

Holos

Dialog Design Standards Guide for Holos

Overview

This document provides guidelines for the design of GUI dialogs and wizards in Holos. This document applies to Holos version 6 and later.

Contents

INTRODUCTION	4
<i>What makes a well-designed dialog?</i>	4
DESIGN SUGGESTIONS	4
<i>Be consistent.....</i>	4
<i>Use plain language.....</i>	5
<i>Aesthetics are important.....</i>	5
<i>Ease of use.....</i>	5
ABOUT VISUAL DESIGN	5
<i>Composition & Organization</i>	5
<i>Hierarchy of Information</i>	5
<i>Focus and Emphasis.....</i>	6
<i>Relationship of Elements.....</i>	6
<i>Readability & Flow</i>	6
<i>Unity & Integration.....</i>	6
<i>Color</i>	7
<i>Fonts.....</i>	7
<i>Dimensionality</i>	8
LAYOUT	8
<i>Capitalization.....</i>	8
<i>Grouping & Spacing</i>	9
<i>Alignment</i>	9
<i>Command Button Placement.....</i>	9
<i>Tabbing Order.....</i>	9
POOR GUI DESIGN EXAMPLE	10
<i>What is wrong with this design?.....</i>	10
<i>Too many functions at the top level</i>	10
<i>Too many buttons.....</i>	10
<i>Poor navigational assistance.....</i>	10
<i>Inappropriate use of controls.....</i>	11
<i>Lack of symmetry</i>	11

IMPROVED GUI DESIGN EXAMPLE.....11
Main screen..... 11
Second screen..... 12

HOW WAS THE DESIGN IMPROVED?12
Order out of chaos..... 12
Reordering input fields..... 12
Improved controls 12

DIALOGS13
Modal vs. Modeless Dialogs 13
When to Use Dialogs or Windows..... 14

BASIC STYLE GUIDE FOR HOLOS DIALOGS14
Dimming controls..... 14
Positioning a dialog 14
Dialogs that expand and contract 14
Buttons on dialogs..... 14
Default buttons 15
Placing of buttons..... 15
Text fonts 15

HANDY HINTS WHEN DESIGNING DIALOGS.....16
Making controls the same size..... 16
Aligning controls on the dialog..... 16
Using the mouse buttons 17
Sizing drop-down lists 17
Locking the position of controls..... 17

GUIDELINES FOR USING CONTROLS18
Label..... 18
Text (Push) Button..... 18
Check Box..... 18
Radio (Option) Button..... 18
Possible Button Alternative..... 19
Edit (Text) Box 19
List Box..... 19
List Edit (Combo) Box..... 19
Drop-down list 19
Group Box..... 20
Format Button & Format Picture 20

CREATING STANDARD CONTROLS IN THE HOLOS DIALOG DESIGNER.....20

HOLOS CUSTOM CONTROLS21
Spin Button 21
Double Slider 21
Slider 21
Progress Bar 21
Edit Spin..... 21
Edit Button 21

CREATING CUSTOM CONTROLS IN THE HOLOS DIALOG DESIGNER22
Guideline for the Number of Controls to Display per Dialog..... 22

HOLOS DATA TAB23
Twist 23

EXAMPLE OF CONTROLS IN USE IN HOLOS REPORT DESIGNER24

GUIDELINES FOR DESIGNING WIZARDS25

Guidelines for Writing Text for Wizard Pages 27

PROGRAMMING STANDARDS**27**

Position of objects in the Application Manager 27

File naming conventions 27

Single file as opposed to multiple files 27

Naming conventions for records, compound records and functions 28

Dialog control naming conventions 28

Other Naming Suggestions..... 28

HOLOS Dialog Object Prefixes 29

MOUSE & KEYBOARD INTERFACE**30**

List of Reserved Words..... 30

Reserved Icons 30

FINDING MORE INFORMATION / REFERENCES.....**31**

CONTACTING CRYSTAL DECISIONS FOR TECHNICAL SUPPORT**31**

Introduction

The purpose of this document is to provide guidelines for the design of GUI dialogs and wizards.

Visual design is a critical aspect of the interface for an application. Visual attributes provide valuable impressions and communicate important cues for the user interaction required. Without a structure and balance of visual elements, there is a lack of order and meaning. The way you organize a window can make a difference between a design that communicates a message and one that leaves a user feeling puzzled or overwhelmed.

What makes a well-designed dialog?

It is important that dialogs are clearly laid out, present the user with the right information and ask the right questions. Bear in mind that every visual element that appears on the screen potentially competes for the user's attention.

Ideally, a dialog should be simple to use, and still accomplish the required task.

Several aspects therefore need careful consideration:

- Placement of buttons
- Cursor movement
- Consistency between dialogs in an application

It is the designer's job to make the dialog as simple as possible for the user to use!

Design Suggestions

There are standards that are recommended when designing a dialog. Both Windows and Macintosh systems offer design conventions for dialogs and controls, to ensure a consistent 'look and feel' across applications. This document draws on the recommendations offered in "The Windows Interface Guidelines for Software Design".

Be consistent

Consistency is important through all aspects of the interface, including names of commands, visual presentation of information, and operational behavior.

Users are accustomed to seeing OK, Cancel and Help buttons in order, vertically placed on the right or horizontally placed on the bottom of the dialog. The Return key should action the OK button and the Escape key should action the Cancel button.

Consistency allows users to transfer existing knowledge to new tasks, making the interface feel more familiar and predictable. Bear in mind that each "new

and exciting” experience you provide in the software can become an anxiety-inducing experience or another call to the help desk!

Use plain language

Use clear, unambiguous language that is free of technical jargon.

Aesthetics are important

The aesthetics of a dialog are the first impression made upon most user. Keep the layout of dialog clear and uncluttered. All elements should be logically related and placed.

Ease of use

Although it is harder to do, the dialog should be designed so that it cannot be used incorrectly. Consider what would happen if the user inadvertently pressed the Enter key, would the control that is in focus by default, result in an update to the database that is awkward to reverse?

About Visual Design

Composition & Organization

We organize what we read and how we think about information by grouping it spatially. We read a screen in the same way we read other forms of information i.e. left to right, top to bottom. The eye is always attracted to the color elements before black and white, to isolated elements before grouped elements and to graphics before text.

Hierarchy of Information

The principle of hierarchy of information addresses the placement of information based on its relative importance to other visual elements. The outcome of this ordering determines what a user sees first and what a user feels encouraged to do first.

Ask yourself:

- What information is most important to the user?
- What does a user want or need to do first, second...?
- What should a user see on the screen first, second...?

What a user sees first should match the user’s priorities when possible but can be affected by the elements you want to emphasize for procedural correctness.

Ensure that features used frequently are readily available. Avoid the temptation to put everything on the first screen or to load the toolbar with rarely used buttons.

Focus and Emphasis

People in western cultures look at the upper left corner of the screen or window for the most important information. Choose the element that must be prominent, isolate it and emphasize it by placing it in top left corner.

Relationship of Elements

The placement of visual element can help communicate a specific relationship of the elements of which it is a part. For example if a button in a dialog affects the content of a list box, there should be a spatial relationship between the two elements. This helps the user to clearly and quickly make the connection just by looking at the placement.

Readability & Flow

Communicate ideas simply and directly. Consider the following:

- Could this idea or concept be presented in a simpler manner?
- Can the user easily step through the interface as designed?
- Do all the elements have a reason for being there?

