

Windows Section	
Overview	Page 1
Windows Definitions and Terms	Page 1
General Windows Standards	Page 3
General Windows Guidelines	Page 9
Recommended Readings	Page 10
Helpful Web Sites	Page 10

Windows

Overview

Windows allow the developer to implement a user interface in several different ways. Keeping the layout simple and consistent are keys to good interface design. Simplicity lies in minimizing the number of windows and keeping the features of those windows consistent.

Each added window burdens the user with complex window management. The greater the number of windows, the more confusing the interface is to the user. Things get visually crowded and navigationally confusing. Alan Cooper in his book, *About Face: The Essentials of User Interface Design*, calls this phenomenon **window pollution**. Don't add windows to the user interface unless they have a special purpose that can't be served by an existing window.

The standards and guidelines contained in this section do not include recommendations for HTML specific standards and guidelines since these windows tend to be controlled by the browser. However, in those instances where an application generates secondary windows, the standards and guidelines outlined here should be followed.

Windows Definitions and Terms

Developers must choose windows carefully and understand why some types of windows work better than other types. Making the right choices means understanding the types of windows.

The definitions of the primary and secondary windows and their types presented here are intended to familiarize the developer with commonly used industry terms and to better understand their role in designing consistent window interfaces. These definitions, particularly related to primary windows, are not intended to serve as recommendations regarding which type to use. Recommendations on selecting window types will be addressed in the Navigation and Metaphors sections.

Primary Windows

A primary window is the application's main window from which all primary commands are initiated. In most applications there is a single primary window. However, in MDI style applications, parent and child windows are considered to be primary windows.

Single Document Interface (SDI)

An SDI is a single primary window that usually contains a menu bar, toolbar and status bar. This provides a one-to-one relationship between the object and the window. Opening an SDI puts the window at the top of all other current windows and places an entry on the task bar. Examples include Microsoft Notepad and WordPad.

Multiple Document Interface (MDI)

An MDI is a collection of related object windows or *child windows* within a single frame or *parent window*. The parent window almost always contains a menu bar, toolbar and status bar. MDI style applications can be used to present multiple views of the same data or multiple views of related data. Each child window shares interface elements on the parent window. This approach is suitable for large applications where users need the ability to multi-task and have the skills to manage all of the windows. Examples include Microsoft Word, Excel, and Visual Basic.

Split Document Interface (split SDI or Explorer)

A split SDI is a single primary window that has a window splitter, with the option of a menu bar, toolbar, and status bar. The splitter provides for the ability to add a second view of an object or, have a tree view or list view to control the contents that are in each side of the window. This approach works well when the user perceives objects to be in a hierarchical order. Examples include Windows Explorer and Outlook.

Secondary Windows

A secondary window is used to provide supplemental information or interaction to a primary window. Interfaces are simpler with fewer secondary windows.

Modal dialog boxes

A secondary window used to collect additional information related to the primary window. The user must complete or dismiss the dialog before being allowed to return to the primary window.

Modeless dialog boxes

A secondary window typically used to interact with the primary window. The user may change focus between the secondary window and the primary window at will.

Property sheet

A property sheet is a modeless secondary window that displays user-accessible properties of some object. These properties are viewable but not necessarily editable. Property sheets typically have tabs to organize the property information.

General Windows Standards

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



Windows will be designed to fit a screen resolution of 800w x 600h

CDC developers will design their windows to fit within a minimum screen resolution of 800 x 600. Users, however, must maintain the ability to set their screen resolution to any size.



CDC applications will be maximized by default upon opening for the first time. Subsequent openings will return the primary window to the size previously determined by the user

The application will “remember” the users last setting and return to it upon opening the application so as to provide the user with control over the interaction.



Secondary windows will be no larger than 7380w x 6390h (twip measurement)

This maximum size will keep windows from becoming too large at some screen resolutions, but still provide enough space to display the supplemental information. Note: A *twip* is defined as screen independent unit used to ensure that placement and proportion of screen elements in the screen application are the same on all screens.



Size measurements (in pixels) will be standard for the following types of secondary windows:

<i>About Box</i>	<i>390w x 264h</i>
<i>Login</i>	<i>290w x 142h</i>
<i>Splash Screen</i>	<i>500w x 310h</i>

Some development tools provide wizards that can be used to develop these forms. If wizards are not provided, the 3 secondary window types will follow the sizes indicated above.



