Controls Section

Controls

Overview

In a graphical user interface (GUI), the controls are how the user communicates with the software application. How well that communication takes place depends on two very important things:

1. The appropriateness of the control for the task the user is trying to perform.
2. The consistency in behavior of each control.

If you select the wrong control for the job, or change the normal rules of behavior under which the controls operate, you will create usability problems for the user. In turn, these problems may cause users to avoid using your application.

While there are many controls in the Windows toolbox from which a developer may choose, this document does not attempt to address all of them. Instead, this document focuses on controls that will be used primarily for data entry. In addition, the focus of this document is on describing standards that will lead to a consistent and logical approach to the use of controls both within and across CDC developed applications. This latter focus is critical since the standards outlined here will be adopted when creating controls in reusable components for use in all CDC developed applications.

To achieve this desired consistency, the developer may notice that a few common Windows controls are not permitted. For example, radio buttons will not be used in CDC developed applications. This is not because the control is not important; rather, it is because the "one or more" choice selections using the list box or combo box will lead to greater consistency within and across all applications. In other cases, common controls, such as tree view controls, may be permitted but have not been included in this section since they relate more closely to navigation. These types of controls will be more fully detailed in the section on Navigation and Metaphors.

General Control Standards

The following control standards are to be adopted across all CDC software applications. These standards should lead to a more consistent and usable interface.

General

Align controls on left edges.

For information that is vertically positioned, align fields to the left edge. This makes it easier for the user to scan the information. Text labels are left aligned and in most cases placed to the left of the areas to which they apply.

Provide control labels that are left justified.

Labels are necessary to identify the contents of the control. All control labels will be left justified, giving them an organized look and making them easier to scan.

Gray out controls and control labels if the control is unavailable.

When controls are disabled or unavailable to the user, indicate they are disabled by making the foreground and background of the control and its label gray. These changes will provide a visual cue to the user that the action can't be performed.

Upon mouseover of a disabled control, alter the appearance of the cursor

If the control is disabled, provide a visual clue to the user that they can't interact with the control by altering the appearance of the cursor. The appropriate symbol will be described in the *Graphics* section of this document.

Provide access keys to controls that can handle access keys.

Access keys provide an alternative means by which the user can navigate to any control on the interface. Access keys should not be assigned to group boxes unless the group box is being used to supply keyboard interaction. The OK and Cancel buttons should not have access keys when they are mapped to default and escape. Otherwise if the control can handle access keys, use them.

Use conventional book title capitalization in command buttons, menus, tabs, title bar text, or icon labels.

Capitalize the first letter of each word unless it is an article or preposition not occurring at the beginning or end of the name, or unless the word's conventional usage requires it to not be capitalized.

Use sentence style capitalization for check boxes, text boxes, and group boxes.

Capitalize only the first letter of the initial word and any words that are normally capitalized.

Check Boxes

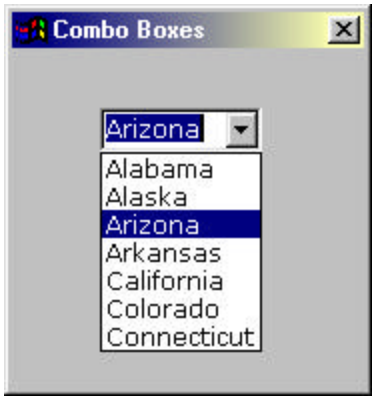
Use check boxes for toggling between two choices.

Check boxes are preferred when dealing with a choice that is either on or off. For example, when determining whether a case has been imported into this country or not, use a simple check box as shown in the figure above. When the box is checked, the data in question is imported; when the box is not checked, the data in question is not imported.

Left justify all check boxes

All check boxes will be left justified with the text following on the right.

Drop Down Combo Boxes



Drop down combo boxes allow you to make a single selection or to enter text in the text box portion of the list. This allows the user to specify an entry that is already in the list, or to specify a new one, when allowed. Some development tools offer simple combo boxes, however these will not be use by CDC applications because they provide no visual clue that there are selections available.

Use drop down combo boxes when there are multiple items from which to choose.

Drop down combo boxes are appropriate when the user needs to make a single choice from multiple items or when entering a single entry text that is not available on the list. Upon tabbing to the field, the drop down combo list will appear to the user. Upon tabbing away from the field, the drop down combo list will collapse. This is a non-standard control that CDC will develop and distribute for use.

