JOINT COMPUTER-AIDED ACQUISITION AND LOGISTIC SUPPORT (JCALS) SYSTEM

PC CLIENT VERSION 2
HUMAN COMPUTER INTERFACE (HCI) STYLE GUIDE FOR MICROSOFT WINDOWS

CSC DOCUMENT CONTROL NO. 4401PCCL.3

CONTRACT NO. DAHC94-89-C-0008

14 October 1996

JCALS

Prepared for:

U.S. Army Information Systems Selection and Acquisition Agency
Alexandria, VA

U.S. Army
Program Manager, JCALS
Fort Monmouth, NJ

Prepared by:

Computer Sciences Corporation
Integrated Systems Division
304 West State Highway 38
Moorestown, NJ 08057

JOINT COMPUTER-AIDED ACQUISITION AND LOGISTIC SUPPORT (JCALS) SYSTEM

PC CLIENT VERSION 2 HUMAN COMPUTER INTERFACE (HCI) STYLE GUIDE FOR MICROSOFT WINDOWS

CSC DOCUMENT CONTROL NO. 4401PCCL.3

CONTRACT NO. DAHC94-89-C-0008

14 October 1996

JCALS

Prepared for:

U.S. Army Information Systems
Selection and Acquisition Agency
Alexandria, VA

U.S. Army
Program Manager, JCALS
Fort Monmouth, NJ

Prepared by:

Computer Sciences Corporation
Integrated Systems Division
304 West State Highway 38
Moorestown, NJ 08057

PREFACE

For specific questions or comments about topics covered in this guide or other issues related to a Microsoft Windows user interface, contact Jo Hall, Helene Iavecchia, or Steve Rabeler at:

CSC Integrated Systems Division
P.O. Box 1038
304 West Route 38, Mail Code 54
Moorestown, New Jersey 08057
Telephone: (609) 983-4400
Fax: (609) 988-8410

TABLE OF CONTENTS

LIST OF FIGURES

1. PURPOSE

The Computer Sciences Corporation (CSC) Microsoft Windows Human-Computer Interface (HCI) Style Guide establishes standards for the basic appearance and behavior of a Microsoft Windows user interface. The Style Guide is intended to serve as a guidance document for CSC, subcontractors, and vendors during design and development of the user interface.

In developing this Microsoft Windows Style Guide, standards and guidelines from the following were reviewed and incorporated:

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

- Department of Defense HCI Style Guide, Technical Architecture Framework for Information Management: Volume 8, Defense Information System Agency, 1994.

This Style Guide addresses only the style and format of user interface design. Adherence to these guidelines alone does not ensure a usable system. Developing a high quality user interface requires input from a multidisciplinary team that includes requirements, software, and HCI design specialists. Further, the task of building the user interface is ideally performed by a small number of specialists who are knowledgeable about the standards and can ensure skilled application of them. Developing standards is only one step in a development process that should include task analysis, prototyping, and usability assessment.

2. HCI DESIGN GOALS AND PRINCIPLES

2.1 Be Consistent

Each application must be consistent within itself and with the other applications that share the environment.

Consistency across applications is a key factor in designing usable systems because it makes the interface familiar and predictable. It allows the user to transfer knowledge from one task to another so that new tasks can be learned more quickly. Consistency also enables standardized, lower cost training by eliminating idiosyncratic instructions for each application.
Key guidelines for achieving consistency are:

- A given action should always produce the same result. For example, the “OK” button should always initiate an action and close the window, while the “Apply” button should initiate an action but leave the window open.

- A fixed set of terms and formats should be selected prior to system design and applied consistently during development. For example, titles and menu options should have a verb-noun format.

- Metaphors, frameworks based on everyday experience, should be used consistently across applications. For example, a print action should be consistently represented in a toolbar by a printer icon.

  See Sections 3.3 and 3.4 for more details on consistency.

2.2 Anticipate the User Skill Level

Match the interface design to users' level of computer experience.

- The anticipated computer experience level of most PC users ranges from novice to intermediate level.

- The user interface must provide flexible strategies that accommodate increasing skill levels as users become more experienced with the system. For example, allow users to choose the icon buttons that are presented in the tool bar.

2.3 Consider Operational Conventions

Match the users' business processes or applicable DoD standards to the terminology and data presentation.

- Menu options should be ordered according to the typical business process.

- Provide unobtrusive cues on what step to perform next in the process, or provide on-line help with step-by-step instructions for performing a process.

2.4 Be Flexible

Provide flexible strategies for performing operations and allow the user to configure the system according to preferences.

- Allow the user to access a frequently used action at several points. For example, provide a menu option to access an application in many windows instead of only one window.

- Provide alternate ways to access a frequently used menu option. For example, a menu option could be accessed through a drop-down menu, an access key, or a keyboard accelerator.

  See Section 7 for details on access keys, shortcut keys, and function keys.

- Allow the user to configure settings and select personal preferences.

3. USER INTERACTION APPROACHES

3.1 Emphasize Use of the Direct Manipulation Model

Use point and click as the primary means to interact with the system.

- Microsoft Windows provides many objects that can be used for direct manipulations (e.g., icons, scroll bars, sliders, buttons).
- Pointing and clicking lets the user interact with an application in a manner that represents real world experiences where tools are used to perform tasks.
- Direct manipulation provides immediate visual feedback for user actions (e.g., the selected object is highlighted).
- Direct manipulation prevents serious errors, encourages exploration, and has high subjective satisfaction.
- Direct manipulation is appropriate for users of any experience level. It is easy for novices to learn and retain while not being a burden on experienced users.

Rather than typing in entries, allow the user to select from a list of available choices (e.g., use combo boxes and spin boxes). This eliminates the need for software developers to create validation routines.

3.2 Consider Other Interaction Styles

While direct manipulation should be the primary means of interaction, text entry and command language may be suitable in some circumstances.

- Use text entry when the system cannot present a list of choices. The disadvantages of text entry are that the user must understand permissible entry values, and that the developer may need to develop validation routines. In addition, training may be necessary in order to use this interaction style.
- Text entry interaction is most appropriate for frequent or knowledgeable users.
- Use command language as an alternate or secondary means of interaction for the expert user in a very specialized functional area.

3.3 Ensure Consistent Interaction

Each mouse action should have only one meaning and produce the same result every time.

- For example, double clicking should always result in simultaneously selecting and opening an object. Double clicking should never carry a dual meaning of opening an object in one context and deleting an object in another context.

Each menu option should have only one meaning and produce the same result each time.

- For example, the “File > Save As” menu option should always present the same window to enter a new file name and location.

  See Section 2.1 for more information on consistency.

3.4 Ensure Consistent Feedback

Every user action should consistently produce a visible or auditory response from the computer.

- As a corollary, lack of a system response is never an appropriate response to a user action. For example, if a database search returns no values, present a message box to inform the user that no items were found.

  See Section 2.1 for more information on consistency.

4. OBJECT-ACTION SELECTION TECHNIQUES

4.1 Object-Action Model

The user should first select the object to manipulate, then select the command.

- For example, the user should select an item in a list before selecting a Delete command.

4.2 Mouse Buttons

The left mouse button should be used as the primary button.

- Mapping of all selection operations to the left button is recommended to ensure compatibility with systems using a single button mouse.

Do not assign any actions to the middle button.

- Due to a diversity of hardware in the fielded system, it cannot be guaranteed that every user will have a three-mouse button.

The rightmost button can be programmed to present a context sensitive pop-up window.

- Pop-up menu options should always be redundantly available on the main menu.
- For example, when the user selects a file, the rightmost mouse button can present a pop-up containing options relevant to the selected object (e.g., Open, Copy, Move).
- The rightmost button can also be programmed to present context-sensitive help on an object.

4.3 Highlighting a Selection

A selected object should be visually distinguishable from nonselected objects.

- Highlight a selected object using reverse video.
- When two lists are present in a window, allow a highlighted item in only one of the lists at a time.

4.4 Single Click

Use a single click to select an object.

- Use only the left mouse button for object selection.
- Single clicking should always result in an object selection, and carry no other meaning.

See Section 3.3 for more information on mouse actions.

4.5 Double Click

Use a double click to simultaneously select an object and start an action.

- Use only the left mouse button for double clicking an object.

