate, 5 – 1

Buttons, 4 – 21
   labeling conventions, 5 – 13
   usage, 1 – 22

C
Calculated fields, 6 – 27
Calendar, 2 – 15
Call Form, 6 – 14
Canvasses, 3 – 19
   content, 3 – 19
   definition, 1 – 6
   stacked, 3 – 19
Check boxes, 4 – 18
   definition, 1 – 7
Color palette, 2 – 2
Combination blocks, 3 – 31
   appearance, 3 – 32
   behavior, 3 – 32
   buttons, 3 – 33
   definition and usage, 1 – 14
   gateways, 3 – 34
   windows, 3 – 32
Committing changes, 6 – 3
Console, 2 – 14
Content canvasses, 3 – 19
Context blocks, 3 – 24
Context, indication of, 1 – 19
Currency, 6 – 23
Current record indicator, 1 – 19, 3 – 28

D
DATE, 4 – 4
DATETIME, 4 – 4
Descriptive flexfields, 4 – 27
Detail format, 3 – 31
Dialog blocks, 3 – 25
    find, 3 – 11
Disabled functions, 6 – 15
Display items, 4 – 9
    definition, 1 – 8
Drilldown, 1 – 23
Dynamic layouts, 1 – 16
    folders, 1 – 17
    profiles, 1 – 16
Dynamic window titles, 3 – 8

Folder menu, 2 – 9
Folders, 1 – 17
    objects, 2 – 16
Fonts, 2 – 2
Forms, 3 – 2
Function invocation, 1 – 22
Functions, disabled, 6 – 15

Gateways, 3 – 34
Go menu, 2 – 8

Headers, spanning, 3 – 49
Help
    About Oracle Applications, 6 – 33
    About This Record, 6 – 32
Help menu, 2 – 10
Highlighting, 6 – 27
Hint messages, 5 – 23
Hybrid formats, 1 – 14

Images, 1 – 8
Information
    Oracle Applications, 6 – 33
    record, 6 – 32
Information messages, 5 – 22
Inquiry forms, 6 – 13
Items
    display, 4 – 9
    text, 4 – 3

Key flexfields, 4 – 30
Keys, 5 – 16

L

Layout, general rules, 1 – 18
Libraries, 2 – 5
Linked forms, 6 – 13
List of Values (LOV), 4 – 23
definition, 1 – 7
Lists, 1 – 8
pops lists, 4 – 11
T–lists, 4 – 14
Long–running processes, 6 – 30

M

Master-detail relationships, 3 – 40
coordination, 3 – 41
general behavior, 1 – 7
other behaviors, 3 – 43
titles, 3 – 40
Menus, 1 – 5, 2 – 6
Action, 2 – 6
customizing, 1 – 24
Edit, 2 – 7
Folder, 2 – 9
Go, 2 – 8
Help, 2 – 10
Properties submenu, 2 – 11 to 2 – 18
Query, 2 – 8
Special, 2 – 10
Tools submenu, 2 – 11 to 2 – 18
Messages, 1 – 9, 5 – 24
Mnemonics, 5 – 16
Modal windows, 3 – 15
See also Dialog blocks
closing, 3 – 17
menus, 3 – 16
position, 3 – 16
size, 3 – 17
title, 3 – 16
Models, validation, 6 – 2
Modules, 3 – 2
display attributes, 3 – 2
functional attributes, 3 – 3
Movement, 1 – 22, 6 – 5
Multi-record blocks, 3 – 28
clearing, 3 – 31
current record indicator, 3 – 28
layout, 3 – 30
prompts, 5 – 9
querying, 3 – 31
scrollbar, 3 – 28
selecting multiple records, 6 – 26
Multi-record formats, 1 – 13
prompts, 5 – 9
Mutually calculated fields, 6 – 27

N

Navigate window, 6 – 10
Navigation, 1 – 22, 6 – 5
alternate region, 6 – 8
block, 6 – 7
control elements, 6 – 6
keyboard, 6 – 5
mouse, 6 – 6
Navigator, 6 – 10
Non-modal windows, 3 – 6
closing, 3 – 10
menus, 3 – 11
position, 3 – 6
resizing, 3 – 10
title, 3 – 7
numeric, Text Items, 4 – 4