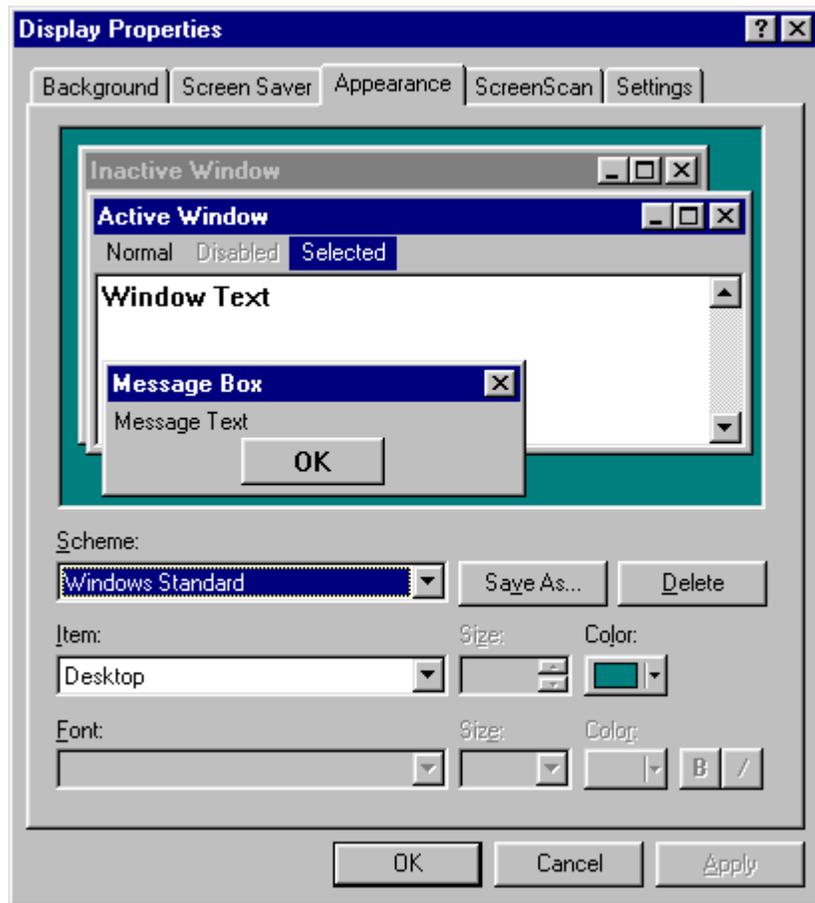
Unity & Integration

When an application's interface is visually unified with the general interface of Windows, the user finds it easier to use because it offers a consistent and "predictable" work environment. (This is a quote from Microsoft's reference guide!!!) The Holos Dialog Designer (HDD) appears to have adopted the Microsoft Windows standards with respect to color, font, dimensionality, border style & visual states for controls.

Consider the following:

- How do all of the different parts of the screen work together visually?
- How does the visual design of the application relate to the system's interface or other applications with which it is being used?

Color



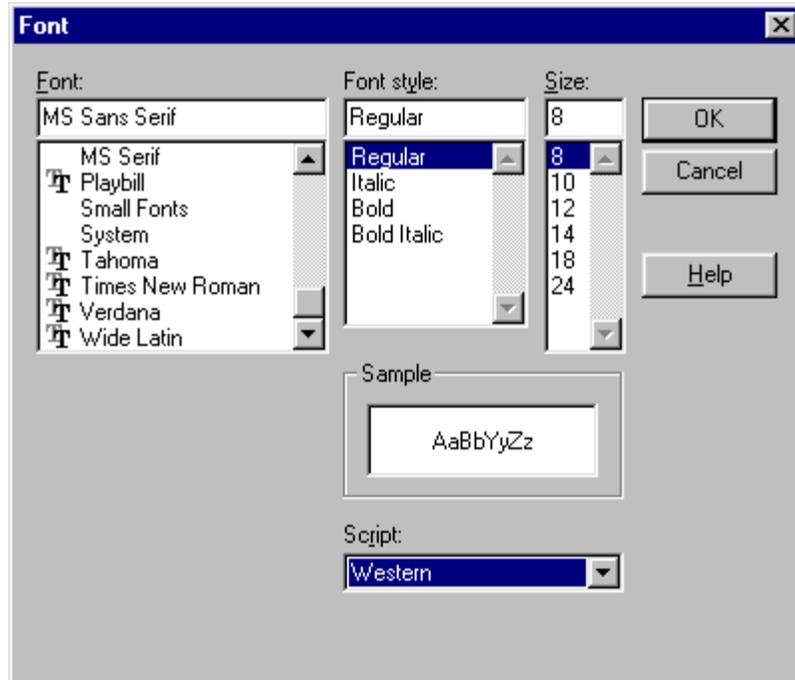
The color of the window is determined by the setting of Properties / Appearance / Scheme, as shown below. Ensure that you consider the user's company standards for system window settings.

There is no option to change colors within the Holos Dialog Designer (HDD).

Fonts

The Holos Dialog Designer (HDD) provides the following interface from which to choose the font type, style, size and script, which are preset to the Windows default of MS Sans Serif for 8 point.

Holos sets the font at dialog level; therefore only one font can be used for the dialog. For example: it cannot be varied per control within the dialog!



Dimensionality

The controls in the Holos Dialog Designer (HDD) use the standard Microsoft Windows perspective, highlighting, shading and border styles to provide a 3-dimensional appearance, which differentiate them from other types of information. Also the standard states of a control, for instance a ticked check box, a pressed command button and a disabled or grayed control for unavailable options have been adopted.

Layout

Sizing, spacing and placement of information are critical in creating a visually consistent and predictable environment. As previously mentioned, for western countries, layout conventions should follow left to right, top to bottom, with most important information located in the upper left corner.

Capitalization

When displaying text in menus, command buttons and tabs use conventional book title capitalization i.e. capitalize the first letter in each word unless it is an article or preposition not occurring at the beginning or end of the name, or the word's conventional usage is not capitalized. Field labels such as those used for option buttons, check boxes, text boxes, group boxes and page tabs should use sentence-style capitalization i.e. capitalize only the first letter of the initial word and any words that are normally capitalized.

Grouping & Spacing

Group related components together. You can use group box controls or spacing. Maintain a consistent margin from the edge of the window and consistent spacing between controls. It is not a common convention to visually group objects with color as this may result in undesirable effects when a user changes color schemes.

Alignment

When information is positioned vertically, align fields by their left (in western countries). This usually makes it easier for the user to scan the information. Text labels are usually left aligned and placed above or to the left of the areas to which they apply. When placing text labels to the left of the text box controls, align the height of the text with text displayed in the text box.

Command Button Placement

Stack the main command buttons vertically in the upper right corner or in a row along the bottom. If there is a default button it is typically the first button of the set. Place OK and Cancel buttons next to each other. The last button is a Help button (if supported). If there is no OK button, but other command buttons, it is best to place the Cancel button at the end of a set of action buttons, but before a Help button. If a particular command button applies only to a particular field, group it with that field in the group rather than with the other buttons, which apply to the entire window.

Tabbing Order

The tab order is the order in which the focus changes from control to control in a finished dialog when you press the TAB key. To enable the Tab key to shift focus, the Tab Stop property is set to True in the detail sheet for each control. If the Tab Stop is False, the control cannot be tabbed to. Consideration must be given to the logical progression from control to control in the dialog and whether the user needs to interact with it.

The tab order in the HDS file code is the order in which the controls were created. If you wish to change the order, you must rearrange the order of the controls in the HDS file using cut and paste.

Poor GUI Design Example

What is wrong with this design?

The screenshot shows a 'Patient Account' dialog box with the following fields and buttons:

- Company:** 002
- Invoice:** AK-3356
- S.S.N.:** 044-52-8995
- Trip Date:** 6/12/92
- Operator:** T. Michaels
- Date:** 6/15/92
- Last Name:** Carlson
- First:** Melissa
- Medicare:** A 74493021
- Medi-Cal:**
- (A) Patient's Address:** 71194 El Camino Real, San Jose, CA 92344, Home: (714) 688-5433, Work:
- (B) Birthday:** 3/5/34
- Sex:** F
- Sig/File:**
- POE:**
- (C) Charge To:** SHP Senior Plan, P.O. Box 3382, San Diego, CA 99445
- (B) Physician:** Marshall, 4550 Stetson Dr., (409) 448-3922
- From Location:** 445 Elm St., Sonoma
- To Location:** General Hospital, Sonoma, Admit: C
- (D) Diagnosis:** 8054 Syncope Episode, 9000 Head injury
- Billing:** 000 Private, Charge 345.90, Paid 290.00, Adjust -50.00, Balance -5.90
- (E) Employer:**
- (R) Reason...:** Transport: 1, Ambulance: 5, Update button
- Caller:** Family, (405) 934-5596
- Buttons on the right:** Notes, Payments, Insurance, Letters, Billing, Trans, Clear
- Bottom row:** Rcvd 03:45, Disp 03:45, Arrive 03:53, Comp 04:13, R.T. button, (U) Auth: 9333, Inst Billing ID: []

Too many functions at the top level

The users requested that the new application provided all information at their fingertips. This resulted in the screen being used for both customer maintenance and ambulance dispatching. If you enter extensive customer information and then press the Update button, the record is updated. However, if you enter minimal customer information, such as Social Security number, diagnosis, from-location, and to-location, then press the Trans button, an ambulance will be dispatched!!! Updating and dispatching functions need to be on separate dialogs.