- The resulting action depends on the object that is double clicked, but typically should be an open or launch operation.
- When a directory is double clicked, the action should be to display the files in that directory.
- When a file is double clicked, the action should be to open the file in an application.
- When an application is double clicked, the action should be to open that application.
- Double clicking should never result in a destructive action.

5. THE CURSOR

5.1 Cursor Placement

Keep the cursor in place until it is moved by the user with the mouse or a keyboard action.

- When a window opens, the cursor should appear in the top, leftmost field or selectable object.

5.2 Cursor Shape

Change the cursor shape to provide a visual cue of available functionality or acknowledgment of user inputs.

- As a user moves the cursor across the screen, its appearance should change to provide feedback about a particular location, operation, or state.

- Default cursor shapes are established when Microsoft Windows is installed, and the user can reconfigure them according to personal preferences. Therefore, they are not specified in this style guide.

6. NAVIGATING FROM THE KEYBOARD

6.1 Usage

Provide an alternative to mouse selection of controls by allowing users to select controls with the tab and arrow keys.

- Allow keyboard navigation for all controls including menus, command buttons, option buttons (radio buttons), check boxes, text fields, combo boxes, spin boxes, sliders, and lists.
- Use the TAB key to navigate between controls. Tab navigates forward; Shift+Tab navigates backwards.
- Use the Up Arrow and Down Arrow keys to navigate among and select choices within a control. For example, when an arrow key navigates to the next choice within a list, combo box, spin box, or a group of option buttons, that choice should be activated.
- Pressing Enter should always invoke the default command button in a window, if one is defined. Note: The default button has a double border.

6.2 Keyboard Navigation Sequence

- Set the keyboard navigation sequence through controls from left-to-right and top-to-bottom.

7. ACCESS KEYS, SHORTCUT KEYS, AND FUNCTION KEYS

7.1 Access Keys (Mnemonics)

Each menu title, menu option, and command button should have an access key, that is, a single keyboard character for selecting the option. An access key is represented by an underlined character in the label.

- The preferred mnemonic is the **initial letter** of the label. The second choice is a **distinctive consonant** in the label. The last choice is a **vowel** in the label.

- Access keys must be case insensitive for activation.
- Use access keys consistently across applications for the same menu title, option, or button.
- Do not assign an access key to the **OK** and **Cancel** buttons if they are mapped to the Enter and Escape keys.

  See Section 22.8 for default access keys.

7.2 Shortcut Keys (Accelerators)

Frequently used menu options should have keyboard shortcut keys. A shortcut key is a combination of a control key and a letter key that is represented as follows: Ctrl+C for Cut.

- Shortcut keys should appear right justified within a menu, next to the corresponding option name.
- Use shortcut keys consistently across applications for the same menu option.

See Section 22.8 for default shortcut keys.

7.3 Function Keys

Minimize the use of function keys (such as F5 or F10) due to the memory burden they impose on the user.

- Use function keys consistently across applications for the same action. For example, the F1 key should be used to activate context sensitive Help on a window object.

8. SYSTEM STATUS INDICATORS

8.1 Provide Feedback when an Action is in Progress

Provide an hourglass or a message box for any action expected to take longer than 2 seconds to complete.

- For actions that take **between 2 and 5 seconds**, use an hourglass cursor as a visual indicator that the action is in progress.
- For actions that take **longer than 5 seconds** and **must be completed** before the user can proceed, display a progress bar in a message box rather than an hourglass.
- For actions that take **longer than 5 seconds** and execute as **background processes**, display an hourglass for 2 seconds, present a progress bar in a message box, and allow the user to regain control of the mouse.

8.2 Provide Feedback when a Lengthy Process Completes

For print operations or other off-line background processes, provide feedback when the process is completed.

- When the process is completed, present a message box or E-mail message informing the user about the status.

9. MODALITY

9.1 Application Modes

Avoid the use of application modes.

A mode occurs when the application is locked in a state that limits the user to specific interactions.   The vi editor is a good example of such an application; in order to insert text, the user must first specify the insert mode by typing “i.”

Modality contrasts with an object oriented user interface and opposes in principle the Object-Action Selection model (see Section 4.1). Modality also places a burden on the user to remember the current mode or state.

- Where modality cannot be avoided, clearly indicate the current mode (e.g., present a distinctive cursor shape).

9.2 Modality in Secondary Windows

- Use a modeless secondary window when the user needs to interact with either the secondary or primary window. For example, a search and replace operation typically uses a modeless secondary window.

- Use a modal secondary window when the user must complete an action and close the secondary window before the process in the primary window can proceed. For example, in a file open operation, the user must select the file in a modal secondary window.

10. ERROR PREVENTION

10.1 Confirm Destructive Actions

Display a message box requesting confirmation of destructive actions that have irreversible consequences.

- For example, any action that would result in deletion of stored data requires user confirmation prior to execution.
- An exception to this rule would be when there is an easy undo command available to reverse the user action.

See Sections 11.1 and 31 for details on warning message boxes.

10.2 Hiding or Graying Window Objects

When a window object is not authorized (i.e., the user lacks the special access privileges to perform that function), gray it out.

- For example, gray the "Options > Approve" menu option if the user does not have manager approval authority.

When a window object is temporarily unavailable until a precondition is satisfied, gray it out until the precondition is met.

- For example, gray the "Edit > Copy" menu option until the user selects an object to copy.

For a window that has more than one state (e.g., Edit and View Only), gray out any window objects that are never available in a given state. (An exception is menus; see below. Another exception for buttons is discussed in Section 20.6.)

- For example, if a window has both an Edit and a View Only state, gray the "Edit" button in the View Only state.

- On-line Help should state the specific conditions in which objects are and are not available.

For a window that has more than one state (e.g., Edit and View Only), hide any <u>menu options</u> that are never available in a given state.

- For example, if a window has both an Edit and a View Only state, hide the "File > Save" menu option in the View Only state.

- Hide menu options before the window opens.

- Do not hide menu options if the state changes after a window is opened (e.g., Edit changes to View Only after saving). Instead, gray them out, then hide them before the window opens again.

- On-line Help should state the specific conditions in which menu options are and are not available.

10.3 <u>Provide a Means to Cancel Inputs</u>

- An **Undo** button or menu option should cancel the previous action.

- A **Clear** button or menu option should blank out all user entries, and reset the window to the original default values. Clear should never close the window.

- A **Cancel** button should ignore all inputs and close the window. Note: Never present a **Cancel** button on a window that has an **Exit** or **Close**.

- The **Escape** key should act the same as the **Cancel** button.

- When an **Exit** or **Close** appears in a menu bar, a warning box should be presented prompting the user to save any data not previously saved. Note: Do <u>not</u> present the warning box if all data has been previously saved.

  See Section 11.1 about Warning message boxes.
  See Section 35 for use of Exit/Quit/Cancel/Close.

10.4 Allow Users to Select Choices

Design the user interface so that users can rely on recognition rather than recall.

- Allow users to select choices, for example, in a list or combo box, rather than forcing the user to type in an entry from memory.

10.5 Prefill Default Values

Anticipate and minimize required user actions by automatically prefilling default values.

- Prefill a text field or combo box with a default that is the most likely value.

- Highlight the most likely selection in a list.

10.6 Validate Data Entries

- Conduct data validity checks (e.g., format and range) immediately after the user exits a text field.

- Present a message box for an invalid value, stating the nature of the problem and guidance for entering a valid value.

  See Section 11 about error handling.

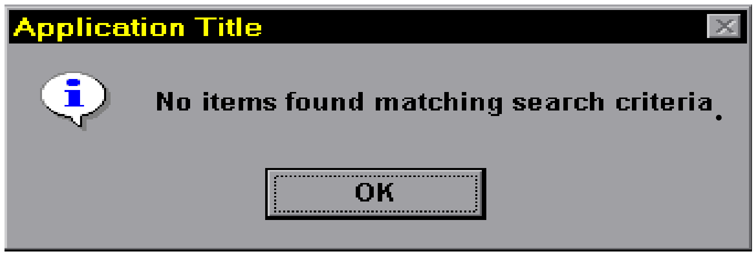
11. ERROR HANDLING

11.1 Use the Correct Message Box for Each Type of Error

Information: Use an Information message box for errors caused by the user. (See Figure 1.)

- An Information message box states the nature of the problem and how to recover.

- Only one button should be presented: the **OK** button.



Application Title

No items found matching search criteria.