All windows will follow standard rules of visual affordance as follows:

Raised items can be clicked

Items highlighted when mouse cursor passes over can be clicked

Recessed items cannot be clicked

Items with a white background and cursor placement can be edited

Items with a gray background cannot be edited

Gray items are disabled

Raised lines can be dragged

Affordance relates to the ability of a user to determine how to use an object by looking at its visual clues. This concept is shown in the figure titled *Visual Affordance Examples*. Users understand affordance through their real world knowledge. Follow these standards to avoid conflicting with a user's real world knowledge.

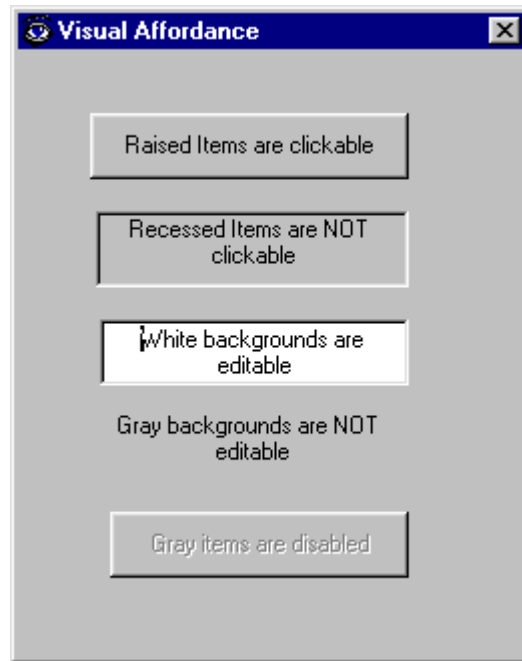


Figure 1 - Visual Affordance Examples

S ***Only primary windows will have title bar icons, menu bars, toolbars and status bars***

Title bar icons are used as a visual distinction between primary and secondary windows and therefore, should only appear on primary windows. This title bar icon and text is what appears in the taskbar. Secondary windows will not appear on the taskbar because clicking on a primary window activates any secondary window. In an MDI style application, child windows are considered to be primary windows and may have application icons to identify the name of the object being viewed. However, child windows will not have menus, toolbars or status bars.

S ***CDC application title bars will contain the following components in the order specified here:***

CDC System Icon

CDC System Name Text

Data Type Name (for MDI applications only)

Title bars identify what the window is viewing. It also is the control point for moving and resizing the window. A primary window of an SDI style application will include the application (CDC system) icon and the name of the application as

shown in the figure titled *Primary Window Title Bar Example*. Secondary windows will not contain an application icon.

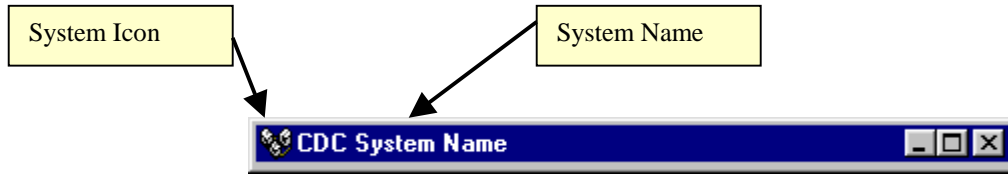


Figure 2 - Primary Window Title Bar Example

If the application is an MDI style, the title bar of the parent window will follow the same format as the primary window, as shown in the figure titled *MDI Title Bars Example*. All child window title bars will contain an icon that represents the data that is being displayed and a description of that data. In our example, if there were multiple patients that could be displayed in this application at the same time, the data name should include the name of the patient.

