Downloaded from http://www.everyspec.com

CM Reference: 74323

Common Operating Environment (COE) User Interface Specifications (UIS)

Version 4.3

31 December 2003

Prepared for:

Global Information Grid Enterprise Services Directorate
Defense Information Systems Agency

Downloaded from http://www.everyspec.com

This page is intentionally blank.

Downloaded from http://www.everyspec.com

Common Operating Environment (COE)
User Interface Specifications (UIS)

Version 4.3

31 December 2003

Submitted by:

JULIE MINTZ
COE Chief Engineer

Approved by:

DAWN C. MEYERRIECKS
Chief Technology Officer
Defense Information Systems Agency

Downloaded from http://www.everyspec.com

This page intentionally left blank.

Downloaded from http://www.everyspec.com

Table of Contents

Foreword xi

Executive Summary xiii

1. Introduction 1-1
   1.1 Background 1-1
   1.2 Purpose 1-2
   1.3 Scope 1-3
   1.4 Style Compliance Requirements 1-5
      1.4.1 Compliance With COE Requirements 1-5
      1.4.2 Minimum COE Style Requirements 1-7
      1.4.3 Compliance with Federal Accessibility Standards 1-8
      1.4.4 Compliance with Microsoft Certification Requirements 1-8
   1.5 Document Overview 1-10
   1.6 Typographic Conventions 1-11
   1.7 Source Documents 1-12

GRAPHICAL USER INTERFACE DESIGN

2. Input Devices 2-1
   2.1 Pointing Device Input 2-1
      2.1.1 Pointers 2-1
      2.1.2 Pointer Shapes 2-1
      2.1.3 Pointing Device Buttons 2-3
   2.2 Keyboard Input 2-5
      2.2.1 Text Entry 2-5
      2.2.2 Fixed Function Keys 2-5
      2.2.3 Mnemonics 2-5
      2.2.4 Shortcut Keys 2-6
      2.2.5 Variable Function Keys 2-7
   2.3 Interchangeability Between Input Devices 2-9
   2.4 Alternate Input Devices 2-10

3. User-Computer Interaction 3-1
   3.1 Navigation 3-1
      3.1.1 Pointing Device Navigation 3-1
      3.1.2 Keyboard Navigation 3-1
   3.2 Selection 3-3
      3.2.1 Pointing Device Selection 3-3
      3.2.2 Keyboard Selection 3-3
   3.2 Activation 3-5
   3.4 Transfer 3-6
      3.4.1 Drag Transfer 3-6
      3.4.2 Clipboard Transfer 3-6
      3.4.3 Primary and Quick Transfer (Motif) 3-7

Downloaded from http://www.everyspec.com

3.5 Interactive Control 3-9
3.5.1 Object-Action Paradigm 3-9
3.5.2 Minimizing User Error 3-9
3.5.3 Immediate Feedback 3-9
3.5.4 Error Detection 3-10
3.5.5 Explicit Destruction 3-10
3.5.6 General Undo Capability 3-11
3.5.7 Use of Processing Modes 3-11

4. Window Management 4-1
4.1 Window Components 4-1
4.1.1 Appearance 4-1
4.1.2 Behavior 4-6
4.2 Window Families 4-11
4.2.1 Parent-Child Relationships 4-11
4.2.2 Window Mode 4-11
4.3 Window Buttons and Window Icons 4-13
4.3.1 Window Buttons (GNOME, MS Windows) 4-13
4.3.1.1 Appearance 4-13
4.3.1.2 Behavior 4-13
4.3.2 Window Icons (Motif) 4-14
4.3.2.1 Appearance 4-14
4.3.2.2 Behavior 4-14
4.4 Window Management Considerations 4-15
4.4.1 Window Size 4-15
4.4.2 Window Location 4-15

5. Menus 5-1
5.1 Pull-Down Menus 5-1
5.1.1 Appearance 5-1
5.1.2 Behavior 5-2
5.2 Pop-Up Menus 5-4
5.2.1 Appearance 5-4
5.2.2 Behavior 5-5
5.3 Submenus 5-6
5.3.1 Appearance 5-6
5.3.2 Behavior 5-6
5.4 Menu Design 5-8
5.4.1 Types of Menu Options 5-8
5.4.2 Available and Unavailable Options 5-9
5.4.3 Format and Wording 5-9
5.4.4 Grouping into Menus and Submenus 5-11
5.4.5 Mnemonics and Shortcut Keys 5-12

6. Controls 6-1
6.1 Push Buttons 6-1
6.1.1 Appearance 6-1
6.1.2 Behavior 6-2

Downloaded from http://www.everyspec.com

6.2 Radio Buttons 6-4
6.2.1 Appearance 6-4
6.2.2 Behavior 6-5
6.3 Check Boxes 6-6
6.3.1 Appearance 6-6
6.3.2 Behavior 6-6
6.4 Toolbar Buttons 6-7
6.4.1 Appearance 6-7
6.4.2 Behavior 6-8
6.5 List Boxes 6-9
6.5.1 Appearance 6-9
6.5.2 Behavior 6-11
6.6 Drop-Down List Boxes 6-12
6.6.1 Appearance 6-12
6.6.2 Behavior 6-12
6.7 Option Menus (GNOME, Motif) 6-14
6.7.1 Appearance 6-14
6.7.2 Behavior 6-14
6.8 Tree Controls 6-16
6.8.1 Appearance 6-16
6.8.2 Behavior 6-16

7. More Controls 7-1
7.1 Text Boxes 7-1
7.1.1 Appearance 7-1
7.1.2 Behavior 7-2
7.2 Combo Boxes (GNOME, MS Windows) 7-5
7.2.1 Appearance 7-5
7.2.2 Behavior 7-5
7.3 Drop-Down Combo Boxes 7-7
7.3.1 Appearance 7-7
7.3.2 Behavior 7-7
7.4 Spin Boxes (GNOME, Motif, MS Windows) 7-9
7.4.1 Appearance 7-9
7.4.2 Behavior 7-9
7.5 Scales 7-10
7.5.1 Appearance 7-10
7.5.2 Behavior 7-10
7.6 Tab Controls 7-11
7.6.1 Appearance 7-11
7.6.2 Behavior 7-12
7.7 Scrolled Areas 7-13
7.7.1 Appearance 7-13
7.7.2 Behavior 7-13
7.8 Paned Areas 7-15
7.8.1 Appearance 7-15
7.8.2 Behavior 7-15