Figure 1. Sample Information Message

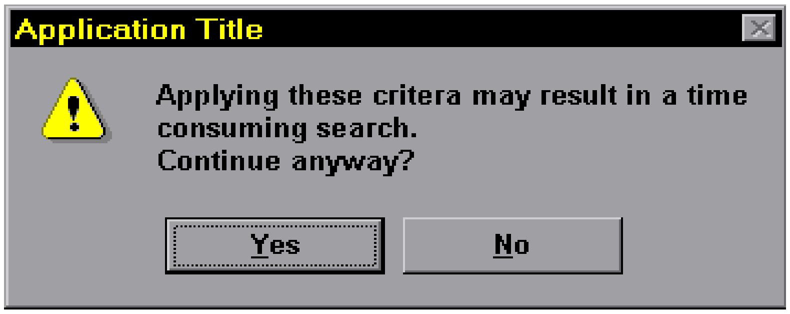
Warning: Use a Warning message box to advise the user of an undesirable condition and solicit a decision. (See Figure 2.)

- The Warning message box informs the user of a potentially dangerous or undesirable condition that has irreversible consequences, such as deleting data.

- The message should be stated in the form of a question. Phrase the question to permit a Yes/No answer. Work cannot proceed until it is answered.

- Two buttons should be presented: the **Yes** and **No** buttons.

- Rarely, a **Cancel** button may appear to the right. **Cancel** allows the user to back up one step. For example, the Warning "Do you want to save the data?" should provide a **Cancel** button that cancels the save operation entirely.


Application Title

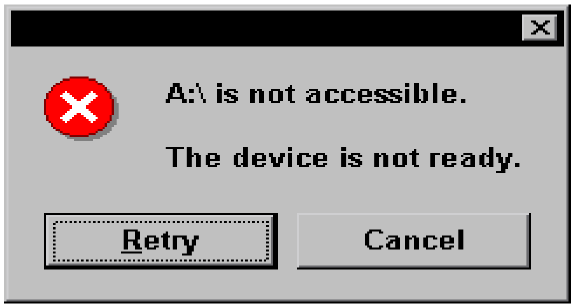
Applying these critera may result in a time consuming search.
Continue anyway?

Figure 2. Sample Warning Message Box

<u>**Critical**</u>**: Use a Critical message box to advise the user of a serious problem that needs correction before the work can continue.** (See Figure 3.)

- The Critical message box informs the user of a serious problem that requires intervention or correction before work can continue.

- Two combinations of buttons are permissible:
  Abort and **Retry**
  Retry and **Cancel**.


A:\ is not accessible.

The device is not ready.

Figure 3. Sample Critical Message Box

See Section 31 for details on message box design.

11.2 Make Messages Informative

- Make messages brief, not more than 2-3 lines.
- Communicate where the error occurred, the nature of the error, how to recover, and where to obtain help.

11.3 Use Neutral Wording

- Messages should not imply blame, personalize the computer, or attempt to make a message humorous.

  See Section 31.3 for more details on message wording.

11.4 Place the Cursor Where the Error Occurred

- Mark the location of the first detected error with the cursor.

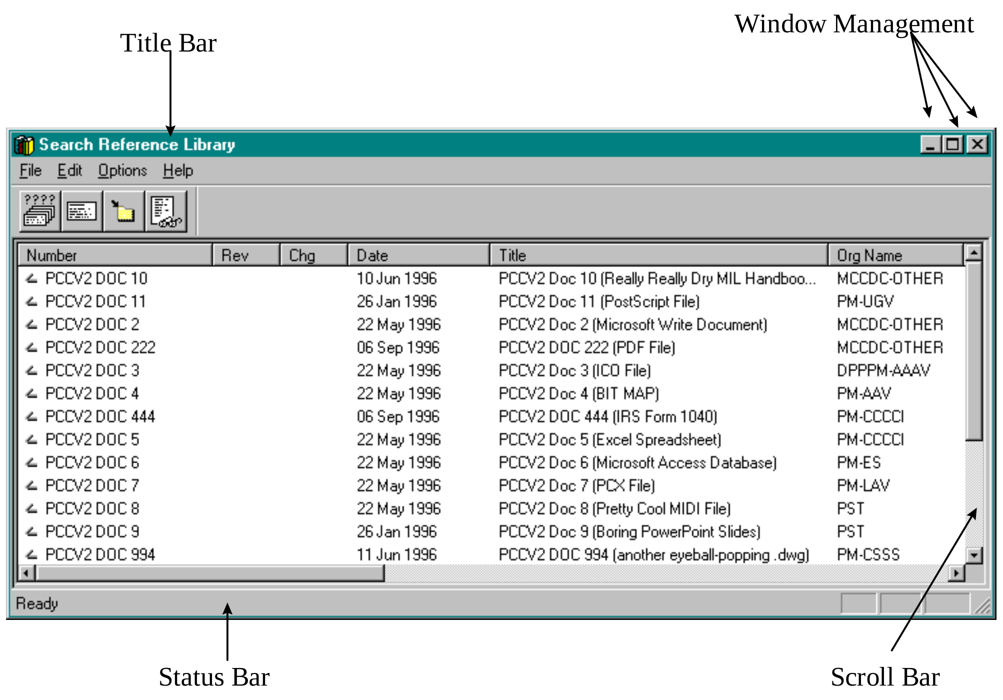
12. WINDOW LAYOUT

12.1 Usage

Use the standard Microsoft layouts for each window type.

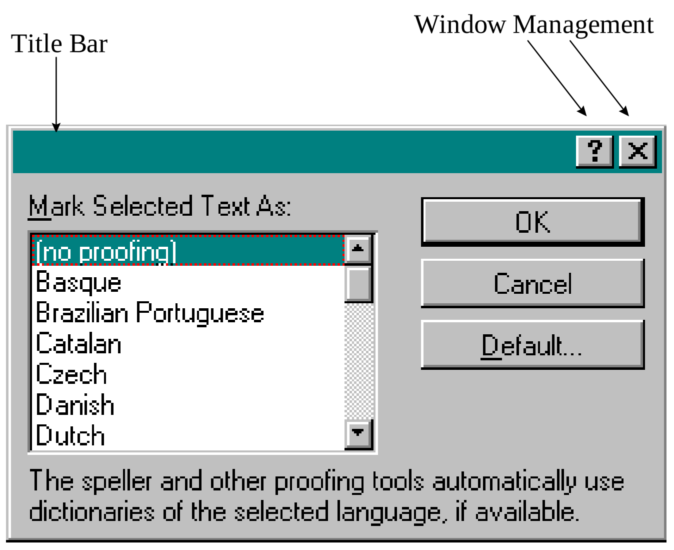
- Each application has a **primary window** where the primary work takes place. A primary window must have a frame and a title bar. It may also have menu bars, tool bars, a status bar, and scroll bars. Single document interfaces (SDI) and multiple document interfaces (MDI) are examples.
- Each application typically has multiple **secondary windows** that supplement the primary window and allow users to enter options, parameters, or more details. Dialog boxes, palettes, message boxes, and property sheets are examples.

- Figures 4 and 5 illustrate standard features of a primary and secondary window.



Number	Rev	Chg	Date	Title	Org Name
PCCV2 DOC 10			10 Jun 1996	PCCV2 Doc 10 (Really Really Dry MIL Handboo...	MCCDC-OTHER
PCCV2 DOC 11			26 Jan 1996	PCCV2 Doc 11 (PostScript File)	PM-UGV
PCCV2 DOC 2			22 May 1996	PCCV2 Doc 2 (Microsoft Write Document)	MCCDC-OTHER
PCCV2 DOC 222			06 Sep 1996	PCCV2 DOC 222 (PDF File)	MCCDC-OTHER
PCCV2 DOC 3			22 May 1996	PCCV2 Doc 3 (ICO File)	DPPPM-AAAV
PCCV2 DOC 4			22 May 1996	PCCV2 Doc 4 (BIT MAP)	PM-AAV
PCCV2 DOC 444			06 Sep 1996	PCCV2 DOC 444 (IRS Form 1040)	PM-CCCCI
PCCV2 DOC 5			22 May 1996	PCCV2 Doc 5 (Excel Spreadsheet)	PM-CCCCI
PCCV2 DOC 6			22 May 1996	PCCV2 Doc 6 (Microsoft Access Database)	PM-ES
PCCV2 DOC 7			22 May 1996	PCCV2 Doc 7 (PCX File)	PM-LAV
PCCV2 DOC 8			22 May 1996	PCCV2 Doc 8 (Pretty Cool MIDI File)	PST
PCCV2 DOC 9			26 Jan 1996	PCCV2 Doc 9 (Boring PowerPoint Slides)	PST
PCCV2 DOC 994			11 Jun 1996	PCCV2 DOC 994 (another eyeball-popping .dwg)	PM-CSSS

Figure 4. Common Primary Window Components



Mark Selected Text As:

[no proofing]
Basque
Brazilian Portuguese
Catalan
Czech
Danish
Dutch

The speller and other proofing tools automatically use dictionaries of the selected language, if available.