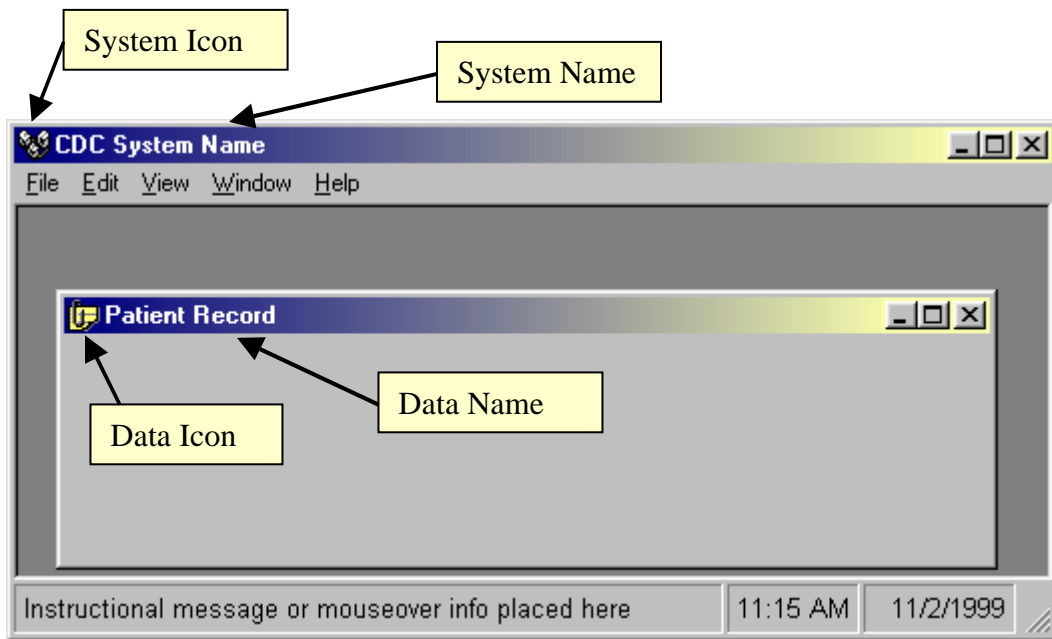


Figure 3 - MDI Title Bars Example

If the child window is maximized within the MDI parent, the child title bar is hidden and merged with the parent window as shown in the figure titled *MDI Merged Title Bar Example*.

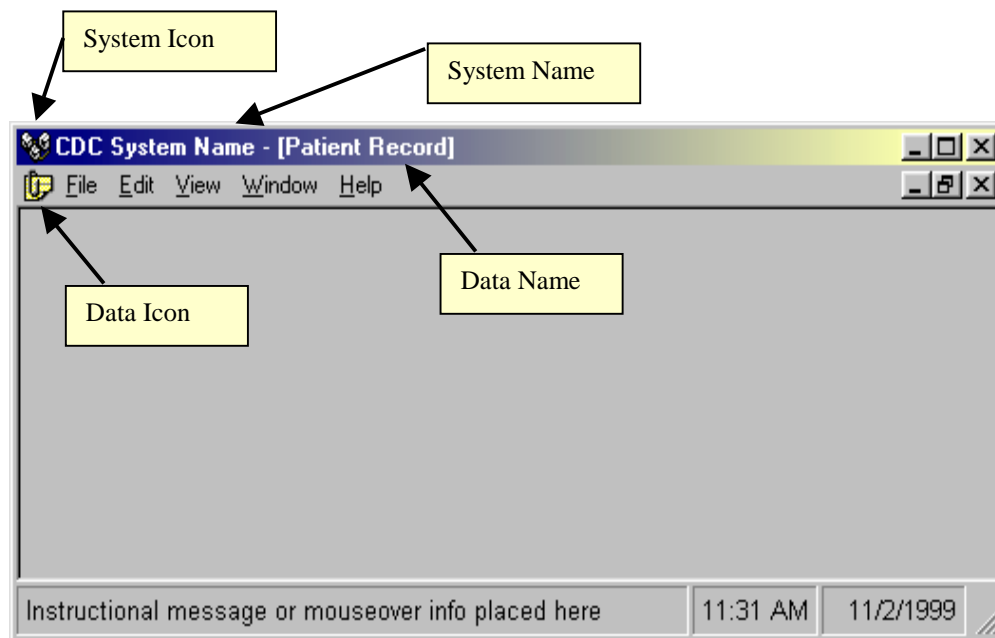


Figure 4 - MDI Merged Title Bar Example



Secondary windows cannot be resized

Secondary windows will have fixed borders and therefore may not be resized. Since the window is not resizable, the minimum and maximum buttons will not be contained in the title bar. Secondary windows will only contain a control box in the title bar.



Primary windows will contain a status bar

All primary windows will contain a status bar in the bottom of the window. The status bar will contain multiple panels. The left panel will be used for instructional text or mouse/rollover text. The far right two panels will contain the system date and time as illustrated in the figure titled *Status Bar Standard*.