Downloaded from http://www.everyspec.com

8. Window Design 8-1
8.1 Primary Windows 8-1
8.1.1 Window Titles 8-1
8.1.2 Window Content 8-2
8.1.2.1 Menu Bars 8-2
8.1.2.2 Toolbars 8-7
8.1.2.3 Status Bars 8-8
8.2 Dialog Windows 8-9
8.2.1 Window Titles 8-9
8.2.2 Window Content 8-9
8.2.2.1 Arrangement of Controls 8-9
8.2.2.2 Push Buttons 8-13
8.2.3 Types of Dialog Windows 8-14
8.3 Message Windows 8-16
8.3.1 Window Titles 8-16
8.3.2 Window Content 8-17
8.3.2.1 Arrangement of Controls 8-17
8.3.2.2 Push Buttons 8-18
8.3.3 Types of Message Windows 8-19

9. User Support Resources 9-1
9.1 Context-Sensitive Help 9-1
9.1.1 Contextual Help 9-1
9.1.2 Tooltip Help 9-1
9.1.3 Help Windows 9-2
9.2 Application Help 9-3

10. Application Design and Configuration 10-1
10.1 Application Design 10-1
10.1.1 Single Document Interface Applications 10-1
10.1.2 Multiple Document Interface Applications (MS Windows) 10-2
10.1.3 Nested Applications 10-3
10.2 Application Configuration 10-5
10.2.1 Visual Design 10-5
10.2.1.1 Icons 10-5
10.2.1.2 Layout and Visual Alignment 10-8
10.2.1.3 Capitalization 10-6
10.2.2 Style Management 10-7
10.2.2.1 Font and Color Attributes 10-7
10.2.2.2 Input Focus 10-8
10.2.2.3 Pointing Device Features 10-8

WEB INTERFACE DESIGN

11. Web Page Design 11-1
11.1 Page Elements 11-1
11.1.1 Page Titles 11-1

Downloaded from http://www.everyspec.com

11.1.2 Page Headers 11-1
11.1 3 Page Footers 11-1
11.1.4 Links 11-1
11.1.4.1 Embedded Links 11-2
11.1.4.2 Structural Links 11-2
11.1.4.3 Link Titles 11-3
11.1.4.4 Page Paths 11-3
11.2 Page Layout 11-4
11.2.1 Design Templates 11-4
11.2.2 Layout Tables 11-4
11.2.3 Frames 11-5
11.2.4 Cascading Style Sheets 11-6
11.3 Page Content 11-7
11.3.1 Text 11-7
11.3.1.1 Font Attributes 11-7
11.3.1.2 Grammar and Editorial Style 11-7
11.3.1.3 Alignment and Spacing 11-8
11.3.1.4 Headings 11-8
11.3.2 Images and Multimedia 11-9
11.3.2.1 Images and Image Maps 11-9
11.3.2.2 Animation and Multimedia 11-10
11.3.3 Applets 11-10
11.4 Page Design Considerations 11-11
11.4.1 Page Dimensions 11-11
11.4.2 Page Size 11-11
11.4.3 Device-Independent Page Design 11-12
11.4.4 Accessible Page Design 11-12
11.4.5 Support for Printing 11-13

12. Web Site Design 12-1
12.1 Site Structure 12-1
12.1.1 Information Organization 12-1
12.1.2 Site Structure 12-2
12.2 Site Elements 12-4
12.2.1 Home Pages 12-4
12.2.2 Menu Pages 12-4
12.2.3 Site Index Pages 12-5
12.2.4 Data Entry Pages 12-5
12.2.5 Search Query and Results Pages 12-5
12.2.6 Content Pages 12-6
12.3 Site Design Considerations 12-8
12.3.1 Site Labeling 12-8
12.3.2 Site Updating and Maintenance 12-8

SUPPLEMENTARY TOPICS

13. Information Presentation 13-1

Downloaded from http://www.everyspec.com

13.1 Text Information 13-1
13.1.1 Font Attributes 13-1
13.1.2 Grammar and Wording 13-1
13.1.3 Formats for Date/Time and Latitude/Longitude 13-1
13.2 Graphical Information 13-3
13.2.1 Line Graphs and Surface Charts 13-3
13.2.2 Bar Charts 13-4
13.2.3 Flow Charts 13-5
13.2.4 Pie Charts 13-6
13.2.5 Graphic Schedules 13-6
13.3 Information Coding 13-8
13.3.1 Color 13-8
13.3.2 Reverse Video 13-9
13.3.3 Size and Shape 13-9
13.3.4 Typography 13-10
13.3.5 Flashing 13-10
13.3.6 Sound 13-10
13.4 Dynamic Information 13-12

14. Map Windows 14-1
14.1 Window Design 14-1
14.2 Displaying and Interacting With Map Information 14-4
14.2.1 Map Display and Manipulation 14-4
14.2.2 Interaction With Map Objects 14-5
14.2.3 Tactical Symbology and Graphics 14-7
14.2.4 Coding Tactical Information 14-8

15. Interacting With Drag-Enabled Objects 15-1
15.1 Drag Transfer Methods 15-1
15.1.1 Supported Transfer Actions 15-1
15.1.2 Drag Site and Drop Site Responsibilities 15-2
15.2 Drag Transfer Results 15-4
15.2.1 Representation of Dropped Objects 15-4
15.2.2 Interacting With Shared Objects 15-5
15.2.2.1 Shared Object Behaviors 15-5
15.2.2.2 Pop-Up Menus for Shared Objects 15-6

16. Object-Oriented Design 16-1
16.1 Object Concepts 16-1
16.1.1 Object Types 16-1
16.1.2 Object Views 16-1
16.1.3 Object and Shortcut Icons 16-2
16.2 Application Design 16-3
16.2.1 Object Model 16-3
16.2.1.1 Objects 16-3
16.2.1.2 Views 16-4
16.2.1.3 Object and Shortcut Icons 16-5
16.2.2 User-Computer Interaction 16-6

Downloaded from http://www.everyspec.com

16.2.3 Window Design 16-6
16.2.3.1 Window Titles 16-6
16.2.3.2 Menu Bars 16-6
16.2.3.3 Toolbars 16-7
16.2.3.4 Controls 16-7
16.2.3.5 Pop-up Menus 16-8