Figure 5. Common Secondary Window Components

12.2 Designing Primary Windows

The primary window should contain the most critical data, and give the "big picture" view.

- The primary window is the user's main control point for each application. Most of the work activity should take place in the primary window.
- A primary window is the only window from which an application can be shut down.
- Critical data should be on the primary window and not hidden in secondary windows.
- Do not design the primary window as a menu bar with a blank screen where no work can be performed. It wastes screen real estate.

12.3 Designing Window Controls

- A window should have either a menu bar or a row of buttons along the side or bottom. A window should never contain both a menu bar and a row of buttons.

12.4 Designing Secondary Windows

Design secondary windows with the following uses in mind:

- Dialog Box	Obtains additional information from the user. (See Figure 6.)
- Message Box	Informs the user about a particular state or condition. (See Figures 1, 2, and 3.)
- Palette Window	Displays a toolbar or color/pattern choices. (See Figure 7.)
- Property Sheet	Displays the properties of an object. (See Figure 8.)

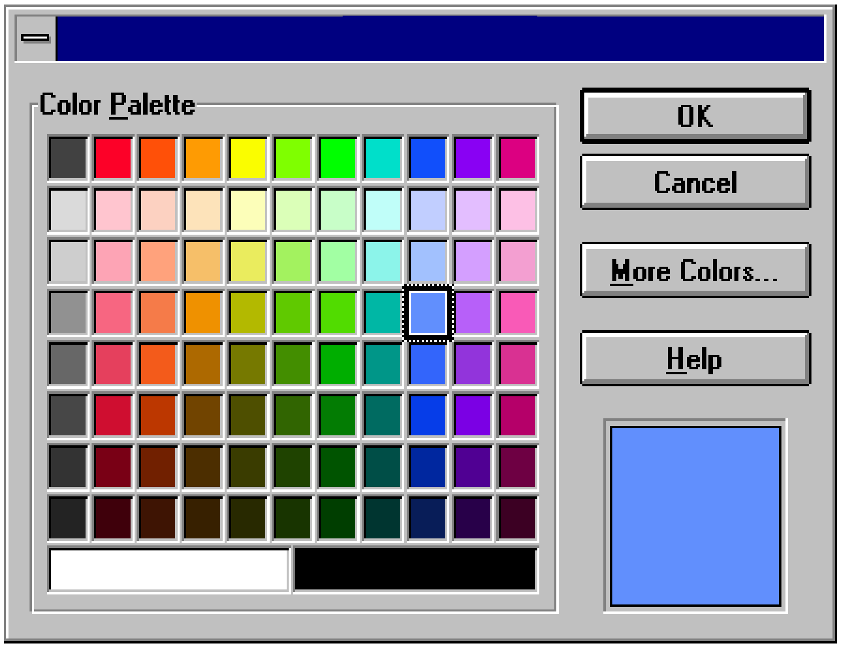


Figure 6. Sample Dialog Box

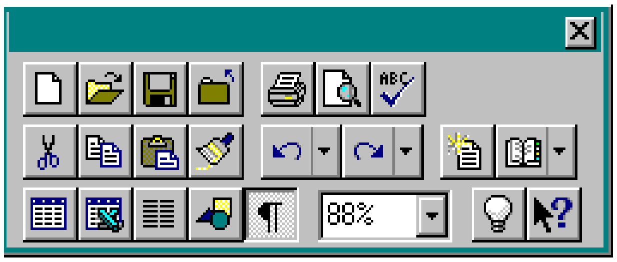
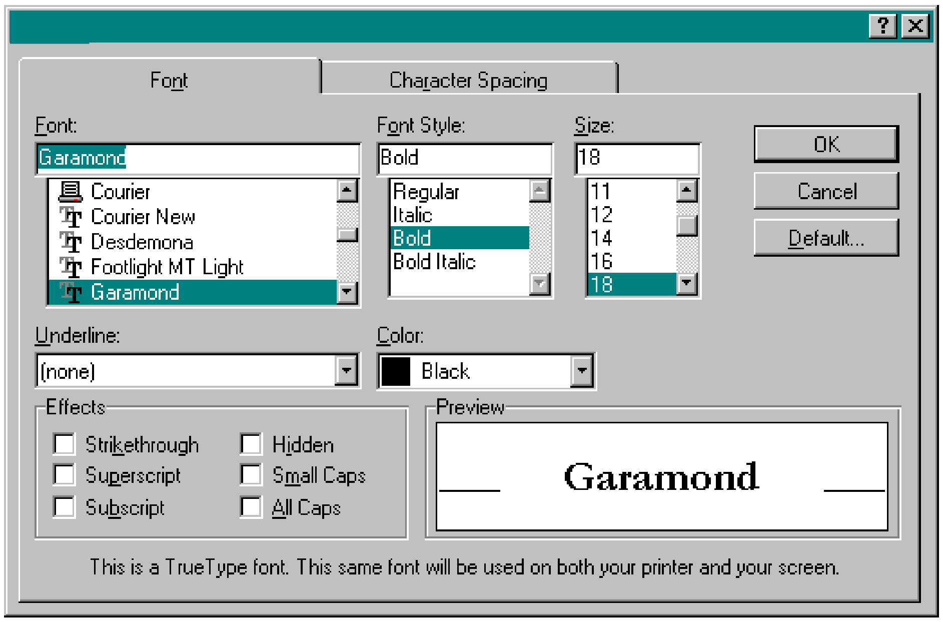


Figure 7. Sample Palette Window



Font: Garamond
Font Style: Bold
Size: 18
Underline: (none)
Color: Black

This is a TrueType font. This same font will be used on both your printer and your screen.

Figure 8. Sample Property Sheet

12.5 <u>Window Placement</u>

- All windows should be overlapping and moveable, not tiled.

12.6 <u>Window Title Bar</u>

Use a verb/noun format for the window title.

- The title bar text should communicate the purpose of the window, and serve as a prompt or provide instructions (e.g., “Select a Report”).

- Capitalize the first letter of each word in the title and use lowercase for the other letters (e.g., “Set Job Privileges”).

- The title of a secondary window should repeat the label of the control that opened it. For example if a secondary window opened after the “View” button was pressed, the title could state: “View Supply and Demand.”

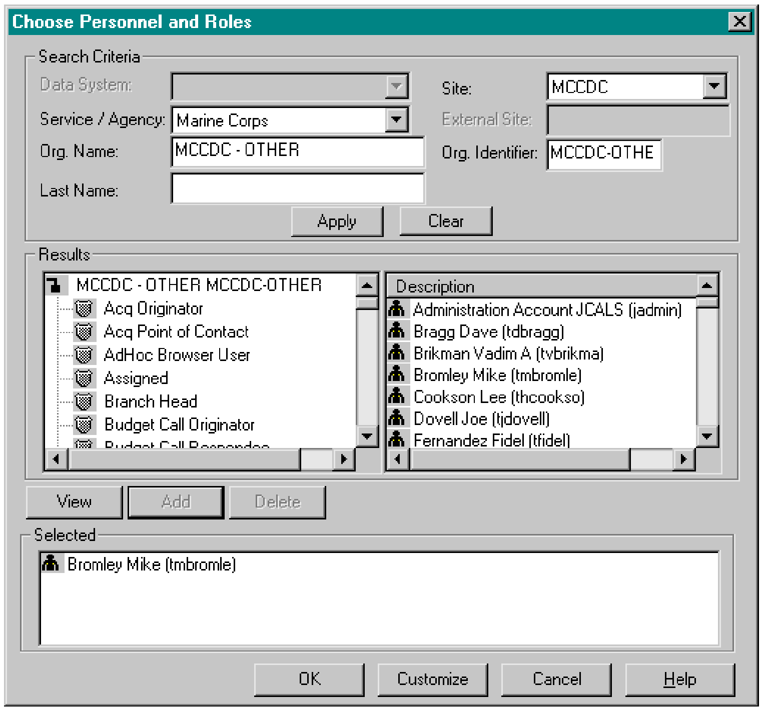
12.7 Arrangement of Window Objects

- Layout the window components according to the conventions of how information is read, that is, left-to-right and top-to-bottom.
- Position related items near each other.
- Use a separator line and header label or a group box to distinguish groups of related items.