Too many buttons

The buttons along the right should be on the application's parent window, possibly in a toolbar.

Poor navigational assistance

GUI controls should be positioned according to frequency of use. The most important field in the upper left; the least important in the lower right. Its hard to imagine how the company and invoice number could be the most important fields when dispatching an ambulance.

Inappropriate use of controls

This many group boxes with text labels in these positions make the screen appear convoluted and makes it difficult to distinguish the data. Editable fields should be identified with a box around them as it is intuitively obvious which fields can be changed.

Lack of symmetry

Just lining up fields, group boxes and buttons will make this GUI much easier to use. Our brains like order, not chaos.

Improved GUI Design Example

Main screen

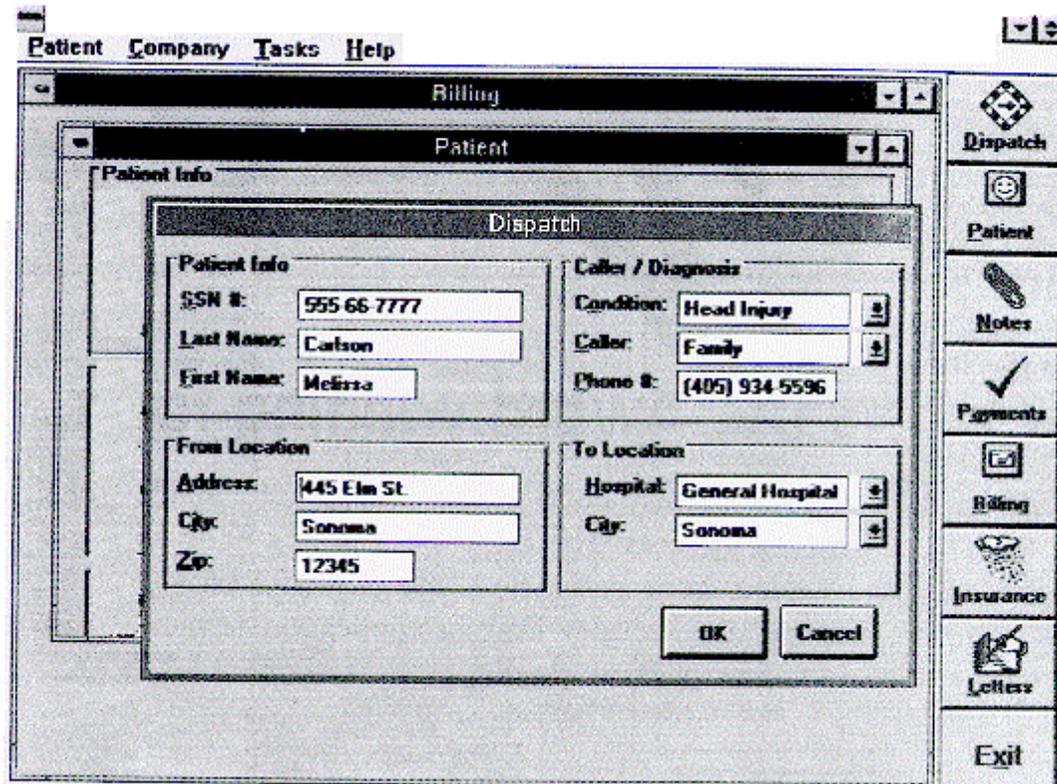
The screenshot displays a software window titled "Billing" with a menu bar containing "Patient", "Company", "Tasks", and "Help". The main content area is divided into several sections:

- Patient Info:** Includes fields for SSN # (955-66-7777), Invoice # (1234567), Last Name (Carlson), Medicare # (1122334455), First Name (Melissa), Date of Birth (12/25/50), Company (002), and Sex (Female selected, Male unselected).
- Address Info:** Includes fields for Street (71194 El Camino Real), City (San Jose), State (CA), and Zip (92344).
- Physician Info:** Includes fields for Last Name (Marshall), First Name (Michael), Street (4550 Stetson Dr), and Phone ((409) 448-3922).
- Trip Info:** A table with columns: Date, Rcv'd, Disp'd, Arrived, Comp'd, and Dispatches.

Date	Rcv'd	Disp'd	Arrived	Comp'd	Dispatches
6/12/92	03:45	03:45	03:53	04:13	T. Michaels
1/21/92	01:55	01:55	02:03	02:20	J. Johnson

At the bottom of the main area are "Save" and "Close" buttons. On the right side, there is a vertical toolbar with icons and labels for "Dispatch", "Patient", "Notes", "Payments", "Billing", "Insurance", "Letters", and "Exit".

Second screen



How was the design improved?

Order out of chaos

This application should contain several child windows for the different tasks a user might perform. These tasks can be accessed easily through the Tasks menu or by pressing one button on the vertical toolbar. The Dispatch button invokes a modal dialog instead of a modeless child window. That way, you can require the user to acknowledge completion of the dispatching task. If it were a modeless window (ie a window that does not require an explicit user choice before continuing), the user might overlay it without ever dispatching the ambulance.

Reordering input fields

The confusing order of fields has been more logically structured based on importance and frequency of use.

Improved controls

The revised interface features consistent use of data entry fields. Any field in which a user can enter data is surrounded by a border. Group boxes are used to group related fields together or to illustrate a domain.

Dialogs

A Dialog provides an exchange of information or dialogue between the user and the application. Use a dialog to obtain additional information from a user that is needed to carry out a particular command or task. Dialogs supplement a user's activities in a primary window.

Dialogs generally appear after the user chooses a particular menu item (including popup or cascading menus) or a command button. Ensure that you provide a traceable path for the user by providing an intuitive menu structure from which to launch the dialog or wizard and descriptive title bars within each dialog which remind the user which menu items or buttons they pressed to bring them to the window that is now in focus.

The title text should echo the name of the associated command or describe the purpose of the window. It should not include an ellipsis i.e. "... " even if the menu name included one nor should it include icons.

Modal vs. Modeless Dialogs

Modal dialogs are useful in complex applications when a user needs to focus on one window at a time, or when a finite task needs to be completed. (Holos set the Modal property to True in the Dialog Details sheet). Use them sparingly for situations when additional information is required to complete a command, or when it is important to prevent any further interaction until a condition is satisfied. Some developers shun modal dialogs because they constrain the user. Make certain the modal window is obvious, visible, the result of an explicit user choice and easy to cancel so the user does not feel the loss of control.



Modeless dialogs provide the user with the flexibility to work in more than one window at a time. They are well suited to situations where a user wishes to repeat an action, such as searching for text. If possible, try to prevent the user from working in more than 3 modeless windows at once.

When to Use Dialogs or Windows

Type	Description	Use	Example
Modal	Dialog	Presentation of a finite task	File Open Dialog Save As Dialog
Modeless	Dialog	Presentation of an ongoing task	Search Dialog Task List Dialog
Application Window	Window frame with document (child) windows contained	Presentation of multiple instances of an object Comparison of data within 2 or more windows.	Word Processor Spreadsheet
Document Window	Modeless dialog or document window contained within & managed by Application window	Presentation of multiple parts of an application	Multiple views of data (sheets)
Secondary Window	Primary window of a secondary application	Presentation of another application called from parent	Invoke Help within an application

Basic Style Guide for Holos Dialogs

Dimming controls

When a control is inactive or should not be selected, then the text and control should be made inactive or grayed. So, for example, if an edit box should not be used until a particular selection has been made, the information label for the control can be grayed by setting the Enabled property to FALSE.

Positioning a dialog

It would be normal to position a Holos dialog in the center of the screen.

Dialogs that expand and contract

Dialogs can be expanded or contracted in order to show further options on the screen. This can be actioned via a button with a title such as Options>>. The button can either be grayed when the window is expanded, or the text can be changed so that the arrows face the opposite direction.

Buttons on dialogs

Buttons should be labeled with the action that they perform. For example, Close or Add. The button will in general be used to close or open another dialog, expand the size of the current dialog or perform another action for the user. If a

button leads to another dialog requesting more information, then the text should be followed by an ellipsis (...).