17. User Interface Internationalization 17-1
17.1 Operating With Non-US Languages 17-1
17.1.1 Character Rendering 17-1
17.1.2 Structural Rules for Character Handling 17-2
17.2 Text Translation 17-4
17.2.1 Creating Internationalized English Text 17-4
17.2.2 Translating Text and Messages 17-4
17.2.3 Translating Documentation 17-5
17.3 Text Input Methods 17-7
17.3.1 Keyboards and Keyboard Input 17-7
17.3.2 Approaches to Text Entry 17-8
17.3.3 Other Text Entry Actions 17-9
17.4 Internationalizing User Interface Features 17-11
17.4.1 Text Expansion 17-11
17.4.2 Nonlinguistic Text Features 17-12
17.4.3 Data Formats 17-13
17.4.4 Graphics 17-15
17.4.5 Keyboard Interaction 17-16
17.4.6 Text Manipulation 17-17
17.4.7 Adjustments for Bi-directional Languages 17-18
17.4.8 Adjustments for Vertical Languages 17-19
17.4.9 Printing 17-19
17.4.10 Internationalizing Web Sites 17-19

APPENDICES

Appendix A: Action Vocabulary A-1
Appendix B: Federal Accessibility Standards B-1
Appendix C: Acronyms and Abbreviations C-1
Appendix D: Terminology D-1
Appendix E: Glossary E-1
Appendix F: Style Requirements for COE Compliance F-1

Index G-1

List of Figures

Figure 2-1. Standard pointer shapes 2-2
Figure 2-2. Example mnemonics and shortcut keys 2-6

Downloaded from http://www.everyspec.com

Figure 2-3. Example variable action keys 2-8

Figure 3-1. Location cursors in MS Windows 3-1

Figure 4-1. Primary window components in GNOME 4-3
Figure 4-2. Primary window components in Motif 4-3
Figure 4-3. Standard window components in MS Windows 4-4
Figure 4-4. Interaction restrictions in modeless and modal windows 4-12
Figure 4-5. Example window buttons 4-13
Figure 4-6. Example window icon 4-14

Figure 5-1. Example pull-down menu 5-1
Figure 5-2. Example pop-up menu in Motif 5-4
Figure 5-3. Example submenu 5-6
Figure 5-4. Examples of menu option types 5-8
Figure 5-5. Example of an unavailable menu option 5-9
Figure 5-6. Organization of options in submenus 5-12

Figure 6-1. Example push buttons 6-1
Figure 6-2. Example radio button group 6-4
Figure 6-3. Example check box group 6-6
Figure 6-4. Example toolbar buttons 6-7
Figure 6-5. Example list box 6-9
Figure 6-6. Example multiple-column list box 6-10
Figure 6-7. Example drop-down list box 6-12
Figure 6-8. Example option menu 6-14
Figure 6-9. Example collapsed and expanded tree hierarchy 6-16

Figure 7-1. Example text box 7-1
Figure 7-2. Example text box labels providing format cues 7-2
Figure 7-3. Example combo box 7-5
Figure 7-4. Example drop-down combo box 7-7
Figure 7-5. Example spin box 7-9
Figure 7-6. Example scale 7-11
Figure 7-7. Example tab control 7-12
Figure 7-8. Example scroll bar in MS Windows 7-14
Figure 7-9. Example sash 7-16

Figure 8-1. Example primary window 8-1
Figure 8-2. Example dialog window 8-9
Figure 8-3. Example group boxes 8-11
Figure 8-4. Correct and incorrect alignment of labels and text boxes 8-12
Figure 8-5. Example message window in GNOME 8-16
Figure 8-6. Example message window in Java 8-16
Figure 8-7. Example message window in Motif 8-17
Figure 8-8. Example progress bar 8-18

Figure 9-1. Example help pop-up window 9-1

Downloaded from http://www.everyspec.com

Figure 9-2. Example Help dialog window 9-2
Figure 9-3. Example format for browser-based application help 9-3

Figure 10-1. Example SDI application 10-1
Figure 10-2. Example nested application 10-3
Figure 10-3. Options for accessing child segments in a nested application 10-4

Figure 11-1. Example layout grids for Web pages 11-4

Figure 12-1. Example Web site structure 12-2

Figure 13-1. Example line graph 13-3
Figure 13-2. Example bar chart 13-4
Figure 13-3. Example of labeling decision points in a flow chart 13-5
Figure 13-4. Example pie chart 13-6
Figure 13-5. Example graphical schedule 13-7

Figure 14-1. Example map windows 14-1
Figure 14-2. Representation of the map information space 14-3

Figure 15-1. Dragging objects from a chart view to a table view in COP 15-2

List of Tables

Table 2-1. Assignment of actions to pointing device buttons 2-4

Table 3-1. Pointing device selection models 3-3
Table 3-2. Keyboard selection models 3-4

Table 4-1. Window components in primary, dialog, and message windows 4-2
Table 4-2. Control menu contents 4-5
Table 4-3. Pointing device interaction methods for windows 4-7
Table 4-4. Keyboard interaction methods for windows 4-9

Table 5-1. Keyboard interaction methods for pull-down menus 5-3

Table 8-1. Common menus 8-4
Table 8-2. Assignment of mnemonics to interface components in dialog windows 8-12
Table 8-3. Common dialog windows 8-15
Table 8-4. Common message windows 8-19

Table 10-1. Icon requirements in GNOME 10-5
Table 10-2. Icon requirements in Motif 10-6
Table 10-3. Capitalization of interface components and labels 10-7

Table 11-1. Maximum allowable page size for response times at different

Downloaded from http://www.everyspec.com

connection speeds 11-12

Table 16-1. Attributes of application-oriented and object-oriented user interfaces 16-3

Table 17-1. Allowances for text expansion 17-11

Downloaded from http://www.everyspec.com

Foreword

This document, which supercedes all previous versions of the COE UIS, is the final release of the specifications for the COE.

Comments and questions about this document should be directed to:

DISA COE Engineering Office
Attention: GE2
Department of Defense
Defense Information Systems Agency
Seven Skyline Place
5275 Leesburg Pike
Falls Church, VA 22041

coeengr@ncr.disa.mil

Downloaded from http://www.everyspec.com

This page intentionally left blank.

Downloaded from http://www.everyspec.com

Executive Summary

Background

This document defines the user interface style to be delivered by software developed for the Common Operating Environment (COE). The style specifications address the appearance and behavior of individual interface components and provide rules for application and window design in software with a graphical user interface (GUI) or a Web user interface. A common user interface style is essential to the overall usability of COE-based systems so that users can interact effectively with a variety of complex, multi-windowed applications within a single system.