12.8 Spacing of Window Objects

Avoid excessive white space. (See Figure 9.)

- Leave no more than 1/4 inch - 1/2 inch of white space around the edges of the window.
- Leave about 1/8 inch between objects.
- Align labels and data in even columns and rows.


Choose Personnel and Roles

Search Criteria
Data System:
Service / Agency: Marine Corps
Org. Name: MCCDC - OTHER
Last Name:
Site: MCCDC
External Site:
Org. Identifier: MCCDC-OTHE

Results
MCCDC - OTHER MCCDC-OTHER
Acq Originator
Acq Point of Contact
AdHoc Browser User
Assigned
Branch Head
Budget Call Originator

Description
Administration Account JCALS (jadmin)
Bragg Dave (tdbragg)
Brikman Vadim A (tvbrikma)
Bromley Mike (tmbromle)
Cookson Lee (thcookso)
Dovell Joe (tjdovell)
Fernandez Fidel (tfidel)

Selected
Bromley Mike (tmbromle)

Figure 9. Example of Aligning Labels and Data

13. WINDOW MANAGEMENT

13.1 Use the Standard Application Structures

Normally, windows within an application conform to one of the following Microsoft application structures:

- **Single Document Interface (SDI)**. Use an SDI when the application can be displayed using a single primary window and a number of secondary supplemental windows. (See Figure 10.)

- **Multiple Document Interface (MDI)**. In an MDI, a primary window serves as the parent window for a number of document (child) windows. The child windows appear only within the parent window. (See Figure 11.)

- **Workbook**. A workbook displays one primary window with tabbed pages. The user selects the tabs to move between windows of the workbook. The Workbook interface is useful for presenting data in an ordered form. An Excel spreadsheet is an example of a workbook.

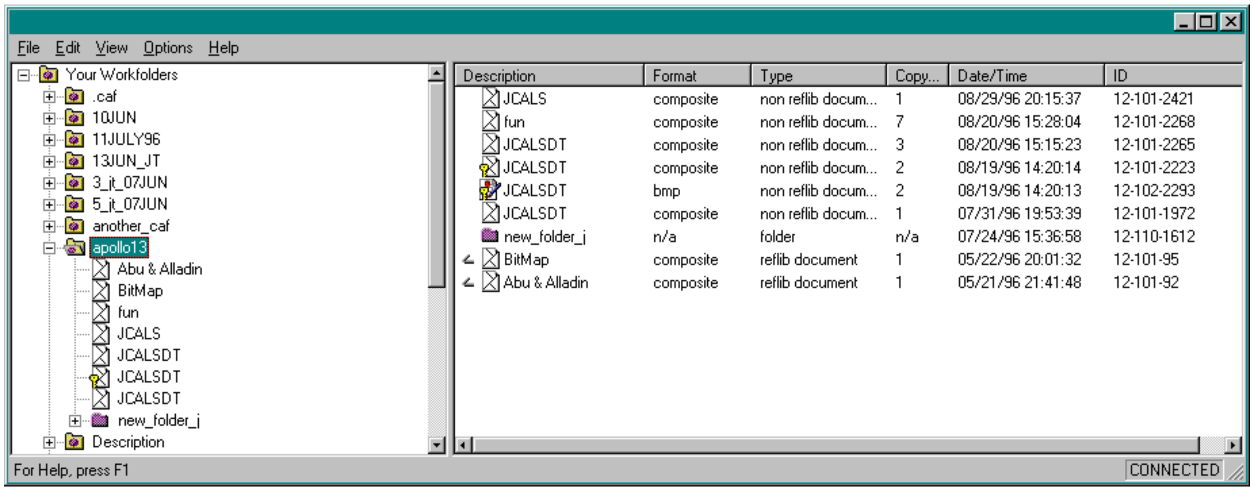
- **Workspace**. A workspace is a primary window that consists of a container that holds a set of icons. Within the parent workspace window, the icons can be opened as child windows. A graphical desktop is an example of a workspace.

- **Project**. A project is a parent window containing objects that can be opened as child windows. Unlike the workbook or workspace, the child windows do not share the menu bar of the parent.

- Note: In some circumstances, an application may be composed of only dialog boxes.

13.2 Use Explicit Focus

- Explicit focus allows the user to click anywhere in a window to bring it forward. Explicit focus is the default focus policy when Microsoft is installed.



File Edit View Options Help

Your Workfolders
- .caf
- 10JUN
- 11JULY96
- 13JUN_JT
- 3_jt_07JUN
- 5_jt_07JUN
- another_caf
- apollo13
  - Abu & Alladin
  - BitMap
  - fun
  - JCALS
  - JCALSDT
  - JCALSDT
  - JCALSDT
  - new_folder_j
- Description

Description	Format	Type	Copy...	Date/Time	ID
JCALS	composite	non reflib docum...	1	08/29/96 20:15:37	12-101-2421
fun	composite	non reflib docum...	7	08/20/96 15:28:04	12-101-2268
JCALSDT	composite	non reflib docum...	3	08/20/96 15:15:23	12-101-2265
JCALSDT	composite	non reflib docum...	2	08/19/96 14:20:14	12-101-2223
JCALSDT	bmp	non reflib docum...	2	08/19/96 14:20:13	12-102-2293
JCALSDT	composite	non reflib docum...	1	07/31/96 19:53:39	12-101-1972
new_folder_j	n/a	folder	n/a	07/24/96 15:36:58	12-110-1612
BitMap	composite	reflib document	1	05/22/96 20:01:32	12-101-95
Abu & Alladin	composite	reflib document	1	05/21/96 21:41:48	12-101-92

For Help, press F1 CONNECTED

Figure 10. Sample Single Document Interface Window

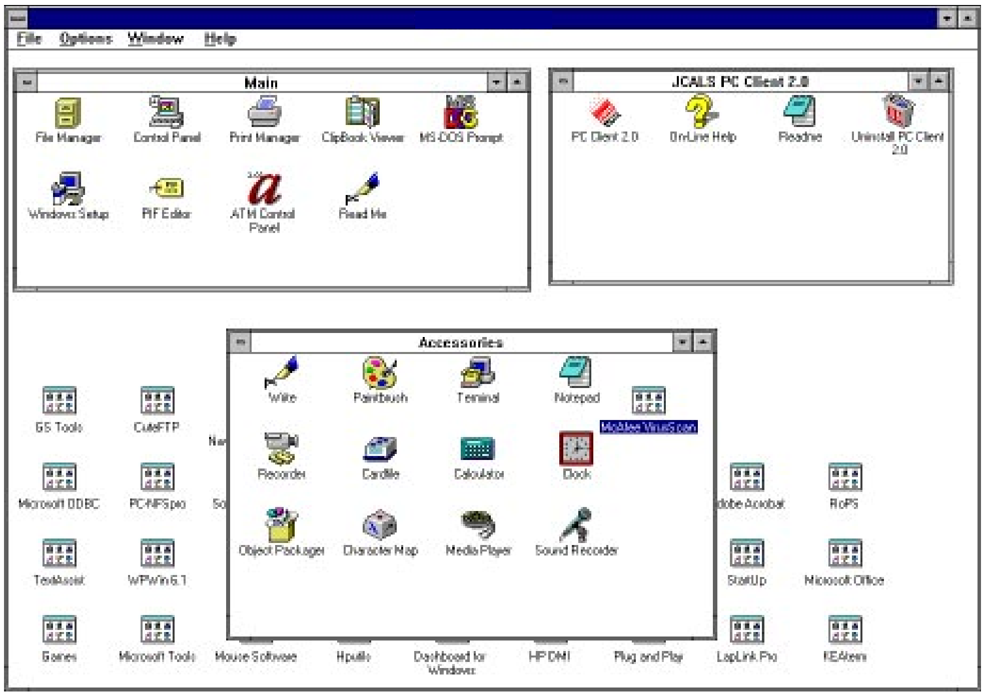


Figure 11. Sample Multiple Document Interface Window

13.3 Maximizing and Resizing Windows

- Set minimum sizes for all windows, so that the smallest size does not cut off critical information or controls.