A message screen may only contain one OK button. Other dialogs, however, must have at least two buttons – one to close the dialog and perform an action (often “OK”) and one to close the dialog without performing an action (usually “Cancel”).

Default buttons

The button, which provides the most sensible next step, would normally be labeled and highlighted as the default one (by setting its Default property to TRUE).

The default button may well be different at different times. For example, if the dialog is complex, various control settings might imply that a different button needs to be clicked at different times.

Placing of buttons

The buttons in a Holos dialog would normally be in two places, for ease of use:

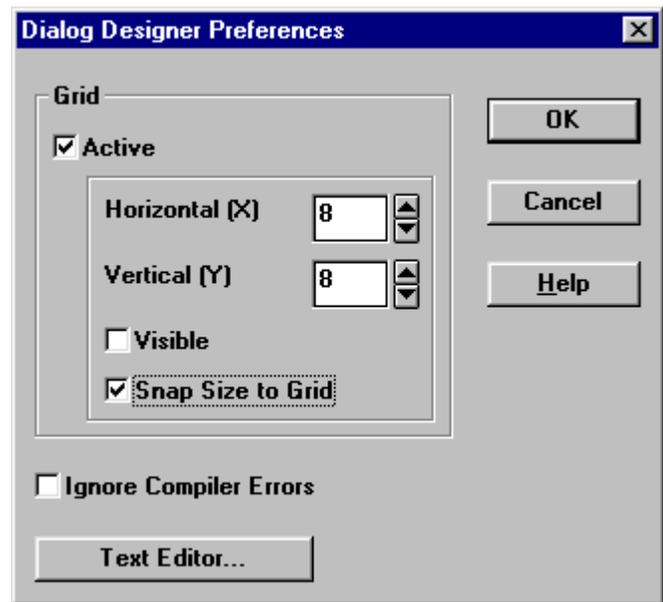
- Down the right hand side of the dialog, with OK and Cancel at the top
- At the bottom of the dialog, with OK and Cancel placed together on the left hand side

Text fonts

The font for an individual Holos dialog is fixed for all of its controls. In order to maintain a standard interface for the user, the same font should be used for all dialogs in the application.

Handy hints when designing dialogs

To ensure related box sizes, especially in a wizard, are the same, the Height, Left, Top and Width properties of the Details sheet should be noted and used for subsequent dialogs.



Making controls the same size

To make several controls, such as buttons, the same size, select each of them by using (Shift+Click), then ensure that the control with the size you wish to match is the anchor item (with the solid blue sizing handles). Select Match Sizes from the Edit menu (same functionality as the Report Designer).

Note that while many controls are selected, the properties they have in common can be changed simultaneously in the details sheet.

If you select a control then create a new control, it will be the same size as the selected one.

When a new dialog is created, the default font is MS San Serif 8 Point, (MS Windows default)

Aligning controls on the dialog

The evenly spaced dots in the Dialog Designer display form a grid, making it easier to align the controls. The grid is visible by default, but you can turn it off. The grid does not have to be visible to be active.

Objects have their positions (top-left corner) snapped if the grid is Active. For fine-tuning of object placement, you can alter grid interval through menu: Options / Preferences.

‘Snap Size to Grid’ determines whether the size (bottom right) is also snapped.

Selected objects can be horizontally or vertically aligned using the special buttons on toolbar (which are available when two controls are selected).

Alternately, to ensure control sizes and placement are consistent from one dialog to the next, set the focus on the control, copy and paste to new dialog.

Using the mouse buttons

If you click the right mouse button, a popup menu of available options is displayed. If more than one control is selected, you can choose to align them, or make them a uniform size. Other options go to the Event tab, or call the Client help for that control.

Sizing drop-down lists

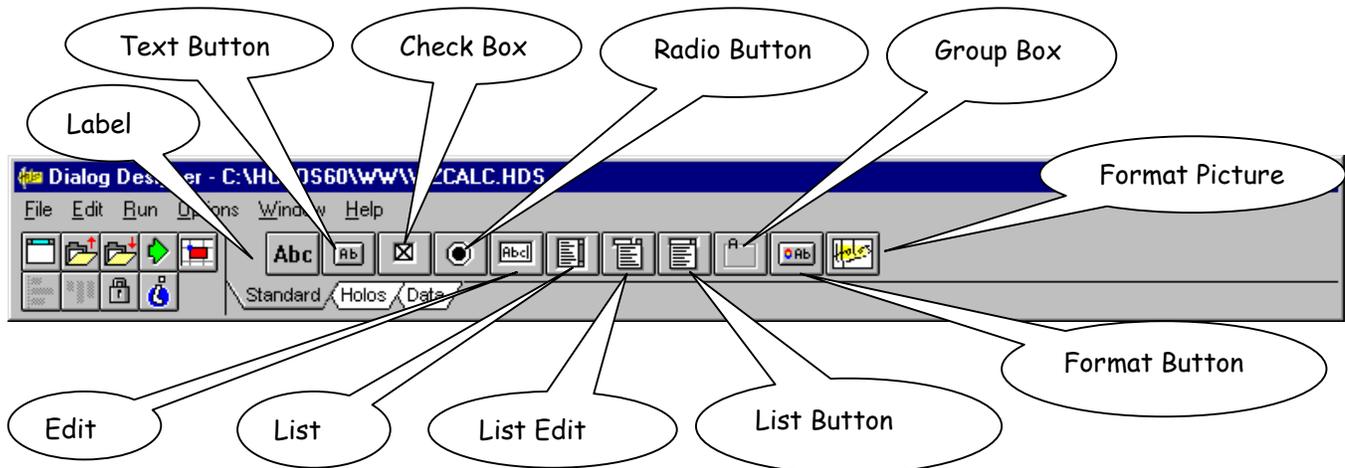
A drop-down list which seems to have no items in it or does not drop down may be too small. In the Dialog Designer the size of a drop-down list determines the size it will be opened at runtime. If a list is not dropping down, it is probably because it was not given enough space in the designer. To rectify this, select the control and pull the bottom-sizing handle down to increase its size.

Locking the position of controls

Once you have designed a dialog, you can lock the positions of its controls, so they can't be accidentally moved. Use the Lock Controls option in the Holos Edit menu or just click the lock / unlock button on the Dialog Designer toolbar. Believe it or not, this will toggle between locking and unlocking all the dialog controls.

Guidelines for Using Controls

Controls are the visual elements that let the user interact with the application.



Label

This control is a textual label often used to label other controls or give instructions.

Text (Push) Button

This is the standard push button used to initiate an action when a user clicks on it.

Check Box

Check boxes act like toggle switches and indicate that the option is either on (if checked) or off. They are typically used for independent or nonexclusive choices. Therefore both states of the choice must be clearly opposite and unambiguous. If this is not the case, then use radio buttons or some other form of single selection choice control instead. The check box may be initially displayed checked or unchecked by setting the Checked property to True or False respectively. Use a single line of text for the label, if possible, as its easier to read at a glance. Use sentence capitalization. Group related check box choices. This grouping does not prevent users from checking the boxes on or off in any combination. You can use a check box to filter the content of a list. If you have a large number of choices or if the number of choices varies, use a multiple selection list box instead of check boxes.

Radio (Option) Button

Radio buttons represent a single choice within a limited set of mutually exclusive choices in a group. You can use a group box to visually define the group. If the number of available options exceeds 6, it is better to use a dropdown list to save space. When the Automatic property is set to true (default setting) for a series of radio buttons, and each button is assigned the same group

name, at run-time, checking a control will automatically uncheck all other controls in the same group. Each option button includes a text label. (If you need graphic labels for a group of exclusive choices, consider using command buttons instead. The standard control allows you to include multiple line labels, use top alignment, unless the context requires an alternate orientation. Define the radio button's label to represent the value or effect for that choice. Also use the label to indicate when the choice is unavailable. Use sentence capitalization.

Possible Button Alternative

Instead of radio buttons and check boxes, consider using 3-d push buttons which do not require mouse target practice.

Edit (Text) Box

The user can type in information, edit it, delete it, or replace it. Information can also be displayed to the user in an edit box. If the text expected in the edit box is a password, setting Password property to True means that the characters typed into the edit box are replaced with asterisks, so that the actual password is not typed on the screen.