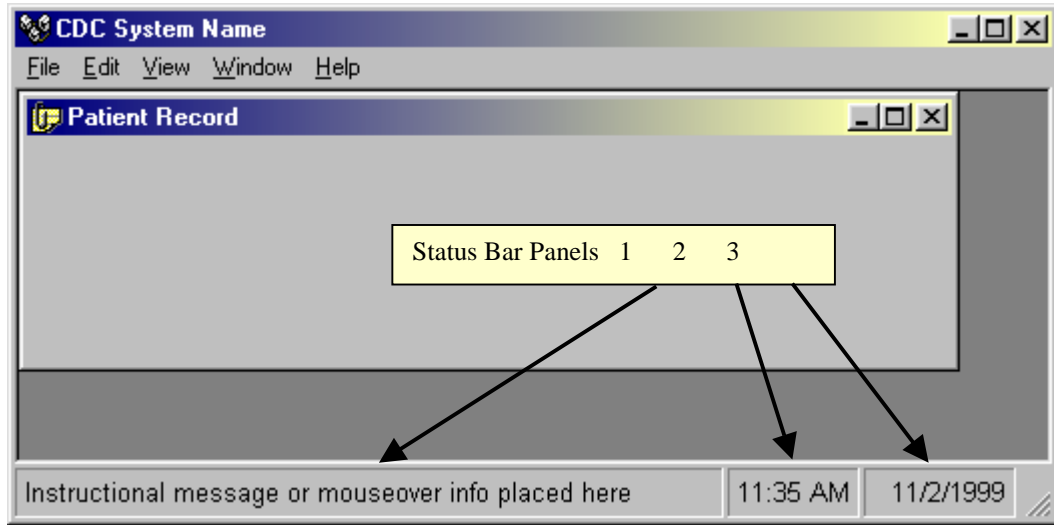


Figure 5 - Status Bar Standard



Use common dialog box interfaces

The Windows platform provides pre-built interfaces for many common operations. These interfaces provide consistency and will allow users to more easily transfer their knowledge. The table titled *Windows Dialogs* lists some common dialog interfaces to be used by the software developer in the Windows environment.

Table 1 - Windows Dialogs

COMMON WINDOWS DIALOGS	Open Save As Find Replace Print Print Setup Page Setup Font Color
---------------------------------------	---



Controls will be spaced within the window according to CDC standards:

Between controls - 7 pixels.

Margins - 12.5 pixels

CDC software applications will maintain consistent margins and spacing between controls and groups of controls. All controls will have 4 dialog box units (DLU) between them, unless you want to align a set of related controls. Four DLU's are approximately 7 pixels. Maintain a consistent margin from the edge of the window of 7 DLU that is approximately 12.5 pixels. The CDC standards for spacing and margins are illustrated in the figure titled *Control Layout and Spacing*.