All drop down combo boxes will follow CDC minimum height and average width standards:

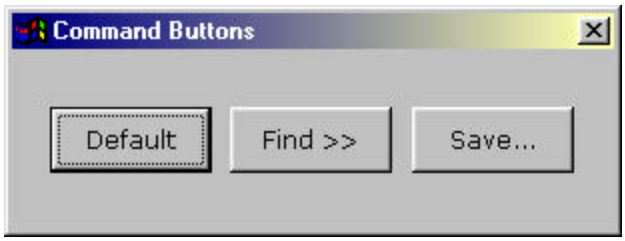
Minimum height	***22 pixels***
Width	***average field entry width***

The height of a drop down combo box is usually defaulted by the development tool. However make sure that the height is not smaller than 22 pixels. The width of a drop down combo box will be based on the average width of the entries that are contained in the list.

Provide auto-selection when keying in choice

Drop down combo boxes have the ability to automatically select an item in the list based on the user keying in the choice. Therefore, when the user knows the item list from memory, and can key in an item code or the actual item, provide this ability.

Command Buttons



All command buttons will follow CDC standard height and minimum width standards

Standard Height	***33 pixels***
Minimum Width	***81 pixels***

Following these command button measurements will give all CDC software applications a consistent, simple appearance. Command buttons with a greater

height tend to distract users. While there is no maximum command button width, a command button wider than 292 pixels should be avoided.

All command buttons will be rectangular in shape

While many of the development tools allow developers to customize the shape of command buttons, anything other than a rectangular command button is considered distracting. Do not clip or round any of the command button corners.

Command button text will be centered