The COE provides a framework for developing and fielding Department of Defense (DoD) systems that meet the needs of the warfighter in a global information environment. DoD relies on the COE to provide the degree of system integration and interoperability needed to meet warfighter requirements. The COE addresses systems in the command, control, communications, computers, and intelligence (C4I) and combat support domains within DoD. The Global Command and Control System and the Global Combat Support System are examples of C4I and combat support systems, respectively, that are based on the COE and support the joint warfighter.

Scope

The style specifications provided here apply to software with a GNOME, Java, Motif, or Microsoft (MS) Windows GUI or a Web user interface. COE specifications for GUIs conform to the style defined in documentation provided for these interfaces and implemented on Linux, UNIX, and MS Windows platforms. In addition, the specifications incorporate best commercial practices as published in the technical literature on user interface design. An integrated set of specifications is provided whenever possible, with separate direction indicated where the styles differ. COE specifications for Web user interfaces are based on features available in version 3.2 or later of Hypertext Markup Language (HTML) and incorporate guidance available in the technical literature on Web design.

The specifications address user interface design for software installed on workstations with a color monitor, keyboard, and pointing device and available in office-like workspaces with normal ambient lighting. The specifications do not address character-based user interfaces or offer direction regarding migration from this type of interface to a GUI-based one. The specifications focus on the style attributes that compliant software must possess in order to be accepted into the segment repository and do not dictate a methodology or set of tools for user interface development. This document focuses on user interface design and does not provide environmental or ergonomic specifications on topics such as lighting and noise or workstation design and layout.

Downloaded from http://www.everyspec.com

Style Compliance Requirements

This section defines style compliance requirements for applications that will be available only on UNIX and MS Windows platforms and for applications that will be available on UNIX, MS Windows, and Linux platforms. The UNIX and MS Windows environments can be characterized in terms of a common user interface style, whereas the Linux environment may contain a mix of styles.

Software targeted for use only on UNIX and MS Windows platforms shall comply with the following style requirements:

- A Motif-based application shall have a COE-compliant Motif style.
- A MS Windows-based application shall have a COE-compliant MS Windows style.
- A Java-based application shall have a COE-compliant style that matches the GUI of the host platform.
- An application with a Web user interface shall have a COE-compliant Web style.

Software intended for use on all COE platforms shall comply with the following style requirements:

- A Motif-based application shall have a COE-compliant Motif style.
- A MS Windows-based application shall have a COE-compliant MS Windows style.
- A GNOME-based application shall have a COE-compliant GNOME style.
- A Java-based application shall have a COE-compliant Java style.
- An application with a Web user interface shall have a COE-compliant Web style.

Style compliance is required in the development of all new software and the migration of existing software submitted for inclusion in the segment repository. DoD organizations are expected to comply with all style specifications, with deviations occurring only when called for by operational requirements. The Defense Information Systems Agency (DISA) specifies the style requirements to be satisfied at each compliance level defined by the COE and makes tools available for assessing style compliance. Developers are expected to address COE style requirements as part of their software development process but do not have to provide documentation to DISA that they have satisfied these requirements when being evaluated for COE compliance.

Document Overview

Sections 2 through 10 of this document address GUI design. Sections 2 and 3 describe the input devices available to users and the manner in which they use these devices to interact with an application. Sections 4 through 7 address the appearance and behavior of individual interface components, including windows, menus, and controls, that may be present in an application. Section 8 covers the design of primary, dialog, and message windows, while section 9 focuses on the availability of user support resources. Section 10

Downloaded from http://www.everyspec.com

describes approaches to application design and considerations in visual design and style management.

Sections 11 and 12 focus on Web interface design. Section 11 describes Web page design, while section 12 discusses design of Web sites.

Sections 13 through 17 cover supplementary topics related to user interface design. Section 13 addresses the presentation of text and graphical information. Section 14 presents user requirements for map windows, while section 15 explains how users interact with drag-enabled objects in a COE-based system. Section 16 describes object-oriented design and provides direction for applications implementing this design approach. Section 17 provides guidance related to user interface internationalization; this section is included for use by DoD organizations with a requirement to provide internationalized software and is not considered in determining COE style compliance.

Appendix A defines standard vocabulary, mnemonics, and shortcut keys for common actions. Appendix B describes federal standards for information accessibility by individuals with disabilities. Appendix C lists the acronyms and abbreviations used in this document. Appendix D maps the terminology used here to the style documentation for each interface, while Appendix E provides a glossary of style terminology. Appendix F maps the style specifications to COE compliance levels.

Summary of Changes

This document is an update to version 4.1 of the Common Operating Environment (COE) User Interface Specifications (UIS). There is no version 4.2 of the UIS; the current document is identified as version 4.3 to correspond with the release of version 4.3 of the COE Integration and Runtime Specification and version 4.X of the COE software.

The specifications have been updated to address the styles provided by GNOME and Java (along with Motif and MS Windows) and to incorporate design guidance for GUIs and Web user interfaces published in the technical literature since the previous release of the document. Because the GNOME interface continues to mature and its style documentation is minimal, the specifications are supplemented with notes indicating where usability issues have been identified and providing a recommended implementation in COE software.

The content of the current document has been reorganized to group sections into three parts dealing with GUIs, Web user interfaces, and supplemental topics related to user interface design. The section on common secondary windows and several of the appendices have been deleted, the sections on window design and Web interface design have been reorganized and rewritten to accommodate new content, and a new section on interacting with drag-enabled objects has been added.

Downloaded from http://www.everyspec.com

This page intentionally left blank.

Downloaded from http://www.everyspec.com

1. Introduction

1.1 Background

The Common Operating Environment (COE)[1] provides a framework for developing and fielding Department of Defense (DoD) systems that meet the needs of the warfighter in a global information environment. As indicated in the Command, Control, Communications, Computers, and Intelligence (C4I) for the Warrior concept, "the warrior needs a fused, real-time true-picture of the battlespace and the ability to order, respond, and coordinate vertically and horizontally to the degree necessary to prosecute the mission in that battlespace." DoD relies on the COE to provide the degree of system integration and interoperability required to achieve this vision.