List Box

A list box is a convenient preconstructed control that displays a list of choices from which the user can select one or more items. If there are too many items to display, a scroll bar is added. Use a text field to label the list box. The width of the list box should be sufficient to display the average width of the items in the list. For particularly long items, an ellipsis (...) can be used in the middle or at the end of the text string. The use of a horizontal scroll bar is acceptable, bearing in mind it restricts the number of items the user can view at a time. When the user presses a text key, the first item in the list with a matching initial character will be highlighted.

List Edit (Combo) Box

List edit boxes combine an edit box with an attached list and is useful when user input is required and there is a list of possible responses i.e. the user can either type in a response or select one from the list. The text box and its associated list have a dependent relationship. As text is typed into the text box, the list scrolls to the nearest match. If the user selects an item in the list box, it automatically replaces the content in the text box and selects it.

Drop-down list

As with a single selection list box, a drop-down list box provides for the selection of one item from a list; the difference is that the list is displayed upon demand. In its closed state the box displays the current value for the control. While drop-down list boxes are an effective way to conserve space they require more user interaction for browsing and selecting an item than a single selection list box. Make the width of the box a few spaces larger than the average width of the items in the list. The open list component of the control should be tall enough to show at least 3-8 items. If the choices in a drop-down list represent values for the property of a multiple selection and the values for that property are mixed, then display no value in the current setting component of the control.

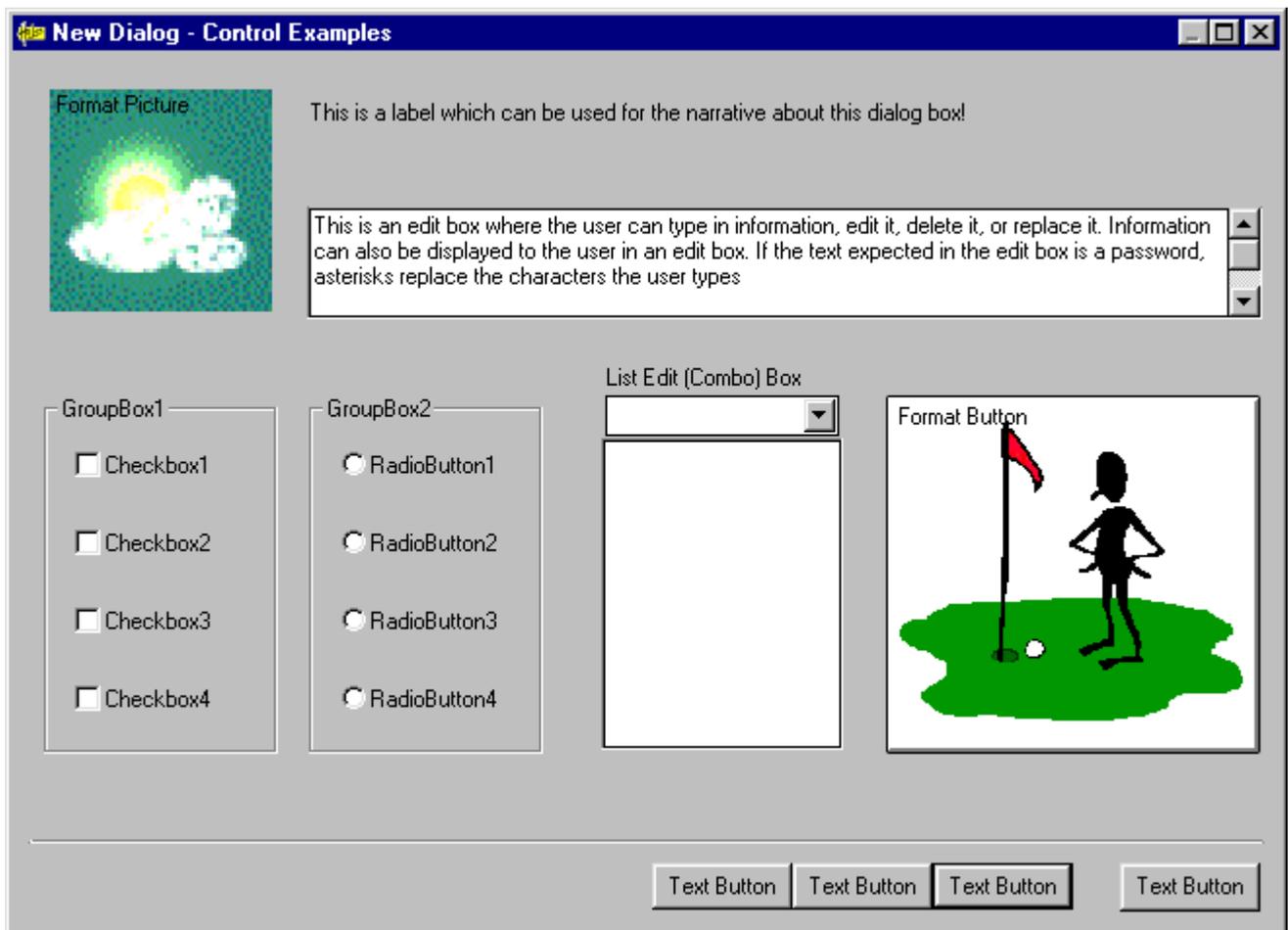
Group Box

Group boxes do not process any input - they are used to provide a logical grouping of related controls. The label is optional and if it contains multiple words, use sentence capitalization.

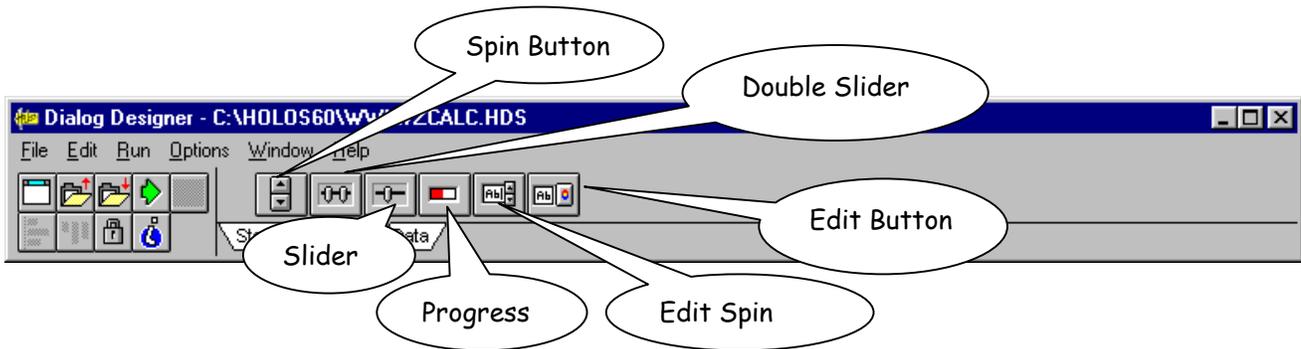
Format Button & Format Picture

These two buttons are standard push button controls. The format picture button displays an icon and/or text, while the format button can display an icon and/or text plus has a button border appearance. The Icon property determines the picture that is displayed on the control.

Creating Standard Controls in the Holos Dialog Designer



Holos Custom Controls



Spin Button

A spin button provides a means for a user to quickly increment or decrement a value in an edit box. Be aware of using the control in situations when the meaning of the arrows may be ambiguous, e.g., dates- does the up arrow mean change to previous date, or increment the date?

Double Slider

This control gives the user a quick method for setting upper and lower boundaries. You can set the slider bars orientation. Use a text field to label the control.

Slider

This control is similar to the double slider but is used when only one limit is to be set.

Progress Bar

This control provides a progress indicator to show the percentage completion of a lengthy process. As the event progresses, the bar is filled with color from left to right. It is useful to add a text label outside the control to communicate its purpose.