Text captions that appear in command buttons will be centered within the button.

Command buttons will not contain graphics

Command buttons will contain text description only and no graphics. If you find yourself also needing a picture to explain the text, you probably need to rename the command button. The exception to this rule involves the use of the chevrons on the expanding dialog command buttons as documented below.

Label OK and Cancel buttons as in primary windows follows:

The OK button will be labeled **OK**, not *Ok* or *OK*. The Cancel button should be labeled **Cancel**, not *Cancel* or *CANCEL*.

Main command buttons will be in the following order:

OK, Cancel, Help

The OK button or its equivalent should always be the first main command button. Cancel should be to the right or below the OK button. The Help button should always be the last button. If there is not an OK button, but other command buttons, it's best to place the Cancel button at the end of a set of action buttons, but before the Help button if one is present.

OK and Cancel buttons will not be used in a modeless dialog

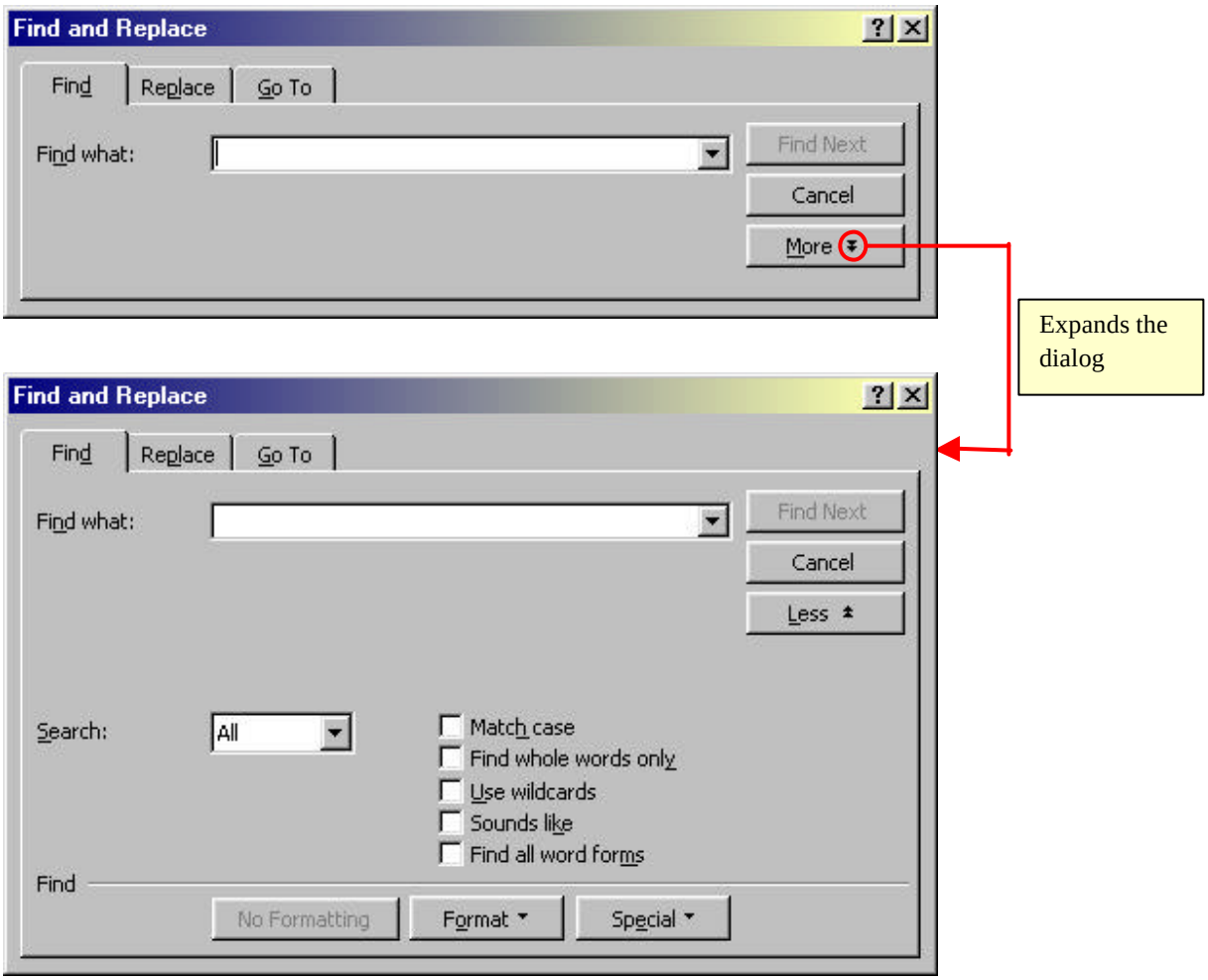
For modeless dialog boxes or dialog boxes used as primary windows, use a Close button rather than OK and Cancel. Using OK and Cancel on a modeless dialog makes it appear as if it were modal. OK and Cancel are not meaningful in a modeless context. Use Close to eliminate confusion. If possible replace OK with Yes, No or some other action.