The COE addresses systems in the C4I and combat support domains within DoD. The C4I domain includes systems that facilitate the command and control of forces by the tactical commander, while the combat support domain includes systems that support logistics, transportation, base support, personnel, and health affairs functions. The Global Command and Control System (GCCS) and the Global Combat Support System (GCSS) are examples of C4I and combat support systems, respectively, that are based on the COE and support the joint warfighter.

The COE provides a client-server architecture for developing reusable, interoperable software from which systems tailored to the specific needs of a user community can be built. A COE-based system is composed of software components, called segments, contributed by different sources and maintained in a segment repository. Some segments are part of the COE because they perform common functions required by most systems, while other segments perform mission-specific functions that are targeted to particular operational communities. Software is included in the segment repository only if it conforms to strict standards and specifications that are required to support "plug and play" integration across a range of hardware platforms.

It is critical to the overall usability of a system that the software in the segment repository provide a user interface with a common appearance and behavior so users can interact effectively with any system built from this software. User interface standardization is particularly important as users are provided the capability to interact with a variety of complex, multi-windowed applications within a single system. The benefits to be gained from standardization are increased user productivity, reduced training requirements, improved system reliability, reduced maintenance costs, and increased efficiency in the development of individual applications as well as entire systems.

[1] The concept of the Defense Information Infrastructure (DII) has been superceded by that of the Global Information Grid, and what was previously known as the DII COE is now referred to as the Common Operating Environment. As a result, references to the term "DII" have been removed from both the title and content of this document.

Downloaded from http://www.everyspec.com

1.2 Purpose

This document defines the user interface style to be delivered by software developed for the COE. The specifications provided here emphasize commonality in "look and feel" because it is a key element of usability as well as a requirement of the runtime environment defined by the COE. Compliance with COE style specifications is mandated for all software in the segment repository because the specifications define the "rules" for a well-behaved application[2] to operate predictably in a standard runtime environment. Compliance is especially important since the applications in a system can be built from multiple segments, each produced by a different organization.

A common "look and feel" is one that provides consistency in the appearance and behavior of user interface objects while allowing flexibility for addressing operational requirements. Implementing a common "look and feel" enables users to identify, remember, and predict the rules and organization of a system. By building consistency in the user interface, users can develop an effective and efficient model of how the system works and can generalize this knowledge to other systems. According to Mayhew in Principles and Guidelines in Software User Interface Design, a consistent user interface is one that provides:

- Consistent location of certain types of information on screens,
- Consistent syntax of commands in a command language,
- Similar execution of analogous operations in different applications,
- Consistent design of command names and abbreviations,
- Consistent grammatical form of error messages and instructions,
- Consistent design of captions and fields on forms and displays,
- Consistent dialog style for different functions, and
- Terminology consistent with the users' existing vocabulary.

[2] In this document, the term "application" is used to refer to a user application, i.e., the software with which users interact to perform one or more related operational tasks. In the COE, the tasks in an application can be performed by software taken from different sources. As a result, an application may contain one or more segments, and a single segment may be present in one or more applications.

Downloaded from http://www.everyspec.com

1.3 Scope

DoD policy concerning user interface standardization is published in the human-computer interface standards section of the DoD Joint Technical Architecture (JTA). This document indicates that user interface development is to be based on the style of a commercial graphical user interface (GUI),[3] with additional guidance on design elements specific to DoD applications provided in domain- and system-level style guides. DoD policy on interoperability and supportability of information technology and national security systems is published in DoD Directive 4630.5. Paragraph 5.6.4 of this directive indicates that the Heads of DoD Components shall "implement procedures to ensure the use of DoD JTA, Common Operating Environment (COE) technical guidance, and COE technology for programs under the DoD Components' cognizance."

COE style specifications serve as the domain-level style guide for the C4I and combat support domains, in accordance with JTA policy on user interface standardization, and provide the COE technical guidance, per DoD Directive 4630.5, on user interface design of compliant software. In addition, COE style specifications are consistent with section 5.15 on user-computer interface in Military Standard (MIL-STD) 1472F and reference other military standards providing direction with relevance to user interface design in COE-based systems. Finally, the specifications address federal standards for information accessibility by individuals with disabilities and indicate the applicability of these standards to COE-compliant software.

This document provides style specifications for applications that have a GNOME,[4] Java, Motif, or Microsoft (MS) Windows GUI or a Web user interface. COE specifications for GUIs conform to the style defined in documentation provided for these interfaces and implemented on Linux, UNIX, and MS Windows platforms. In addition, the specifications incorporate best commercial practices as published in the technical literature on user interface design. An integrated set of specifications is provided whenever possible, with separate direction indicated where the styles differ. COE specifications for Web user interfaces are based on features available in version 3.2 or later of Hypertext Markup Language (HTML) and incorporate guidance available in the technical literature on Web design.

The specifications address user interface design for software installed on workstations with a color monitor, keyboard, and pointing device and available in office-like workspaces with normal ambient lighting. The specifications do not address character-based user interfaces or offer direction regarding migration from this type of interface to a GUI-based one. The specifications focus on the style attributes that compliant software must possess in order to be accepted into the segment repository and do not dictate a methodology or set of tools for user interface development. This document focuses on

[3] The JTA contains standards for the design of Motif and MS Windows user interfaces but does not address GNOME or Java interfaces.
[4] GNOME (GNU Object Model Environment, GNU being a recursive acronym for "GNU's Not Unix") is a desktop environment developed by the open source community.

Downloaded from http://www.everyspec.com

user interface design and does not provide environmental or ergonomic specifications on topics such as lighting and noise or workstation design and layout.

Downloaded from http://www.everyspec.com

1.4 Style Compliance Requirements

1.4.1 Compliance With COE Requirements

This section defines style compliance requirements for applications that will be available only on UNIX and MS Windows platforms and for applications that will be available on UNIX, MS Windows, and Linux platforms. The UNIX and MS Windows environments can be characterized in terms of a common user interface style, whereas the Linux environment may contain a mix of styles.

Systems installed on UNIX and MS Windows platforms contain native Motif- or MS Windows-based applications built using a toolkit or builder product and Java-based applications intended for cross-platform use. In order to provide consistency with the native style available on these platforms, software targeted for use only on UNIX and MS Windows platforms shall comply with the following style requirements:[5]

- A Motif-based application shall have a COE-compliant Motif style.
- A MS Windows-based application shall have a COE-compliant MS Windows style.
- A Java-based application shall have a COE-compliant style that matches the GUI of the host platform.
- An application with a Web user interface shall have a COE-compliant Web style.