Edit Spin

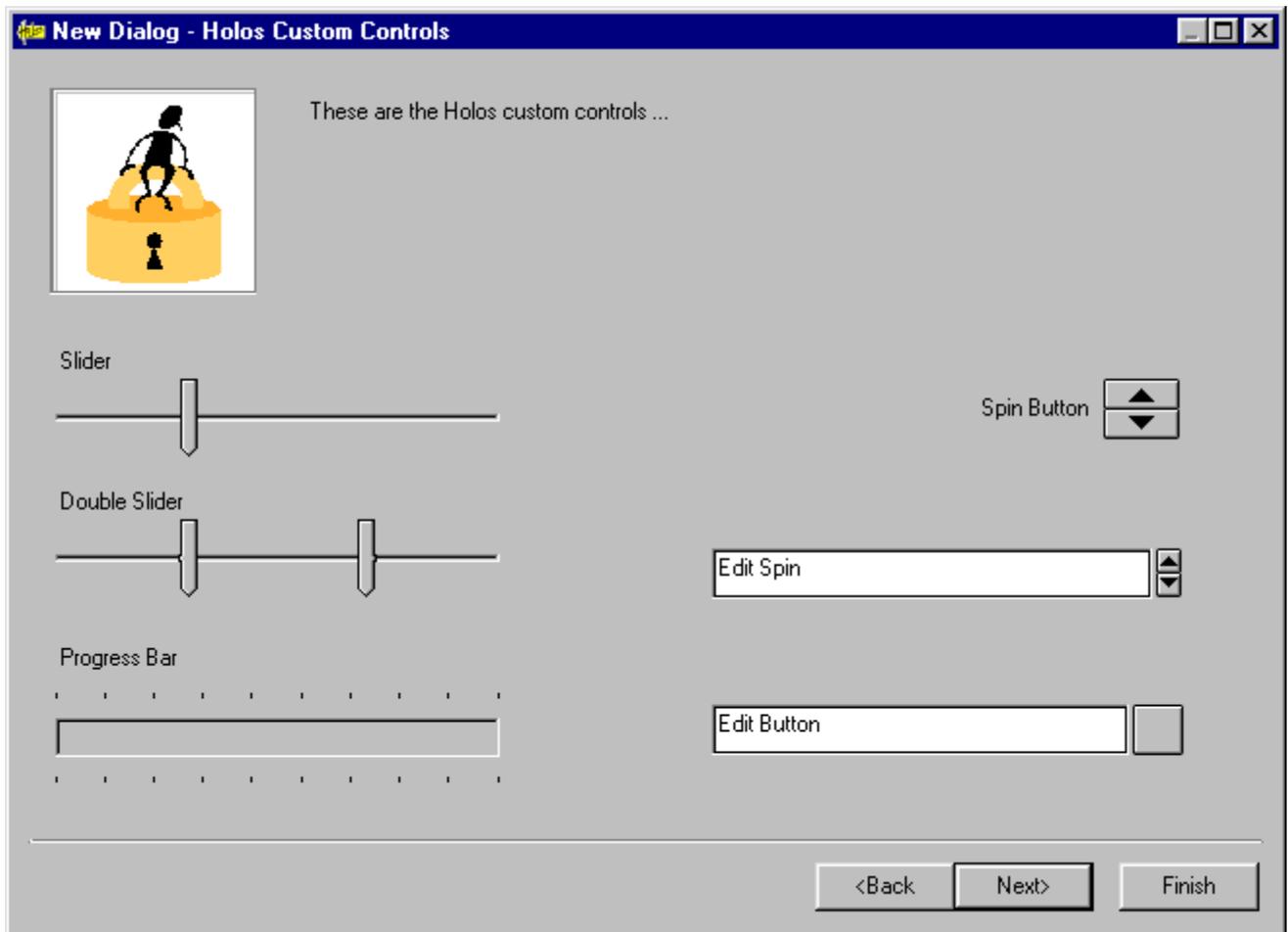
This control provides an integrated combination of a spin control and edit box, so the user can either type in a value (or the application can display a value), or the user can use the up and down arrows.

Edit Button

An edit button control consists of an edit box and a text button. An edit box will either display information, or allow the user to type in a value. A text button is a push button control, which will initiate an action when the user clicks on it.

When something is typed in the edit box or the application displays information to the user in the edit box, the value of the edit button's Text property changes.

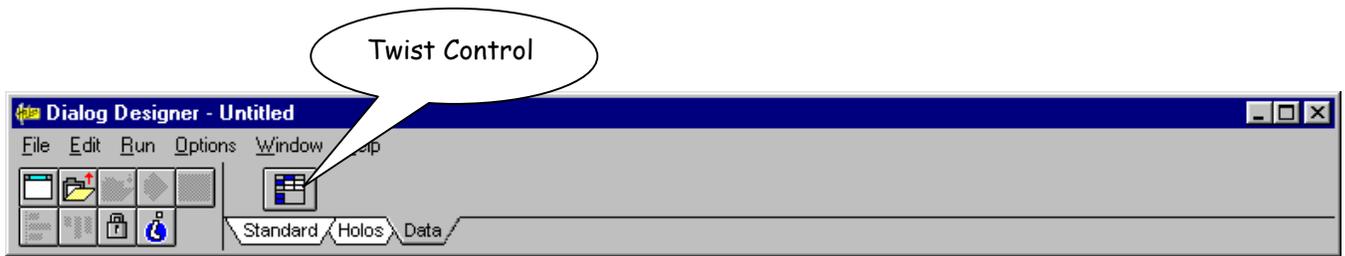
Creating Custom Controls in the Holos Dialog Designer



Guideline for the Number of Controls to Display per Dialog

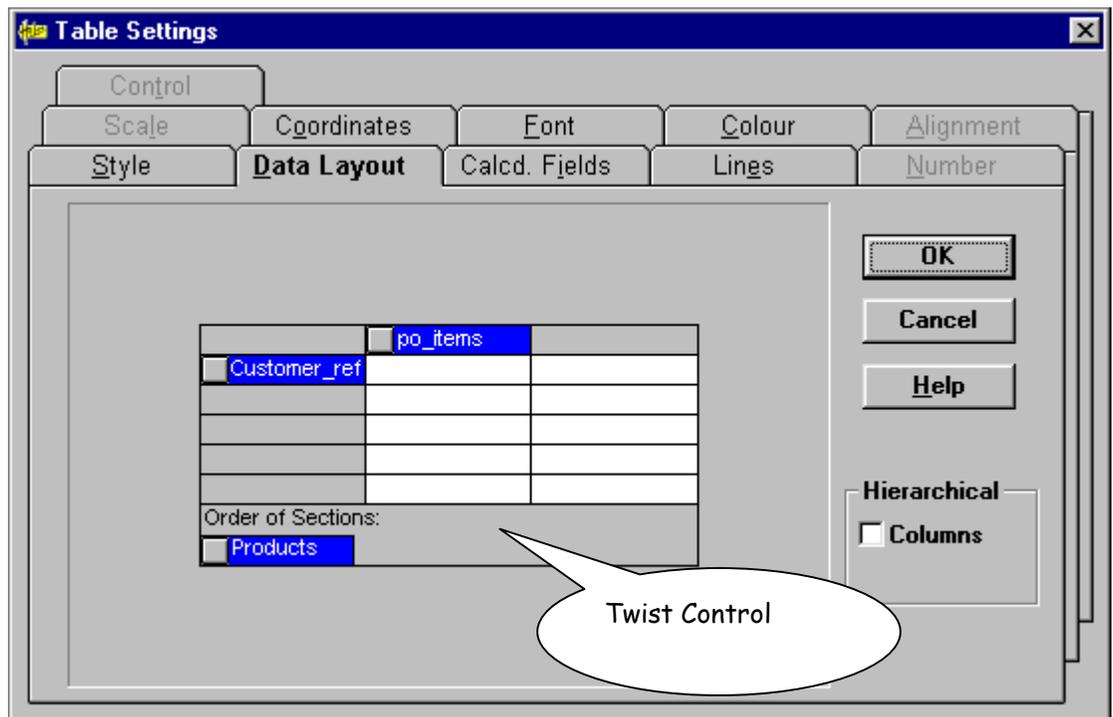
Control	Number of Choices in Domain Shown	Type of Controls
Text Button	1 for each button, maximum 6 per dialog	Static action
Check Box	1 for each box, maximum 10 to 12 per group	Static set/select value
Radio Button	1 for each button, maximum 6 per group box	Static set/select value
List Box	50 in list, display 8 to 10 rows	Dynamic set/select value
Drop-down List Box	Display 1 selection in control at a time, up to 20 in drop-down	Dynamic set/select single value
Spin Button	Maximum 10 values	Static set/select value
Slider /Double Slider	Dependent on data displayed	Static set/select value in range