Use chevrons (>>) on command buttons to expand dialog boxes

Use chevrons on command buttons to imply an expanding dialog box to show additional information. The figure titled *Expanding Dialog* illustrates the Find and Replace expanding dialog that appears in Microsoft Word. Chevrons may be illustrated on the button using the greater than (>), less than (<), or carrot (^) in

the label text. Chevrons may also be illustrated using a black filled triangle as shown in the example below.

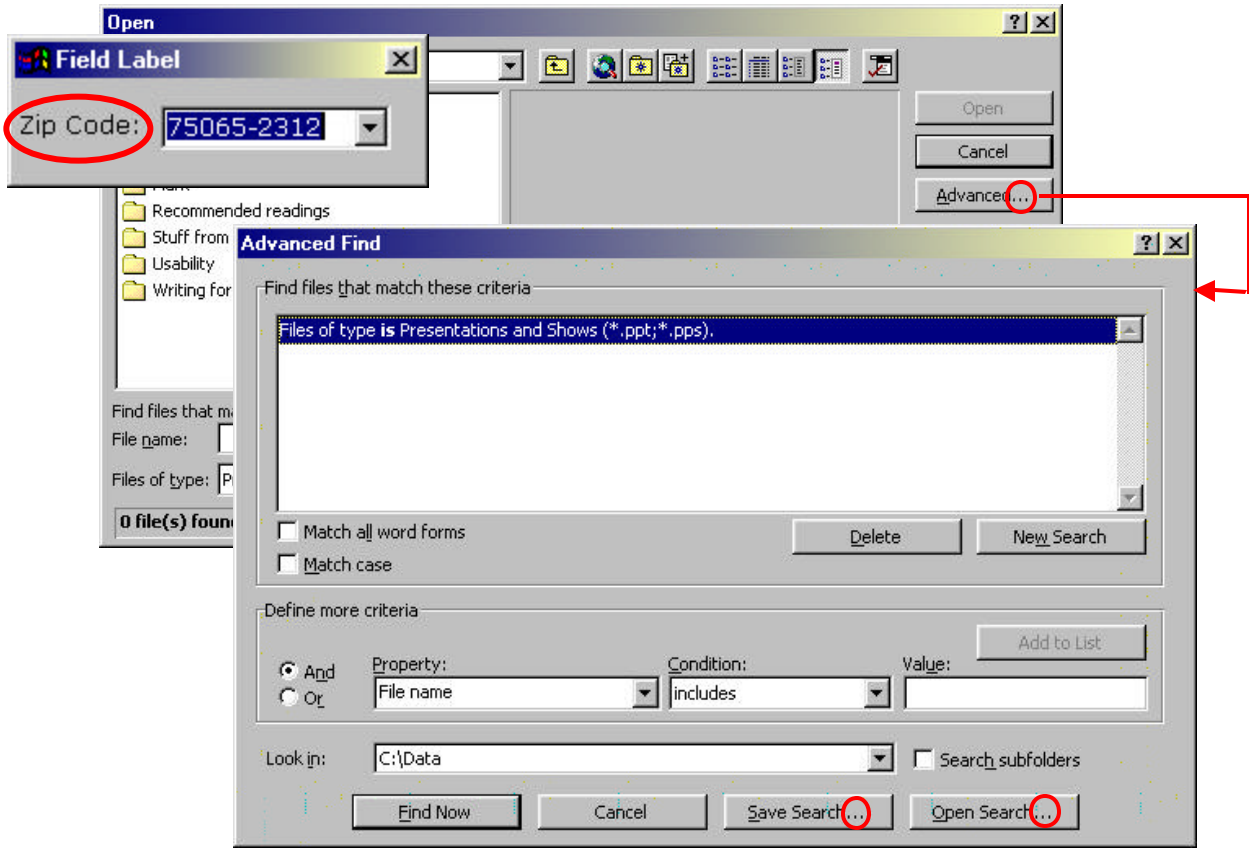
Expanding Dialog



Use ellipses (…) on command buttons to provide additional dialog information for a single dialog.

Use ellipses on command buttons to imply than an additional dialog box of supplemental information is to be displayed. The figure titled *Supplemental Dialogs* illustrates the use of ellipses in Microsoft Powerpoint on the Open Dialog. If the user cannot locate a file to open with the folders, an advanced find dialog can be initiated by clicking on the Advanced… button.

Supplemental Dialogs



Field Labels

Labels will follow the CDC height standard:

Height 24 pixels

Make all labels heights consistent across the interface.

Left justify field labels

Left justification of field labels gives them an organized look, making it easier for the user to scan.

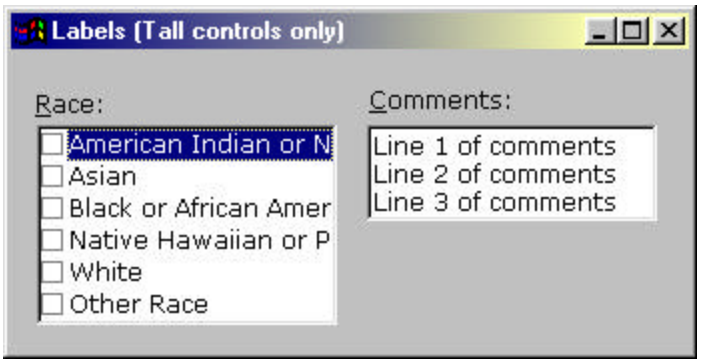
Place all field labels to the left of their associated control.

This alignment makes it easier to find the label and its associated control. The obvious exception to this rule involves group boxes and tall controls where the control takes up more than a single line of text. When using a drop down list, there also may be situations where the layout of the window can be better optimized if the label occurs at the top left like other lists.

Place all tall control labels on top and to the left of their associated control.

Tall controls are those controls that take up more than one line such as lists, trees and multi-line text boxes. In these cases, place the text on top of the control and to the left as shown in the figure titled *Tall Control Labels*.

Tall Control Labels



Use colons at the end of field labels

Use of a colon clearly indicates that the text is to identify the contents of a control.

Group Box labels will not have colons.

Group box labels contain supplemental information and therefore will not contain colons.