An application is expected to select the set of interface components required to support effective performance of mission-related tasks and to construct a predictable, well-behaved interface using these components. A Motif- or MS Windows-based application ensures that these components provide the “look and feel” and follow the layout rules defined in Motif and MS Windows standards. A Java-based application is configured to provide the same interface style as that of a native application on the host platform. Style is set to “native” (rather than “Java”) so the interface has a Motif appearance on UNIX platforms and a Windows appearance on MS Windows platforms. Where the GUIs differ in layout rules, a Java-based application follows MS Windows rules as defined in COE style specifications, given that users are expected to interact with these applications on a Windows client.

Systems installed on a Linux platform can contain one or more user interface styles, each with its own rules for appearance, behavior, and layout. While GNOME, Java, Motif, and MS Windows support a similar set of interface components, there are differences in the “look and feel” of these components that cause each GUI to possess unique style features. In an environment where users can interact with multiple styles, it is important that an application present interface components that conform to unique appearance, behavior, and layout attributes defined by a GUI style. Consistency in user interface design contributes to the development of expertise by enabling users to anticipate how they interact with an application and to develop automaticity in their performance. When a user interface is consistent, users can be confident that a window with the “look” of a

[5] These compliance requirements are the same as those included in previous versions of this document.

Downloaded from http://www.everyspec.com

particular GUI will also possess the behavior of that GUI and follow its layout rules. Mixing the rules that define the appearance, behavior, and layout of interface components creates a hybrid style that is unpredictable and has a negative impact on usability. In order to provide a consistent style across GUIs, software intended for use on all COE platforms shall comply with the following style requirements:

- A Motif-based application shall have a COE-compliant Motif style.
- A MS Windows-based application shall have a COE-compliant MS Windows style.
- A GNOME-based application shall have a COE-compliant GNOME style.
- A Java-based application shall have a COE-compliant Java style.
- An application with a Web user interface shall have a COE-compliant Web style.

Both GNOME and Java support themes that can alter the visual appearance of interface components.[6] While themes increase the visual uniformity of a user interface, they can create a hybrid style by removing the visual cues that indicate to users where there may be differences in behavior and/or layout. It is recommended that GNOME-based applications be displayed in a visually unique theme (e.g., one that does not mimic the appearance of another GUI) and that Java-based applications be displayed in "Java" style so these applications have a unique appearance.

COE style specifications have been derived from documentation published for each GUI, then confirmed based on a review of the style provided on each platform. When the four GUIs were compared, it was found that the style guides and implementations were not fully synchronized, consistent, or mature. These inconsistencies are documented in notes included with the COE specifications, along with a recommendation for implementation in COE software. It is recognized that unless the GUIs change, developers may be unable to provide the desired behavior.

COE style specifications require consistency in the implementation of commonly used interface components regardless of the development environment selected to render the components. DoD organizations are responsible for selecting a development tool that produces a user interface with the required style attributes. Software that makes use of commercial off-the-shelf (COTS) tools or integrates COTS products shall be configured to comply with COE specifications as much as possible. DoD organizations need to determine the extent to which a tool or product will generate a user interface with a noncompliant style and whether these divergences will have a negative impact on system usability when the software is integrated with other COE-compliant applications.

The Defense Information Systems Agency (DISA) specifies the style requirements to be satisfied at each compliance level defined by the COE. Compliance with COE style specifications is required in the development of all new software and the migration of existing software submitted for inclusion in the segment repository. DoD organizations are expected to comply with all style specifications, with deviations occurring only when

[6] It is assumed that the "native" style provided by Java will be extended to support the theme currently in effect.

Downloaded from http://www.everyspec.com

called for by operational requirements. New software shall be developed in accordance with COE requirements and be fully compliant with all style specifications; existing software is expected to migrate to full COE compliance. DoD organizations are expected to comply with the intent of the specifications which define the style requirements for compliant software and do not preclude all possible inappropriate, incorrect, or unacceptable implementations.

Appendix F maps COE style specifications to each of the style-related items in the compliance checklist published in the COE Integration and Runtime Specification (I&RTS). A segment must satisfy all of the requirements for a given style-related item in order to be considered to comply with the item. As indicated in the I&RTS, the compliance level assigned to a segment is the highest numbered level where all of the checklist items have been satisfied. While the COE calls for compliance with all of the items in the checklist, only a subset of the requirements related to runtime environment is considered essential to establishing the compliance level. As a result, while "style guide" is one of the compliance areas defined by the COE, software being evaluated for compliance can achieve a particular level without submitting verification to DISA that it has met the style-related requirements defined at that level. Developers are expected to address COE style requirements as part of their software development process, even though they do not have to provide documentation to DISA that they have satisfied these requirements when being evaluated for compliance.[7]

1.4.2 Minimum COE Style Requirements

While COE style specifications allow DoD organizations to tailor the user interface to address operational requirements, some elements of a user interface shall not be modified in COE-compliant software. The following elements are central to a GUI paradigm and are considered essential to the COE integration process at the user interface level:

- The hotspot of the pointer shall indicate the locus of user input with the pointing device.
- The location cursor shall indicate the locus of user input from the keyboard.
- Only one window shall have input focus at any time and can accept keyboard input.
- Window management operations shall be governed by parent-child relationships within the window family.

COE-compliant software must satisfy these minimum requirements in order to achieve level 1 style compliance. Requests for waivers to diverge from these requirements shall be directed to the cognizant Chief Engineer and considered on a case-by-case basis.

[7] With previous releases of COE style specifications, DISA provided a software tool to assist developers in assessing style compliance. The Style Compliance Test Protocol (SCTP) supported automated administration of a checklist of style attributes derived from COE style specifications. An update to SCTP based on the specifications in the current document is not available.

Downloaded from http://www.everyspec.com

1.4.3 Compliance with Federal Accessibility Standards

Section 508 of the Rehabilitation Act requires that individuals with disabilities have access to and use of information that is comparable to that provided to federal employees and members of the public who are not disabled. The standards created under Section 508 define technology accessibility requirements for all types of information technology in the federal sector. Appendix B provides an overview of Section 508 and lists the accessibility standards that apply to software applications and operating systems, web-based intranet and internet information and applications, telecommunications products, video and multi-media products, self-contained closed products such as information kiosks, and desktop and portable applications.