Holos Data Tab

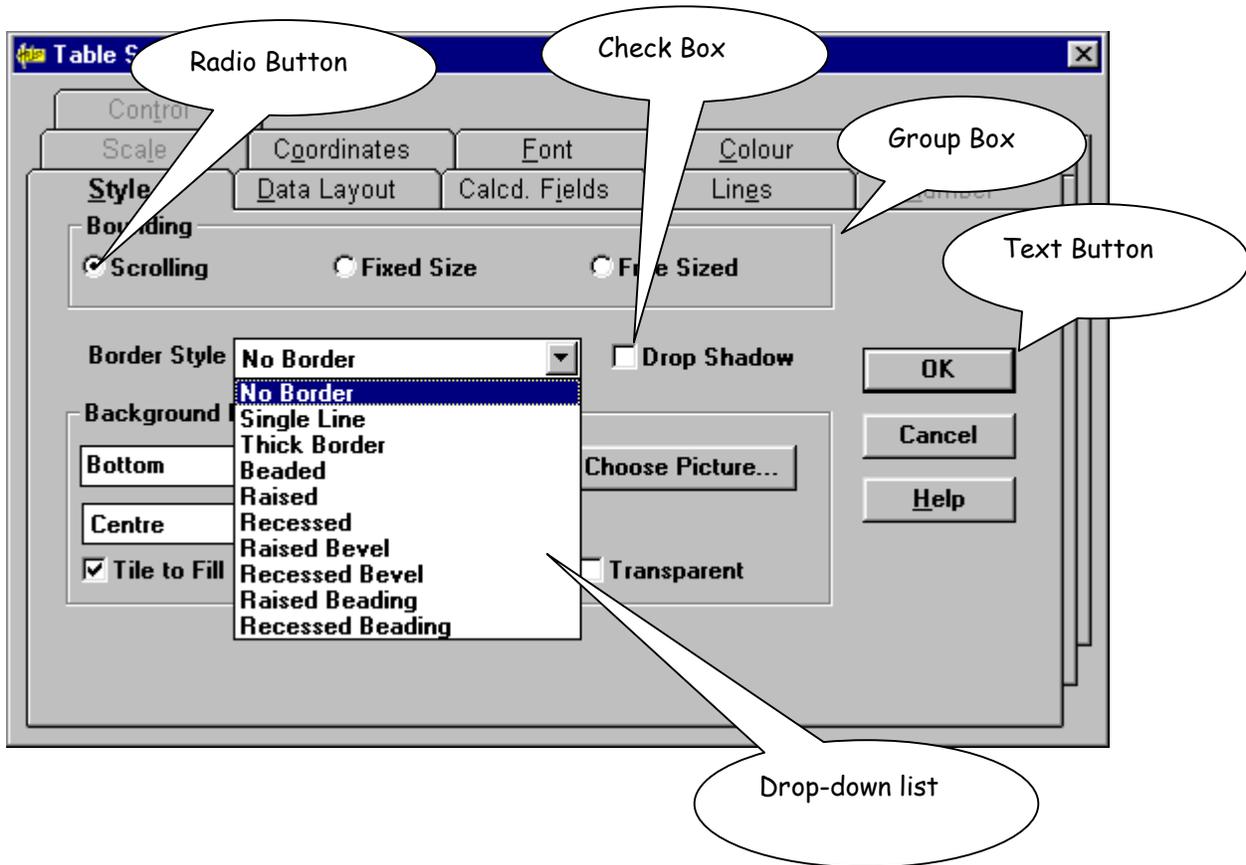


Twist

The twist control allows you to display multi-dimensional data in a table. For example it is use in the Data Layout tab in the report designer for switching the dimension orientation in tables. See below...



Example of Controls in Use in Holos Report Designer

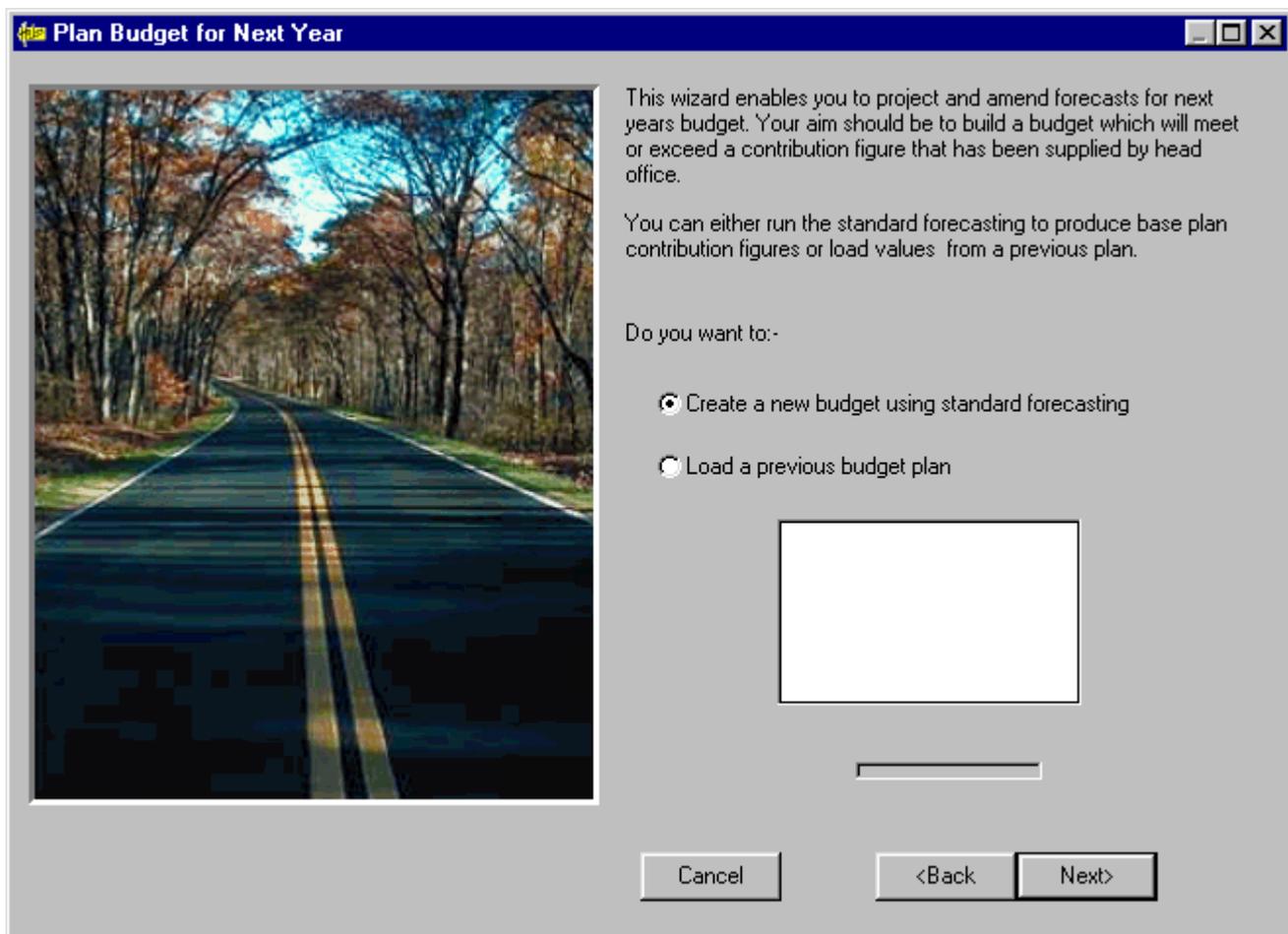


Guidelines for Designing Wizards

A wizard is a special form of user assistance that automates complex or infrequent tasks through a series of presentations or pages displayed in secondary windows. Wizards are not well suited for teaching a user how to do something, as they hide many of the steps and complexities. They are not intended to be used for tutorials.

A wizard should be a supplement rather than a replacement to the user's ability to perform a specific task.

The wizard dialog pages include controls that you design to gather input from the user, which is then used to complete the task.