Labels will not have borders

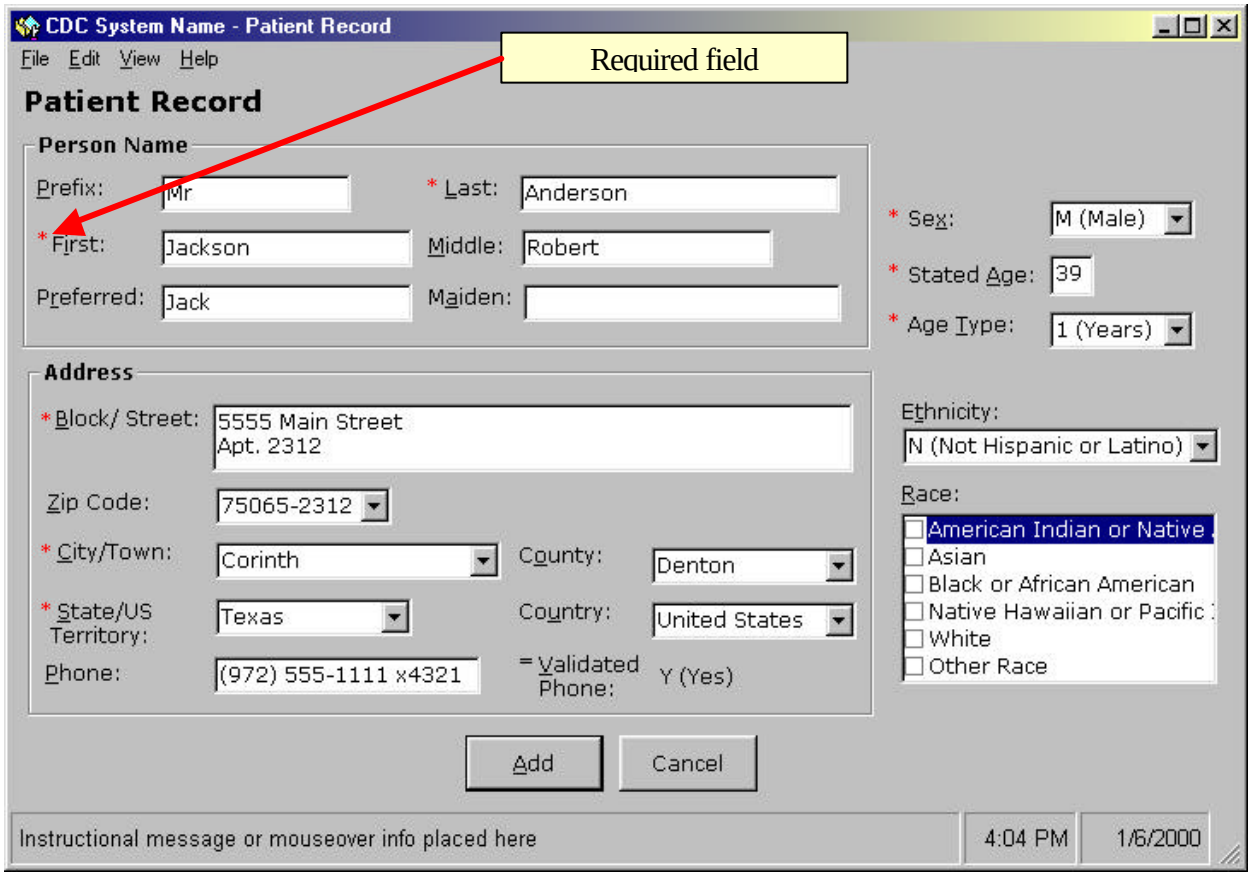
All labels, whether to label a field, or to give supplemental information, will appear flat with no borders.

Required field labels will be designated according the the following CDC standard:

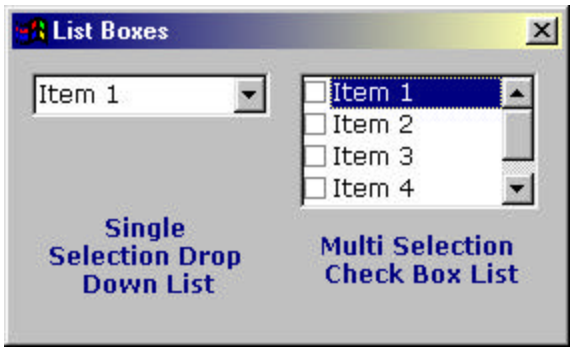
asterisk (*) preceding the label *Label name

Users need clues that indicate which fields are required and which fields are optional. All fields that *must* have user entered data will be designated as illustrated below in the figure titled *Required Fields Example*.

Required Fields Example



List Boxes



There are 2 types of list boxes that will be used in CDC software applications: Single selection and multiple selection. The table titled *List Box Behaviors* defines the 2 types of list boxes. Some development tools implement drop down list b

List Box Behaviors

List Box Type	Behavior
Single Selection Drop Down List	Allows the user to make only a single selection from the list. These lists can be regular lists or *dropdown* lists. CDC software will only use the drop down list for single selections.
Multi Selection Check Box List	Allows multiple selection. Clicking any item toggles its selection state in the check box to on regardless of the Shift and Ctrl keys.

Use a single selection drop down list when there is only one selection from multiple items and the user is not allowed to create new items in the list.

Always choose a single selection drop down list box when listing multiple items is required, but where the user cannot key in a new entry as they can in the combo box. Upon tabbing to the field, the drop down list will appear to the user. Upon tabbing away from the field, the drop down list will collapse. This is a non-standard control that CDC will develop and distribute for use.