All federal agencies are required to comply with Section 508 standards, with a limited exemption for systems used for military command, weaponry, intelligence, and cryptologic activities. COE-compliant software intended for use only in C4I and combat support systems is considered to fall within the limited exemption allowed for defense-related systems. However, the exemption does not apply to routine business and administrative systems used for other defense-related purposes or by defense agencies or personnel. Consequently, COE-compliant software that will be used in these systems is required to comply with the accessibility standards listed in Appendix B.[8]

COE style specifications already comply with some of the accessibility standards created under Section 508 because these standards represent good design practice for all users, whether or not they are disabled, and will increase software usability for the warfighter in an operational environment. COE-compliant applications that are exempt from Section 508 standards are expected to comply with these style specifications. The remaining accessibility standards focus on providing redundancy in information presentation and interaction so individuals with disabilities can use different modalities to access information. COE specifications call for information redundancy where it will provide value-added for the warfighter but not at the level called for in the accessibility standards. COE-compliant applications that are exempt from Section 508 standards are expected to support information redundancy as defined in COE specifications. COE-compliant applications that are non-exempt must comply with the standards related to information redundancy listed in Appendix B of this document.

1.4.4 Compliance with Microsoft Certification Requirements

Microsoft publishes specifications that define the technical requirements for desktop applications to earn the “Certified for Microsoft Windows” logo. These specifications define the following requirements to be certified as “optimized for accessibility:”

- Support standard system size, color, font, and input settings.
- Ensure compatibility with the High Contrast option.

[8] In the remainder of this document, an application is “non-exempt” if it is required to comply with Section 508 standards, and “exempt” if it is not subject to the standards.

Downloaded from http://www.everyspec.com

- Enable programmatic access to all user interface elements and text.
- Provide keyboard access to all features.
- Expose the location of the keyboard focus.
- Provide user-selectable equivalents for non-text elements.
- Do not rely exclusively on sound to convey information.
- Avoid flashing elements.
- Create accessible documentation about accessibility features.

The COE already complies with some of these requirements because, as with federal accessibility standards, the requirements represent good design practice for all users. The COE requires compliance with Microsoft accessibility specifications by applications that are non-exempt from Section 508 standards, with exceptions allowed for exempt applications in order to meet operational requirements.

Downloaded from http://www.everyspec.com

1.5 Document Overview

The remainder of this document is divided into three parts dealing with GUIs, Web user interfaces, and supplemental topics related to user interface design. Unless otherwise indicated, the specifications apply to GNOME, Java, Motif, and MS Windows interfaces; if a specification applies to a specific GUI, it is identified as such.

- GUI design is covered in sections 2 through 10. Sections 2 and 3 describe the input devices available to users and the manner in which they use these devices to interact with an application. Sections 4 through 7 address the appearance and behavior of individual interface components, including windows, menus, and controls, that may be present in an application. Section 8 covers the design of primary, dialog, and message windows, while section 9 focuses on the availability of user support resources. Section 10 describes approaches to application design and considerations in visual design and style management.

- Web interface design is addressed in sections 11 and 12. Section 11 describes Web page design, while section 12 discusses design of Web sites.

- Supplementary topics related to user interface design are covered in sections 13 through 17. Section 13 addresses the presentation of text and graphical information. Section 14 presents user requirements for map windows, while section 15 explains how users interact with drag-enabled objects in a COE-based system. Section 16 describes object-oriented design and provides direction for applications implementing this design approach. Section 17 provides guidance related to user interface internationalization; this section is included for use by DoD organizations with a requirement to provide internationalized software and is not considered in determining COE style compliance.

- Appendix A defines standard vocabulary, mnemonics, and shortcut keys for common actions. Appendix B describes federal standards for information accessibility by individuals with disabilities. Appendix C lists the acronyms and abbreviations used in this document. Appendix D maps the terminology used here to the style documentation for each interface, while Appendix E provides a glossary of style terminology. Appendix F maps the style specifications to COE compliance levels defined in the COE I&RTS.

Downloaded from http://www.everyspec.com

1.6 Typographic Conventions

The following typographic conventions are used in this document:

- The first letter of each word in push button actions (e.g., Cancel, OK) and menu titles and options (e.g., File, Save As) is capitalized.

- The left, middle, and right buttons on the pointing device are referred to as BLeft, BMiddle, and BRight, respectively.

- The names of keys on the keyboard are presented in upper-case letters (e.g., RETURN). Simultaneous key combinations are indicated by presenting the key names separated by a plus; for example, CTRL+HOME means that users hold down the CTRL key and then press the HOME key.

- References to the Microsoft Windows™ user interface and style are identified as "MS Windows" in order to differentiate them from statements about generic windows.

- Comments identifying inconsistencies in interface styles are indicated in "NOTES" and accompanied by a recommendation for implementation in COE software.

Downloaded from http://www.everyspec.com

1.7 Source Documents

COE style specifications are based on the user interface standards and guidelines listed below.

Government Documents

Architectural and Transportation Barriers Compliance Board. Electronic and Information Accessibility Standards. Published in the Federal Register on December 21, 2000. http://www.access-board.gov/sec508/508standards.htm

Defense Information Systems Agency. Common Operating Environment (COE) Integration and Runtime Specification (I&RTS), Version 4.3, 2003.

Department of Defense. Interoperability and Supportability of Information Technology (IT) and National Security Systems (NSS). DoD Directive 4630.5, 2002. http://www.deskbook.osd.mil/htmlfiles/DBY_dod.asp

Department of Defense. Military Standard 1472F. Department of Defense Design Criteria Standard: Human Engineering. U.S. Army Aviation and Missile Command, Huntsville, AL, 1999.

Department of Defense. Military Standard 2525B. Common Warfighting Symbology, 1999. http://www-symbology.itsi.disa.mil/symbol/mil-std.htm

Federal IT Accessibility Initiative. http://www.section508.gov/

Joint Technical Architecture Development Group. Department of Defense Joint Technical Architecture, Version 5.1, 2003.

Non-Government Documents

Anuff, E. Java Sourcebook: A Complete Guide to Creating Java Applets for the Web. New York: John Wiley & Sons, 1996.

Apple Computer, Inc. Aqua Human Interface Guidelines. Cupertino, CA: Apple Computer, Inc., 2002.

Apple Computer, Inc. Guide to Macintosh Software Localization. Reading, MA: Addison-Wesley Publishing Co., 1992.