Please refer to the wizard example above when considering the following tips:

- At the bottom of the window, include the following navigational command buttons:
 - **<Back:** Returns to the previous page. (Remove or disable the button on first page)
 - **Next>:** Moves to the next page, maintaining settings the user selected
 - **Cancel:** Discards any user-supplied settings, terminates the process, and closes the wizard
 - **Finish:** Applies user-supplied or default settings from all pages to complete task.
- Use the title bar text of the wizard window to clearly identify the purpose of the wizard.
- As wizards are secondary windows they should not appear on the taskbar. You may optionally include a context sensitive What's this? Help title bar button as well.
- On the first page of the wizard, include a graphic on the left side of the window to establish a reference point or theme e.g. a preview of the result.
- On the top right of the wizard window provide a short paragraph that welcomes the user to the wizard and explains what it does.
- On subsequent pages, you may also include a graphic for consistency or, if space is critical, use the entire window space for instructional text and controls for the user input.
- Use default values or settings for all controls wherever possible.
- You can include the Finish button at any point that the wizard can complete the task. For example, if you provide reasonable defaults, you can include the Finish button on the first page. Place the Finish button to the right and adjacent to the Next button. On the last page replace the Next button with the Finish button and indicate to the user that when they click the Finish button the wizard will complete the task.
- Design wizard pages to be easy to understand. It is better to have a greater number of simple pages with fewer choices than a smaller number of complex pages with too many options or text blocks.
- Avoid a wizard design that requires the user to leave the wizard to complete a task.
- Make it visually clear that the user interface elements that are part of a graphic illustration page are not interactive. For example, vary them from normal size or render them abstract.
- Avoid advancing pages automatically. The user may not be able to read the information before a page advances. Wizards are intended to allow the user to be in control of the process that the wizard automates.
- Make certain that the design alternatives offered by your wizard provide the user with positive results.
- Make sure that it is obvious how the user can proceed when the wizard has completed its process, perhaps with text on the last page of the wizard.

Guidelines for Writing Text for Wizard Pages

- Use a conversational rather than instructional style for the text. Concise wording of text labels, user error messages and online help is often a challenge for the developer, but well worth the investment of a little extra time to achieve an intuitive efficient wizard.
- Use words such as “you” and “your”.
- Start most questions with phrases like “Which option do you want...” or “Would you like...” Users respond better to questions that enable them to do a task than being told what to do. For example, “Which layout do you wish to use?” rather than “Choose a layout.”
- Use contractions and short common words.
- Avoid using technical terminology that may be confusing to a novice user.
- Try to use as few words as possible. For example, “Which style do you want to use for this report?”, could be shortened to: “Which style do you want to use?”
- Keep the writing clear, concise and simple and remember not to be condescending.

Programming standards

Position of objects in the Application Manager

While a dialog is being developed in the Application Manager, it is useful to group the dialog, update function and dialog record together in one place. This makes it simple to test and develop the dialog. However there is no reason to place the objects together once the dialog has been tested.

File naming conventions

The file name extension used for a file containing a dialog or a number of dialogs is .HDS. For example, TOY_DEMO.HDS. The file must be in a PC working directory when the application is running, although you can store the HDS file on the Host machine.

Single file as opposed to multiple files

Holos allows the developer to store more than one dialog in a file, making them easier to maintain. If the file grows to large, however, Holos will take longer when compiling the dialogs as it has to compile each of the dialogs in the HDS file.

Naming conventions for records, compound records and functions

The records and functions used when a dialog is running follow the same naming conventions as other HOLOS functions and records i.e. .FNC or .REC. Use of the words dlgreg or dlgfnc in their names, although not essential, helps in distinguishing them as dialog records or functions. For example, for the dialog input_rates, the associated record and function could be named:

input_rates_dlgreg_rec and input_rates_dlgfnc_fnc.

Dialog control naming conventions

There is no fixed convention for naming controls on a dialog (except for the standard naming convention for Holos objects)

A sensible naming convention that makes a control easy to recognize should be used. The following list is proposed in the “Designing Holos Dialogs”:

Type of control	Suggested Name
Radio button	RB
List button	LB
Button	Button?
Check box	CB
List	List
Edit spin	ES
Single Slider	SS
Double slider	DS

Other Naming Suggestions

Alternately, this excerpt from “Holos Application Development Guidelines” by Steve Martinez contains other suggestions:

The HOLOS Dialog Designer gives each object a default name upon creation but, in order to ease development, it is good practice to rename each object with a more descriptive name. Each object should be named in the following format:

idName

where :

- **id** is a unique three-letter identifier for the type of object
- **Name** is a descriptive name that identifies the function, purpose, or contents of the Dialog object.

A complete list of HOLOS Dialog objects and their identifiers appears below.

The Dialog itself should be named to reflect its functionality or purpose. For instance, if a dialog was created that sets some report attributes, then the dialog should be given the name ReportAttributes_dlg. Each dialog must have a record associated with it. For ease of identity, this record should contain the name of the dialog it is associated with followed by 'rec', the three-letter identifier for a record object (i.e. for the dialog ReportAttributes_dlg, the associated record should be named reportattributes_dlgrec). For records associated with the non-standard controls, the name should reflect the dialog and the object the record belongs to. For example, the control sbnMaxValue ought to work with the record ReportAttributes_sbnMaxValue_rec.

HOLOS Dialog Object Prefixes

Object Type	Object ID	Examples
Standard Objects		
Text Button	tbn	tbnOK, tbnCancel
Label	lbl	lblTitle, lblDimension
Check Box	chk	chkAllObjects
Radio Button	rbn	rbnShowPCFile, rbnShowHostFile
Edit Box (Text Box)	txt	txtLowerLimit
List	lst	lstFields
List Edit	led	ledDimension
List Button	lbn	lbnReport
Group Box	grp	grpFileStore
Format Button	fbn	fbnConfigure
Format Picture	pic	picLogo
HOLOS Objects		
Spin Button	sbn	sbnMaxValue
Double Slider	dsl	dslKPI
Slider	ssl	sslMaxValue
Progress Bar	pro	proCompleted
Edit Spin	esp	espHour, espMinute
Edit Button	ebn	ebnReport
Data Objects		
Twist	twst	twstToys_str

Mouse & Keyboard Interface

You should provide equal keyboard and mouse support for all window and menu operations. Features such as mnemonics (alphanumeric keys used with Alt key), accelerator keys and toolbar buttons with meaningful icons allow the power user to control the GUI and rate of data entry.

Icon design and toolbar layout have more than an ancillary effect on users; they are the interface between the user and the application and must also be designed to provide simple, obvious, inoffensive and universal meanings. A successful icon is easy to recognize and understand, and is unambiguous.

List of Reserved Words

Text	Meaning & Behavior	Appears On Button	Appears On Menu	Mnemonic Keystroke	Shortcut Keys
OK	Accept data entered or acknowledge information presented & remove window	Yes	No	None	<Return> <Enter>
Cancel	Do not accept data entered & remove the window	Yes	No	None	<Esc>
Close	Close current task & continue working with application; close view of data	Yes	Yes	<Alt>+C	None
Exit	Quit the application	No	Yes	<Alt>+X	<Alt>+<F4>
Help	Invoke the application's Help facility	Yes	Yes	<Alt>+H	<F1>
Save	Save data with a new name	Yes	Yes	<Alt>+S	<Shift>+<F12>
SaveAs	Save the data with a new name	No	Yes	<Alt>+A	<F12>
Undo	Undo the latest action	No	Yes	<Alt>+U	<Ctrl>+Z
Cut	Cut the highlighted characters	No	Yes	<Alt>+T	<Ctrl>+X
Copy	Copy highlighted text	No	Yes	<Alt>+C	<Ctrl>+C
Paste	Paste copied or cut text at insertion point	No	Yes	<Alt>+P	<Ctrl>+V

Reserved Icons

- i** Information message
- !** Warning message
- ?** Question message
- X** Critical message

Finding More Information / References

Here are the references used for the dialog and wizard design guidelines:

- The Windows Interface Guidelines for Software Design
- Designing Holos Dialogs
- Holos Developer Tools Guide
- Holos Application Development Guidelines
- Unix Review Magazine

Contacting Crystal Decisions for Technical Support

We recommend that you refer to the product documentation and that you visit our Technical Support web site for more resources.

Self-serve Support:

<http://support.crystaldecisions.com/>

Email Support:

<http://support.crystaldecisions.com/support/answers.asp>

Telephone Support:

<http://www.crystaldecisions.com/contact/support.asp>