Use a multi selection check box list when there are multiple sections and the user is not allowed to create new items in the list.

Always choose a multi selection check box list when selection of more than one item is required.

Multi selection check box list height will follow CDC height standards:

no fewer than 2 items
no more than 8 items.

All multi selection check box lists will display no less than 2 items and no more than 8 items. If there are more than 8 items a vertical scroll bar will be provided. The only exception to this standard is when the size of the list varies with the size of the window.

List box width will will follow CDC width standards:

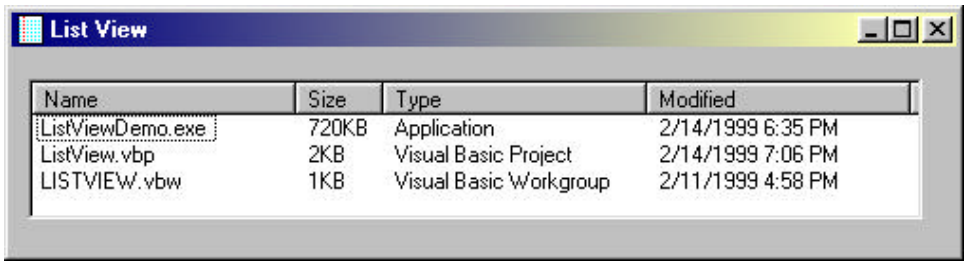
based on average entry width.

The width of a list box will be based on the average width of the entries that are contained in the list.

Provide auto-selection when keying in choice

List boxes have the ability to automatically select an item in the list based on the user keying in the choice. Therefore, when the user knows the item list from memory, and can key in an item code or the actual item, provide this ability.

List Views



The ListView control displays and arranges items into columns with or without column headings. It can also display accompanying icons for each column, if necessary.

Use list views when three or more columns of information must be displayed or when columns require sorting.

List views should be used when three or more columns of information are needed for display. List view column headers can serve two purposes: sorting and column content identification. While list boxes and combo boxes can have tabs set to display multiple columns of information, they do not have the ability to display individual column labels for sorting. When displaying multiple columns of information that needs a content label to make the data displayed in that column more meaningful, the list view is the control of choice.

Spin Boxes

Use spin boxes when incrementing or decrementing a numeric values.

Spin boxes will be used when dealing with numeric input only. They are most useful for having the user select some numeric choice.

Spin boxes will follow CDC height standards:

Height 24 pixels

Make all components of the spin box 24 pixels high. This includes the text box and the up/down arrows.

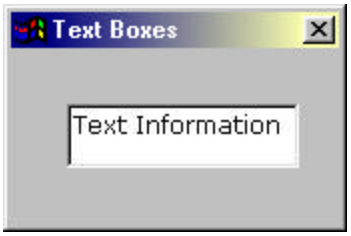
Spin boxes will support alternative keyboard entry.

To allow the user flexibility in specifying a numeric choice, the text portion of the spin box will allow keyboard entry as an alternative to the up/down arrows.

All spin boxes will have left aligned labels.

Just like text boxes, spin boxes will have labels to identify the contents of the spin box. The label will be to the left of the spin box.

Text Boxes



Text boxes will follow the CDC height standard:

Height 24 pixels

Make all text box heights consistent unless it is a multi-line text box.

Text box width follow CDC width standards

Width reflects data width

When possible make the text box as long as the data field. This gives the user a clue as to the length of the field that is to be entered. However, there will be times that the text box width may need to be shortened or lengthened to assist in the vertical alignment of the window. Fields excessive in length need not reflect the width of the data since text boxes are automatically scrollable.

Enterable text fields will be recessed with a white background.

To conform to the rules of visual affordance noted in the Windows section, all fields that can have data entered into them will be recessed with a white background.

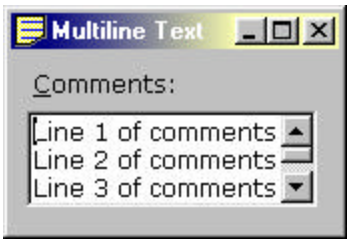
All text boxes will have a field label

Labels are necessary to identify the contents of the text box. Labels that are to the left of the text box should be aligned vertically.

Include vertical scroll bars on multi-line text boxes

Providing vertical scroll bars gives the user a clue that multiple lines of entry are allowed as shown in the figure titled *Multi Line Text Example*. This will help distinguish it from the single line text box.

Multi Line Text Example



General Control Guidelines

The following guidelines have been established within the Windows user interface design community and should be applied across CDC software applications.