- Do not set maximum height and width sizes on windows that contain resizable objects, such as lists or scrolled areas.

- Do not allow resizing on windows where an increased size will not allow the user to see increased information.

- All primary window must have a “minimize” button.

14. COLORS AND FONTS

Fonts and colors are established when Microsoft Windows is installed, and the user can reconfigure them according to personal preferences. Therefore, they are not specified in this style guide.

15. ABBREVIATIONS AND ACRONYMS

15.1 Avoid Abbreviations and Acronyms

- Only use abbreviations and acronyms when window space is seriously limited.

15.2 Use Accepted Terminology

- Develop a standard set of acronyms and abbreviations familiar to the user community, and use them consistently.

15.3 Provide a Cross Reference Table

- Provide an on-line cross reference in which abbreviations and acronyms are spelled out.

15.4 Usage in Selectable Lists

- In selectable lists (e.g., combo boxes and lists), present the acronym followed by its long name spelled out.

16. STATUS BAR

Use the status bar to present descriptive textual information about what is being displayed in the primary window.

- For example, the status bar can display descriptive messages about a selected menu or toolbar button.
- The status bar can be divided into multiple panes to display more information.

17. SCROLL BAR

17.1 Usage

Use scroll bars if the data exceeds the size of the visible area in the window, list, scrolled editable text field, or scrolled static text field.

- Provide a vertical scroll bar when the vertical length of the data exceeds the visible area.
- Provide a horizontal scroll bar when the horizontal width of the data exceeds the visible area.
- Provide both horizontal and vertical scroll bars when the data is too long and too wide for the area.
- Do not display a scroll bar until it can actually be used.

17.2 Sizing

- At least four lines of text should be visible.

- Make the window resizable to allow the user to view a larger text area.

18. CHECK BOXES

18.1 Usage

- Use check boxes when choices are not mutually exclusive.

18.2 Arrangement of Check Boxes

- Check boxes can be placed in columns or rows.

18.3 Default Selection

- Place a check in the default button to identify it to the user.

18.4 Presentation in a View Only Mode

- Disable (gray) any checkbox that is not selected.

- Do not disable (gray) a selected checkbox. However, do not accept user inputs to deselect a selected checkbox.

19. COMBO BOXES

19.1 Usage

Use combo boxes to allow the user to type in an entry or choose from a list.

- A **simple combo box** allows the user to type in an entry or choose from a list which is always visible.

- A **drop-down combo box** allows the user to type in an entry or choose from a list which first must be opened. Use this type to conserve window space.

- Note: A **drop-down list box** only allows the user to choose from a list. No type in is allowed.

  See Section 21.1 about drop-down list boxes.

19.2 Presentation in a View Only Mode

In a View Only window, present the current value and remove all other items in the list.

- Do not disable (gray) the combo box.

- For a simple combo box, only display the selected value.

- For a drop-down combo box, do not accept user input to open the list.

20. COMMAND BUTTONS (Push Buttons)

20.1 Use of Buttons Versus Menus

- When the total number of primary controls is less than six, present them in a row along the bottom or a column on the side. Note: Primary controls apply to the window as a whole (e.g., “OK”, “Cancel”, “Help”), not to buttons that apply solely to a list or other widget (e.g., “Add” and “Delete” buttons next to a list).

- When the total number of primary controls is greater than six, present them in menus rather than as buttons.

- A window should have either a menu bar or a row of buttons along the side or bottom. A window should never contain both.

20.2 Button Size

- Buttons can use either a text label or an icon as an identifier.
- Buttons with text labels should be 20 pixels high by 100 pixels wide. Only make buttons larger if absolutely necessary.
- Buttons with icon labels should have bitmaps that are 16 pixels high by 15 pixels wide.

20.3 Arrangement of Buttons

- Place primary buttons in a straight row along the bottom toward the right.
- Place primary buttons in a column along the right and toward the top only when necessitated by space constraints in the window.

20.4 Default Action

- The default action is generally the **OK** button presented on the lower left portion of the window.

20.5 Use of Case in Button Labels

- Button labels should appear upper and lowercase, for example, "Cancel," "Help," "Yes," with the exception of the "OK" button.
- The **OK** button should be all capitals.
- If a button label contains two words, make the first letter of each word uppercase (e.g., "List Frequencies").

20.6 Graying or Hiding Buttons

- When a button is not available or not authorized, gray it out.
- For a window that has more than one state (e.g., Edit and View Only), gray out any button that is never available in a given state.

Alternately, a button can be hidden rather than grayed, provided that it does not create an empty space between buttons.

- For example, a common window in an Edit state has an OK, Cancel, and Help button. In a View Only state for that window, hide the OK button and change the name of the Cancel button to Close.

See Section 34 on usage of OK and Apply buttons.
See Section 35 on Exit/Quit/Cancel/Close operations.

21. LIST BOXES

21.1 Usage

Use list boxes to display large numbers of selectable choices.

- Use a **list box** when you have enough available window space or need to display multiple columns.

- Use a **drop-down list box** when window space needs to be conserved.

- Use a **tree view control** to display information in an indented outline when the data logically fits into a hierarchical relationship. Icons can be used to distinguish hierarchical levels. Lines can also be drawn to connect objects in the list that share the same hierarchical level. (See Figure 12).

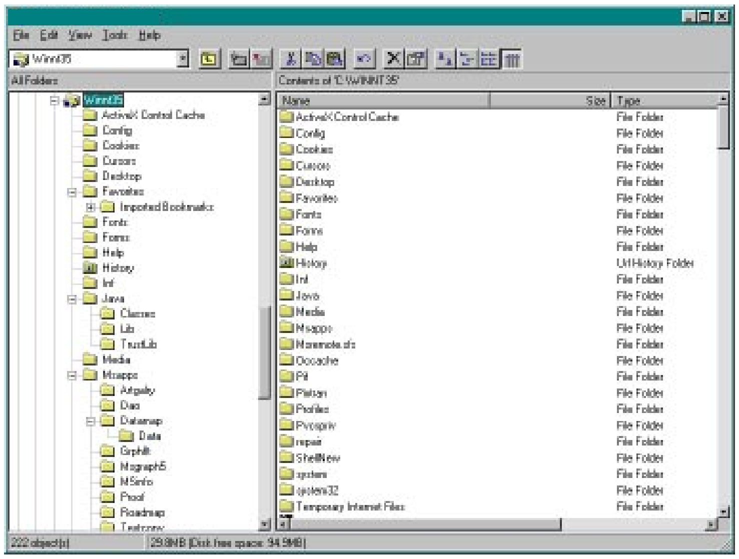
- Use the **list view control** to display items in one of four different views:

  Large icons with a label below it, which can be dragged anywhere.

  Small icons with a label to the right, which can be dragged anywhere.

  List with small icons and a label, presented in columns.

Report in a multicolumn format. The icon and its label are in the two leftmost columns, followed by other columns supplied by the application.



Contents of 'C:\WINNT35'

Name	Size	Type
ActiveX Control Cache		File Folder
Config		File Folder
Cookies		File Folder
Cursors		File Folder
Desktop		File Folder
Favorites		File Folder
Fonts		File Folder
Forms		File Folder
Help		File Folder
History		Url History Folder
Inf		File Folder
Java		File Folder
Media		File Folder
Msapps		File Folder
Msremote.sfs		File Folder
Occache		File Folder
Pif		File Folder
Plstsan		File Folder
Profiles		File Folder
Pvcspriv		File Folder
repair		File Folder
ShellNew		File Folder
system		File Folder
system32		File Folder
Temporary Internet Files		File Folder

222 object(s) | 29.8MB (Disk free space: 94.5MB)

Figure 12. Sample Tree View Window

21.2 <u>Spacing and Alignment of Lists</u>

- Make the list box wide enough to display the widest entry in the list.
- Make the list tall enough for three to eight items to be visible without scrolling.
- Left justify alphanumeric lists and indent any subclassifications.
- Justify numeric lists by decimal point.

21.3 <u>List Titles</u>

- A list box should have a title above the box, either centered or left justified, describing its purpose or contents.