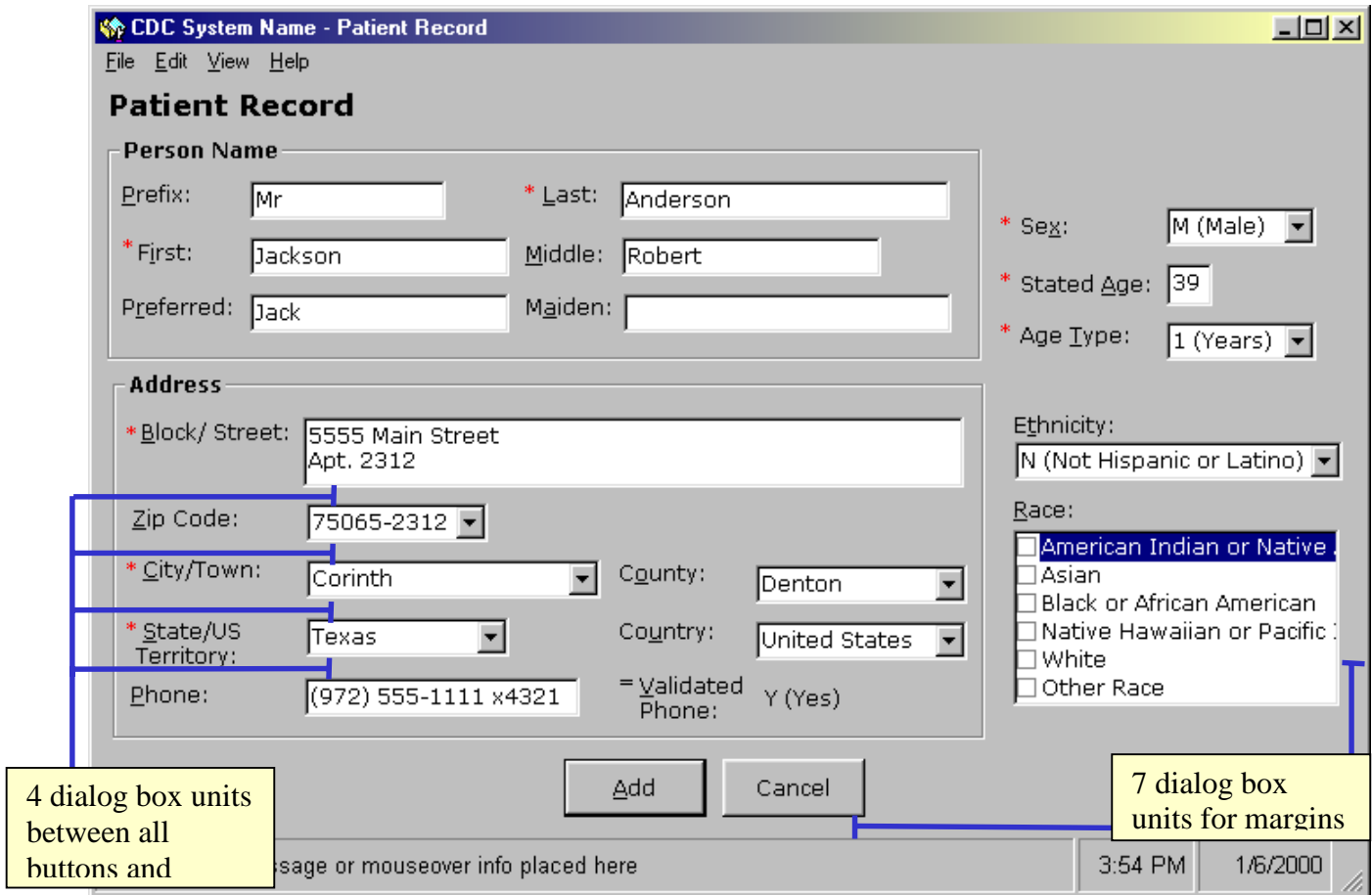


Figure 6 - Control Layout and Spacing



Avoid horizontal scrolling windows

Break information contained in horizontal scrolling windows into multiple windows, or allow the user to expand or collapse to show only some information at a time.

General Windows Guidelines

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

Follow a consistent arrangement of information in the window

CDC applications should conform to the Western culture way of reading in which people read from left to right and from top to bottom. Therefore, place the more important information on top and to the left in the window.

Use expanding dialog boxes instead of cascading dialogs when possible

Expanding dialog boxes allow the user to understand the flow of information in the application. Complex cascades of dialog boxes, however, can be confusing to the beginning user because even visual clues, such as the disabled caption bar, may not provide enough direction to understand their modality.

Avoid “Always On Top” behavior

Avoid having secondary windows appear on top of another application's primary window (or on top of the other application's dependent secondary windows) when the user activates a window in that application. “Always On Top” behavior should be a feature chosen by the user rather than a system default.

Avoid using application and system modal windows

Two types of modes, *application modal* and *system modal*, are often available to developers. In the application modal, users can not access any other part of the interface until the current window is closed. Application modal should be avoided unless additional information is required to complete a command or when it is important to prevent any further interaction until satisfying a condition. System Modal, which should never be used by CDC software applications, occurs when users are not allowed to work anywhere else on the computer until the current window is closed or until the application ends.

Size secondary windows appropriately

Secondary windows, other than those listed in the standards, are meant to complement the primary window to which they belong. Make the secondary window only as big as is necessary to hold the supplemental information. Secondary windows should be displayed within the borders of the primary window, preferably center to it, unless there will be multiple cascading secondary windows.

Recommended Readings

Cooper, Alan. *About Face: The Essentials of User Interface Design*. IDG Books Worldwide, Inc., 1995

Chapter 7 - Windows-With-a-Small-w

McKay, Everett N. *Developing User Interfaces for Microsoft Windows*. Microsoft Press, 1999

Chapter 10 - Good User Interfaces are Visible

Chapter 15 - Keep it Simple

The Windows Interface Guidelines for Software Design. Microsoft Press, 1995

Chapter 6 - Windows

Chapter 8 - Secondary Windows

Chapter 13 - Visual Design

Helpful 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>