O

Open Form, 6 – 13
Option groups, 1 – 8, 4 – 15
Oracle Applications, information, 6 – 33
Oracle tools, accessing, 2 – 11 to 2 – 12
Order by, 6 – 31
Ordering, 6 – 31
Overflow regions, 3 – 47 to 3 – 49
  cosmetics, 3 – 48
  navigation, 3 – 48
  position, 3 – 47
  usage, 1 – 14

P
Palette, 2 – 2
Percent fields, 5 – 14
Poplists, 4 – 11
Portability, 1 – 3
Position
  modal windows, 3 – 16
  non-modal windows, 3 – 6
  windows, 3 – 5
Printing, 6 – 28
Processes, long-running, 6 – 30
Profiles, 1 – 16
Prompts
  multi-record blocks, 5 – 9
  properties, 5 – 5
  single-record blocks, 5 – 7
  use of data as, 5 – 18
Property classes, 2 – 4
Pulldown menus, 2 – 6

Q
Query menu, 2 – 8
Querying, 1 – 20
Question text, 5 – 21

R
Radio groups. See Option groups
Ranges, 6 – 20
Record indicator, 1 – 19, 3 – 28
Records
  information, 6 – 32
  querying, 1 – 20
  selecting, 6 – 26
Regions, 1 – 6, 3 – 44
  alternative, 1 – 12, 3 – 51
  combination blocks, 1 – 14
  cosmetics, 3 – 45
  definition and usage, 1 – 12
  hybrid formats, 1 – 14
  multi-record blocks, 3 – 49
  multi-record formats, 1 – 13
  navigation, 3 – 45
  overflow, 1 – 14, 3 – 47
  scrolling, 3 – 50
  single-record formats, 1 – 13, 3 – 46
  title, 3 – 44
Requirements analysis, 1 – 2
  portability, 1 – 5
Resizing modal windows, 3 – 17
Resizing windows, 3 – 10
Row LOV, windows, 3 – 14
Running totals, 6 – 19

S
Saving changes, 6 – 3
Scroll bars, 1 – 8
Scrolling regions, 3 – 50
Selecting multiple records, 6 – 26
Semi-modal, windows, 3 – 17
Single-record blocks, 3 – 26
  clearing, 3 – 27
  querying, 3 – 27
Single-record formats, 3 – 46
  prompts, 5 – 7
usage, 1 – 13
Spanning headers, 3 – 49
Special menu, 1 – 24, 2 – 10
Stacked canvases, 3 – 19
Summary/detail formats, 3 – 31

T
T–lists, 4 – 14
TAB, 1 – 24
Task flow analysis, 1 – 2
Text
   buttons, 5 – 13
   general properties, 5 – 2
   percent fields, 5 – 14
   prompts, 5 – 5
   titles, 5 – 5
   translation requirements, 5 – 2
Text items, 4 – 3
TIME, 4 – 4
Title, modal windows, 3 – 16
Titles, 5 – 5
   block, 3 – 22
   non–modal windows, 3 – 7
   region, 3 – 44, 3 – 49
Toolbar, 1 – 6, 2 – 12
   customizing, 1 – 24
Tools, Oracle, 2 – 11 to 2 – 12
Translation, 1 – 4, 5 – 2

V
Validation, 6 – 2
Validation models, 6 – 2
Visual attributes, 2 – 2

W
Warning messages, 5 – 20
Window size, limits, 1 – 3
Windows, 3 – 4
   closing, 3 – 10, 3 – 17
   definition, 1 – 5
   find, 3 – 11
   look and feel, 3 – 4
   menus, 3 – 11
   modal, 3 – 15
   non–modal, 3 – 6
   position, 3 – 5
   relation to blocks, 1 – 15
   resizing, 3 – 10
   Row LOV, 3 – 14
   semi–modal, 3 – 17
   size, 3 – 5
   titles, 3 – 7
Workbench, 3 – 34

Z
Zoom, 6 – 14
Reader’s Comment Form

Oracle Applications User Interface Standards
A58193–01

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

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

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

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

Please send your comments to:
Oracle Applications Documentation Manager
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065 USA
Phone: (650) 506–7000 Fax: (650) 506–7200

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

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

Thank you for helping us improve our documentation.