- Capitalize only the first letter of each word in the title, for example: “Contractor Names.”

21.4 Object Selection in Lists

Keep the following uses in mind when designing list boxes:

- A single selection list box allows the selection of only one item in the list.
- Multiple selection list boxes are optimized for selecting noncontiguous items by toggling selections on and off.
- Extended selection list boxes are optimized for selecting a single item or a range of items, although noncontiguous items can be selected using the Shift key.

21.5 Sorting Lists

- Place entries in a list using an order that fits the context, such as numerical, chronological, or alphabetical.
- Use the “View” menu to present sort options. “By Name,” “By Date,” and “By Type” are examples of “View” sorting options.

21.6 Scrolling Lists

See Section 17 for guidelines on usage of horizontal and vertical scroll bars.

22. MENUS

22.1 Usage

- **Drop-down menus** may be used on primary windows.

  See Section 22.8 for standard menu options.

- Use **cascading menus** for progressive disclosure of additional choices for a menu option. Cascades should have no more than three levels, which means the primary drop-down plus two cascades.

- Minimize the use of **pop-up menus,** because they are hidden. However, they can be effective for some frequently used commands, provided that the controls are redundantly available in the drop-down menus.

22.2 Use of Buttons Versus Menus

- When the total number of primary controls is less than six, present them in a row along the bottom or a column on the side. Note: Primary controls apply to the Window as a whole (e.g., “OK,” “Cancel,” and “Help”), not to buttons that apply solely to a list or other widget (e.g., “Add” and “Delete” buttons next to a list).

- When the total number of primary controls is greater than six, present them in menus rather than as buttons.

- A window should have either a menu bar or a row of buttons along the side or bottom. A window should never contain both.

22.3 Naming Menus

- Use the standard menu titles: File, Edit, View, and Help.

  See Section 22.8.

- Other menu titles may be added when needed, provided each title is a single word.

- Generally, menu option names should be either a verb or a verb-noun sequence, with the first letter of each word capitalized (e.g., “Open,” “Save As,” “View Report”).

- When a menu option requires additional information prior to execution, place an ellipsis (...) to the right of the option. A dialog box is usually displayed following selection of this menu option.

22.4 Indicating Activated Selections

- Place a check mark before a menu item to indicate that it is active. For example, in a word processing application, place a check before the “Ruler” menu option when it is turned on.

22.5 Grouping and Ordering Menu Options

- The preferred grouping of menu options is according to **order of use**. The second choice is by **frequency of use**. The last choice is **alphabetical**.

- Use a separator bar to visually define groupings.

- Do not overuse separator bars; generally grouping is not necessary when a menu contains less than seven items.

22.6 Hiding or Graying Menu Options

When a menu option is temporarily unavailable until a precondition is satisfied, gray it out until the precondition is met.

- For example, gray the “Edit > Copy” menu option until the user selects an object to copy.

When a menu option is not authorized (i.e., the user lacks the special access privileges to perform that function), gray it out.

- For example, gray the “Options > Approve” menu option if the user does not have manager approval authority.

For a window that has more than one state (e.g., Edit and View Only), hide any menu options that are never available in a given state.

- For example, if a window has both an Edit and a View Only state, hide the “File > Save” menu option in the View Only state.

- Hide menu options before the window opens.

- Do not hide menu options if the state changes after a window is opened (e.g., Edit changes to View Only after saving). Instead, gray them out, then hide them before the window opens again.

- On-line Help should state the specific conditions in which menu options are and are not available.

22.7 Dynamically Changing Menu Options

The only permissible change to a menu option while a window is open is graying or ungraying.

- Do not dynamically change the label of a menu while the window is open.

- Do not dynamically hide a menu while the window is open.

- Do not dynamically change the structure of a menu while the window is open.

22.8 Standard Menu Bar

When applicable, use the following standard menus, access keys, and shortcut keys:

File		
	New	Ctrl+N
	Open	Ctrl+O
	Save	Ctrl+S
	Save As	
	Print	Ctrl+P
	Exit	
Edit		
	Undo	Ctrl+Z
	Clear	
	Cut	Ctrl+X
	Copy	Ctrl+C

	Paste	Ctrl+V
	Find	
	Duplicate	
	Delete	Ctrl+D
View		
	Show Ruler	Ctrl+R
	Large Icons	Ctrl+L
	Small Icons	Ctrl+I
	Details	
	Zoom	
	Refresh	
Help		
	On Window	
	Topics	
	About *(Application Name)* ...	

23. OPTION BUTTONS (Radio Buttons)

23.1 Usage

- Use option buttons to present two to three mutually exclusive choices.

- Alternately, use a drop-down list for three or more choices because option buttons require a large amount of screen space. Also, as the interface is refined, new choices can be more easily accommodated in a drop-down list.

23.2 Arrangement of Option Buttons

- Option buttons can be placed in columns or rows.

23.3 Presentation in a View Only Mode

- Disable (gray) any option button that is not selected.

- Do not disable (gray) the selected option button. However, do not accept user inputs to deselect the selected option button.

24. SLIDER

24.1 <u>Usage</u>

- Use a slider to set or adjust a value in a range of continuous values.

24.2 <u>Presentation in a View Only Mode</u>

- Do not disable (gray) a slider in a View Only mode. However, do not accept user inputs to change the slider position.

25. SPIN BUTTON

25.1 <u>Usage</u>

- Use a spin button for selecting from a limited set of discrete ordered values that make up a circular loop (i.e., incremental/decremental years).

25.2 <u>Presentation in a View Only Mode</u>

- Do not disable (gray) a spin button in a View Only mode. However, do not accept user inputs to change the spin button value.

26. TAB

- Use a tab control to group information into sections. (Note: A tab control looks like the tabs implemented on the property sheet in Figure 8.)

- Do not use a tab control if the user needs to access information embedded in separate tabs to make a decision.
- When a tab control is implemented, it should be the only control in a window other than a menu, toolbar, or row of buttons along the bottom of the window.

27. TEXT DISPLAY AND STATIC TEXT FIELD

27.1 <u>Usage</u>

- Use a multiple line read-only text field to display lengthy scrolling text.
- Use a static text field to display single line read-only text information.
- Do not use a noneditable text field to display read-only text, instead use a static text field.

27.2 <u>Sizing of Text Display</u>

- At least four lines should be visible.
- Make the window resizable to allow the user to view a larger text area.

27.3 <u>Case in Text Display</u>

- Display text conventionally in mixed upper and lower case.

27.4 <u>Use of Scroll Bars in Text Display</u>

See Section 17 for details on scroll bars.

28. TEXT FIELD

28.1 Usage

- Use a text field to accept single line or multiple line typed entries from the user.

- Use a rich text box when text formatting is necessary (such as setting font, bold, italic or underline).

- Do not use a noneditable text field to display read only text, instead use a static text field.

28.2 Size of Text Fields

- Set the width of the text field to the maximum length of a valid entry, with the exception of lengthy entries which must be scrolled.

28.3 Case in Text Fields

- Allow the user to enter mixed upper and lower case.

28.4 Alignment of Data Within a Text Field

- Data within a text field should be left justified.

28.5 Vertical Alignment of Text Fields in a Window

- When two or more text fields are vertically stacked, perform the following actions:

  (1) Align the labels of the text fields vertically and left justify them.

  (2) Align the text fields vertically and left justify them.

(3) Separate the labels from the fields so that the longest label is 1/8 inch from its text field.

28.6 Cursor Shape in Text Fields

- The Microsoft standard for the text entry cursor is the vertical I bar. The user may prefer to override this default.

28.7 Use of Scroll Bars in Multiple Line Text Fields

See Section 17 for details on scroll bars.

28.8 Range and Format Hints for Text Fields

When a text field requires entry of data in a specific unit, format, or range, provide a cue to the user.

- For example, place a unit at the end of the label for the text field or after the text field itself. (See Figure 13).
- Measurement units should be operationally significant. For example, use knots instead of miles per hour for a maritime application.
- On-line Help should state the accepted data unit, format, or range.