General

Avoid using unnecessary custom controls

Nonstandard controls do not look or behave like most other Windows interfaces, and may draw attention to interface elements that shouldn't really be receiving attention.

Eliminate unnecessary scroll bars

Scroll bars make the interaction with any control more complex. Make controls long enough and wide enough to eliminate the need for scroll bars whenever possible.

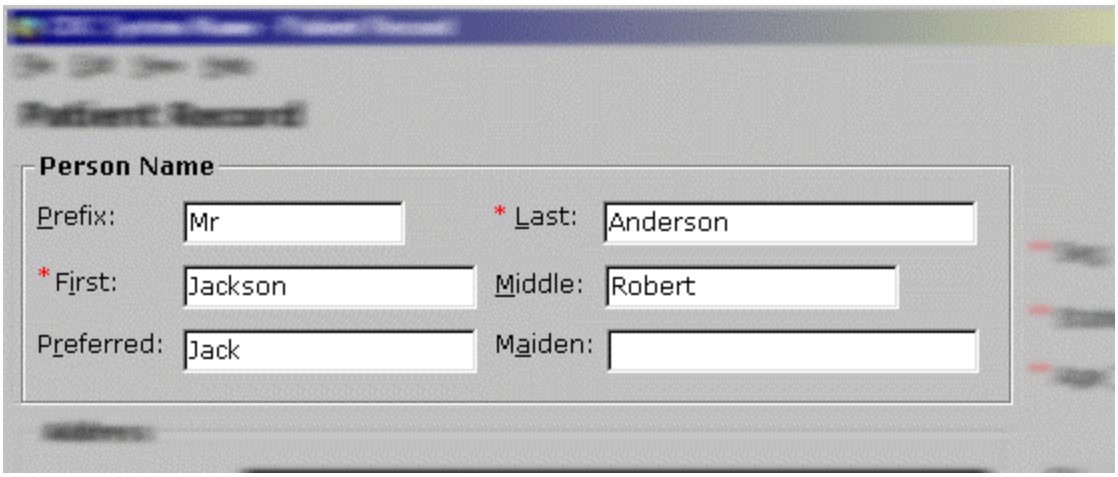
Avoid abbreviations.

When providing text labels or descriptions, avoid using abbreviations unless all users, no matter what their skill level, will understand them. Acronyms should also be avoided unless they are clearly understood by the entire user population.

Frame related information together

Framing related information makes reading and scanning easier for the user. When grouping the related information that is being shown in a *single window view*, provide a descriptive label for the group. A group box makes it clear what information is related by providing clear borders. For example, on a window rich with information, it would be wise to group all information related to a patient's name in a group box as illustrated in the figure titled *Framing Example*.

Framing Example



Default control values whenever possible.

User interfaces are intuitive and effective when there are default values displayed during insert or add actions. A user should not be required to select or type values that are normally populated into given fields.

Controls that will never be available to the user for a particular window session due to access rights should not be visible.

Often times access rights can prevent users from being able to access particular functionality during their windows session. In other words the user cannot interact with the window to make the control available. Only changing the access rights outside of this window session could impact accessibility to that control. When large areas of functionality are unavailable, the control(s) associated with that functionality should not be shown. An example would be if the users access rights prevent them from being able to access all of the menu items in a menu, then the menu should not be shown. Another example might be if the user does not have access to several nodes on a treeview, don't show the nodes. However, when the user's interaction with the window impacts what they have access to

then disabling the control or making it not visible is left to the discretion of the programmer.

Command Buttons

No more than 6 command buttons will be shown per window

Command buttons are for frequent and critical actions. If the actions are not frequent or critical, put them in the menu.

Command buttons should be positioned left to right or top to bottom

Position command buttons the way people read in Western cultures. Main command buttons should be positioned on the bottom or to the right, in a column making them easier to find. Place command buttons on the bottom if they have different sizes. Command buttons that are on secondary dialog boxes or property sheets should be positioned on the bottom and to the right of the window.

Avoid multiple rows or columns of command buttons

Multiple rows and columns of buttons tend to overwhelm the user. Consider using a command menu instead. If you must have many buttons, multiple rows are preferred to multiple columns.

All windows should have a default command button

Make the most common, nondestructive button on the window the default button.

List Boxes, Drop Down Combo Boxes, and List Views

Provide full row selection on all lists.

Drop down combo boxes and list boxes automatically highlight the entire row when a column is selected. List views do not function similarly. There are Windows APIs available which enable you to give the list view the same full-row highlighting behavior as the drop down combo box and list box.

Provide clickable column headers only if the list can be sorted.

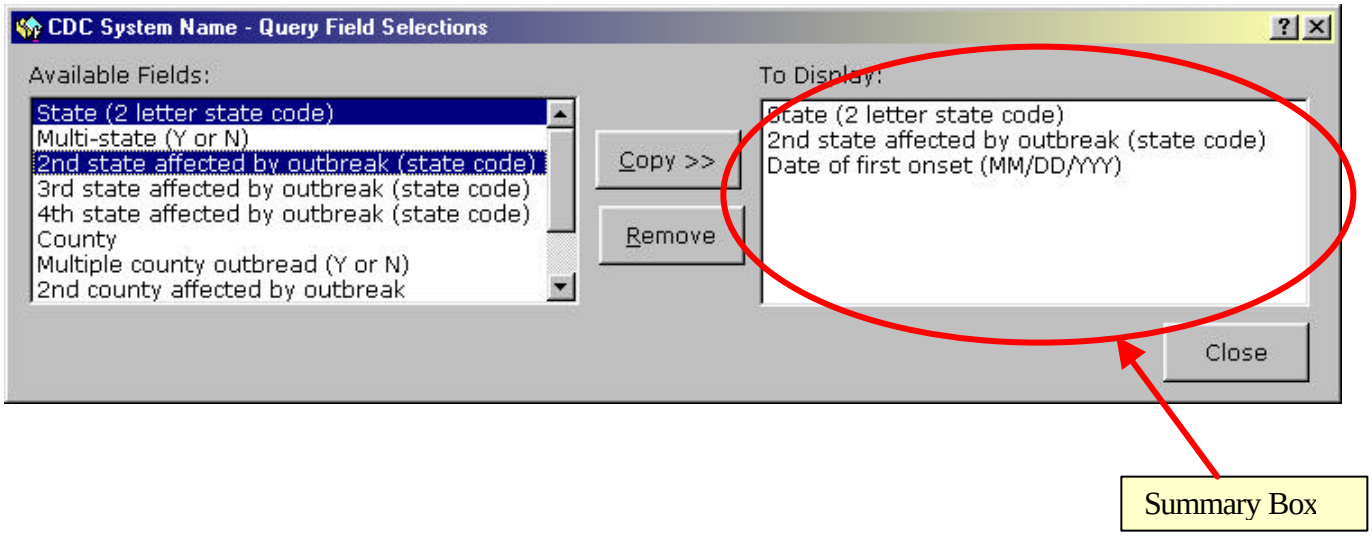
Clickable column headers should not be used for anything other than sorting. The first click sorts the list in normal order; the second click sorts the list in reverse order.

Provide summary boxes for very long multiple selection lists.

When lists are very long and the user is allowed to make multiple selections, provide some type of summary box which tells the user what they have selected

so that they don't have to scroll through the lists to view their selections. In addition, if it is likely that the user will want to select all of the items or deselect all of the items, provide a select all or deselect all option. In the figure titled *Summary Box Example*, the user may select from a list of dozens of fields to display. The summary box on the right indicates those fields selected to be displayed.

Summary Box Example



Recommended Readings

Cooper, Alan. *About Face: The Essentials of User Interface Design*. IDG Books Worldwide, Inc., 1995
Chapter 25 - Imperative and Selection Gizmos
Chapter 26 - Entry and Display Gizmos

McKay, Everett N. *Developing User Interfaces for Microsoft Windows.* Microsoft Press, 1999
Chapter 4 - Establish a Consistent User Interface Style
Chapter 10 - Good User Interfaces are Simple
Chapter 29 - Check your Dialog Boxes

The Windows Interface Guidelines for Software Design. Microsoft Press, 1995
Chapter 7 - Menus, Control, and Toolbars

Useful Web Sites

The Microsoft Developers Network Online Library of Books, specifically the online version of *The Windows Interface Guidelines for Software Design*

http://msdn.microsoft.com/isapi/msdnlib.idc?theURL=/library/books/winguide/PLATFRM2/D5/S115B5.HTM

MacIntosh Human Interface Guidelines Online version Chapter 7 - Controls
http://developer.apple.com/techpubs/mac/HIGuidelines/HIGuidelines-144.html#HEADING144-0

Isys Information Architects Interface Hall of Shame and Interface Hall of Fame. These sites are updated regularly to show examples of the best and worst in windows interface design.
http://www.iarchitect.com/msoft.htm

Jakob Neilsen's online article titled '*The Difference Between Web Design and GUI Design"*.
http://www.useit.com/alertbox/9705a.html