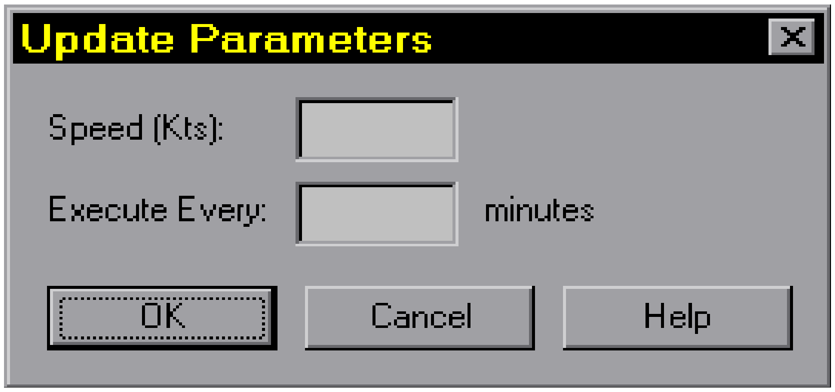


Figure 13. Example of Placing a Unit at the End of a Label or Window Object

29. LABELS FOR WINDOW OBJECTS

29.1 Usage

- Provide labels for every window object (e.g., text fields, sliders, combo boxes).

- Implement labels as static text fields.

  See Section 27 for details on static text fields.

- A list box should have a label above it, either centered or left justified, describing its purpose or contents.

- A text display, scrolled editable text field, or scrolled static text field should have either:

  A label above it that is centered or left justified, or

  A label to the left with a minimum 1/8 inch separation.

- Labels should be consistent across all system applications for the same window object.

29.2 Case and Format of Labels

- Labels should be presented in mixed upper and lower case.

- Capitalize only the first letter of each word in the label, for example: “Contractor Names.”

29.3 Vertical Alignment of Labels and Window Objects

- When two or more window objects (e.g., text fields) are vertically stacked, perform the following actions:

  (1) Align the labels vertically and left justify them.

  (2) Align the window objects vertically and left justify them.

(3) Separate the labels from the window objects so that the longest label is 1/8 inch from the associated window object.

30. ICONS

30.1 Usage

- Create an icon for each application. This icon is used to represent the application executable in the Program Manager or Explorer, and in the window title bar.

- For any application that produces a data file, create an icon. Use this icon to represent the document type in the File Navigator and in the window title bar of the file.

- When an application produces more than one type of data file, it may not be necessary to create a unique icon for each type.

30.2 Size of Icons

- For each icon, create a representation in each of the following sizes:
  Large (32 by 32 pixels)
  Small (16 by 16 pixels)
  For a 256 color display (48 by 48 pixels)

- Note: Depending on the fielded hardware, the representation for the 256 color display may not be needed.

30.3 Presentation of Icons

- Icons should not contain embedded words.

- Icons should appear three-dimensional, rather than flat line drawings.

31. MESSAGE BOXES

31.1 Usage

<u>Information</u>: Use an Information message box for displaying a progress bar, presenting system feedback for user actions, notification of events, or errors caused by the user. (See Figure 1.)

- Information messages do not require a decision or choice. The **OK** button is used to acknowledge receipt of the message.
- When an Information message box is used for error notification, state the nature of the problem and how to recover.

  See Section 11.1 for details on error handling.
- For processes that take longer than 5 seconds, present a progress bar in an Information message box.

  See Section 8 for details about indicating system status.
- Use an Information message box with a progress bar to display system progress, rather than a series of text messages, so that the user is not forced to acknowledge numerous messages about stages of completion.
- Only one button should be presented in the Information message box: the **OK** button.

<u>Warning</u>: Use a Warning message box to solicit a decision that is required before work can proceed. (See Figure 2.)

- The message should be stated in the form of a question. Phrase the question to permit a Yes/No answer.
- Two buttons should be presented: the **Yes** and **No** buttons.
- Rarely, a **Cancel** button may appear to the right. **Cancel** allows the user to back up one step. For example, the Warning: “Do you want to save the data?” should provide a **Cancel** button that cancels the save operation entirely.

<u>Critical</u>: Use a Critical message box to advise the user of a serious problem that needs intervention or correction before work can continue. (See Figure 3.)

- The message is <u>not</u> in the form of a question.
- Two combinations of buttons are permissible:

  Abort and **Retry**
  Retry and **Cancel**.

31.2 <u>Make Messages Informative</u>

- Make messages brief, not more than 2-3 lines.

31.3 <u>Use Neutral Wording</u>

- Avoid phrasing that blames the user or implies a user error.
- Avoid the following words altogether:

Error	Invalid	Warning	Mistake	Null
Failure	Illegal	Stop	Sorry	constraint_error
Please	Thank You			

- Do not use exclamation points (!).
- Messages should not personalize the computer or attribute human qualities to it.
- Do not attempt to be humorous or sarcastic.
- Shorten messages by eliminating unnecessary words as illustrated in the examples below:

"Select the Search button"	-->	"Select Search"
"In order to"	-->	"To"

“Are required to”	-->	“Must”
“This window allows you to”	-->	“Use to”
“Allows you to”	-->	“Lets you”
“Are not allowed to be used”	-->	“Cannot be used”

- Avoid negatives such as:

  “Are you sure that you don't want to save the file?”

31.4 Title of the Message Box

- In the title bar of the message box, display the name of the application posting the message.

31.5 Message Box Buttons

- Arrange command buttons in a row along the bottom of the message box.
- If a message requires the user to make a choice, include a button for each option.
- Set the default button to the least destructive, most likely command.
- Recommended button combinations are:

Information

OK		

Warning

Yes	No	
Yes	No	Cancel

Critical

Abort	Retry
Retry	Cancel

32. MANDATORY ENTRIES

- Present an information message box to inform the user of a missed mandatory entry upon “Save” or “OK.”
- When more than one mandatory entry is missed, present a message about ALL missed entries in a single information box.
- On-line help must clearly indicate mandatory fields.
- In addition to help, it is best to have a visual indication in the window; for example, place an asterisk beside a mandatory field.

33. DATE AND TIME

- Format date as DDMMMYYYY.
- Format time as HH:MM:SS. (Seconds are optional.)
- Time should be presented as local time.

34. OK AND APPLY BUTTONS

- An **OK** button should always initiate an action and close the window.
- An **Apply** button should initiate an action but leave the window open.
- **OK** and **Apply** should only be presented as buttons, never as menu options.

See Section 20.6 about graying or hiding buttons.

35. EXIT/QUIT/CANCEL/CLOSE OPERATIONS

- **Exit** terminates the application and closes all windows associated with it. Exit may be placed in either the “File” menu or on a button.

  See Section 22.8 for use of Exit in standard menus.

- **Quit** should not be used. Use **Exit** instead.

- **Cancel** is used in secondary windows. Cancel ignores all inputs and closes the window. Cancel only appears as a button, never as a menu option. Do not use Cancel to terminate an application.

  See Section 30.1 for use of Cancel in message boxes.

- **Close** closes a window within an application. Close may be placed in either the "File" menu or on a button. Do not use Close to terminate an application or to cancel user inputs.

- **Cancel**, **Close**, and **Exit** should never appear on the same window. **Cancel** is typically used in windows where the user can either cancel all inputs or activate them with the OK button. **Close** is used in windows with a View Only mode. **Exit** is used in windows where the main application and all its secondary windows are closed.

  See Section 20.6 about graying or hiding buttons.

- When an **Exit** or **Close** appears in a menu bar, a question box should be presented prompting the user to save any data not previously saved.
  Note: Do <u>not</u> present the question box if all data has been previously saved.

36. ON-LINE HELP

36.1 <u>Usage</u>

- Every window, with the exception of simple dialog boxes, should have either a Help button or Help menu.

- Help should be presented as either the rightmost menu title or as the rightmost button along the bottom of a window.

  See Section 22.8 for use of Help in standard menus.

36.2 <u>Accessing Help</u>

Allow access to help through the following means:

- Help menu or button.
- F1 key for context sensitive help.
- Hypertext link within a Help page.

- **Help Menu or Button:** Use to provide help on the window. Include an overview description of the window and provide hypertext access to detailed information on window objects and procedures.

  See Section 22.8 for use of Help in standard menus.

- **Context-Sensitive:** Use to provide detailed help on an object in the window by first selecting the object and then pressing the F1 key.

- **Hypertext:** Use hypertext links to provide jumps to related information or topics.

36.3 Help Text

- Make sentences short.

- At least 50 percent of a help window should be white space.

- Concentrate on developing window-level help, rather than help on objects, if resources are limited.