Constantine, L.L. & Lockwood, L.A.D. Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design. Reading, MA: Addison-Wesley Publishing Company, 1999.

Fowler, S.L. GUI Design Handbook. New York: McGraw-Hill, 1998.

Downloaded from http://www.everyspec.com

Fowler, S.L. & Stanwick, V.R. The GUI Style Guide. Boston, MA: Academic Press, Inc., 1995.

Galitz, W.O. User-Interface Screen Design. Boston, MA: QED Publishing Group, 1993.

Johnson, J. GUI Bloopers: Don'ts and Do's for Software Developers and Web Designers. San Francisco: Morgan Kaufmann Publishers, 2000.

Lynch, P.J. & Horton, S. Web Style Guide, 2nd Edition. New Haven, CT: Yale University Press, 2001.

Madell, T., Parsons, C. & Abegg, J. Developing and Localizing International Software. Englewood Cliffs, NJ: Prentice Hall, 1994.

Mandel, T. The Elements of User Interface Design. New York: John Wiley & Sons, Inc., 1997.

Marcus, A., Smilonich, N. & Thompson, L. The Cross-GUI Handbook for Multiplatform User Interface Design. Reading, MA: Addison-Wesley Publishing Co., 1995.

Mayhew, D.J. Principles and Guidelines in Software User Interface Design. Englewood Cliffs, NJ: PTR Prentice Hall, 1992.

Microsoft Corporation. "Designed for Microsoft Windows XP" Application Specification, Version 2.3, 2002.

Microsoft Corporation. Microsoft Certified for Windows Server 2003 Application Specification, 2003.

Microsoft Corporation. Microsoft Windows User Experience: Official Guidelines for User Interface Developers and Designers. Redmond, WA: Microsoft Press, 1999.

Morris, M.E.L. & Hinrichs, R.J. Web Page Design. Mountain View, CA: SunSoft Press, 1996.

Morrison, M. HTML & XML for Beginners. Redmond, WA: Microsoft Press, 2001.

Nielsen, J. Designing Web Usability. Indianapolis, IN: New Riders Publishing, 2000.

Downloaded from http://www.everyspec.com

O'Donnell, S.M. Programming for the World: A Guide to Internationalization. Englewood Cliffs, NJ: Prentice Hall, 1994.

Olsen, Jr., D.R. Developing User Interfaces. San Francisco: Morgan Kaufmann Publishers, Inc., 1998.

Perry, P.J. Creating Cool Web Applets with Java. Foster City, CA: IDG Books Worldwide, Inc., 1996.

Shafran, A. Enhancing Netscape Web Pages. Indianapolis, IN: Que Corporation, 1996.

Spool, J.M. & Perfetti, C. The Secret Design Strategies of Highly Successful Web Sites. User Interface Engineering, 2003.

Sun Microsystems, Inc. Java Look and Feel Design Guidelines: Advanced Topics. Palo Alto, CA: Sun Microsystems, Inc., 2001.

Sun Microsystems, Inc. Java Look and Feel Design Guidelines. Second Edition. Palo Alto, CA: Sun Microsystems, Inc., 2001.

Sun Microsystems, Inc. Common Desktop Environment: Style Guide and Certification Checklist. Palo Alto, CA, Sun Microsystems, Inc., 2000.

Sun Microsystems, Inc. Common Desktop Environment: Style Guide Certification Check List. Palo Alto, CA, Sun Microsystems, Inc., 2002.

The Open Group. CDE 2.1/Motif 2.1 - Style Guide and Glossary. Witney, Oxon, UK: Open Group Publications, 1997.

The Open Group. CDE 2.1/Motif 2.1 - Style Guide Certification Checklist. Witney, Oxon, UK: Open Group Publications, 1997.

The Open Group. CDE 2.1/Motif 2.1 - Style Guide Reference. Witney, Oxon, UK: Open Group Publications, 1997.

Uren, E., Howard, R. & Perinotti,T. Software Internationalization and Localization: An Introduction. New York: Van Nostrand Reinhold, 1993.

Online Documents

Barlow, L. The Spider's Apprentice: A Helpful Guide to Web Search Engines. Monash Information Services, 2000. http://www.monash.com/spidap4.html

Berners-Lee, T. Style Guide for Online Hypertext. Massachusetts Institute of Technology, 1995. http://www.w3.org/pub/WWW/Provider/Style/

Downloaded from http://www.everyspec.com

Gillespie, J. Web Page Design for Designers. Pixel Productions UK, 2001. http://www.wpdfd.com/wpdhome.htm

The GNOME Usability Project. GNOME Human Interface Guidelines (1.0). http://developer.gnome.org/projects/gup/hig/1.0/

IBM Web Design Guidelines, undated. http://www-3.ibm.com/ibm/easy/eou_ext.nsf/publish/572#top

Karp, T. Art and the Zen of Web Sites. TLC Systems Corporation, 1999. http://www.tlc-systems.com/webtips.html

Microsoft Corporation. Microsoft Inductive User Interface Guidelines, 2001. http://msdn.microsoft.com/library/default.asp?url=/nhp/default.asp?contentid=28000443

National Cancer Institute. Research-Based Web Design and Usability Guidelines, 2002. http://usability.gov/guidelines

Nielsen, J. The Alertbox: Current Issues in Web Usability. Sun Microsystems, 2001. http://www.useit.com/alertbox/

Raggett, D. W3C HTML 3.2 Reference Specification. WWW Consortium, 1997. http://www.w3.org/TR/REC-html32.html

Steel, W. Hints for Web Authors. University of Mississippi, 2000. http://www.mcsr.olemiss.edu/~mudws/webhints.html

Sullivan, D. Search Engine Watch. Calafia Consulting, 2001. http://searchenginewatch.internet.com/webmasters/

The National Center for Supercomputing Applications. A Beginner's Guide to HTML. University of Illinois at Urbana-Champaign, 2000. http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimerAll.html

Tilton, J.E. Composing Good HTML. Carnegie-Mellon University, 1998. http://www.ology.org/tilt/cgh/

Standards relating to the design of workstations, associated furniture, and the facilities in which they are placed can be found in MIL-STD 1472F and the following document:

American National Standards Institute. National Standard for Human Factors Engineering of Visual Display Terminal Workstations. Santa Monica, CA: The Human Factors Society, Inc., 1988.

Downloaded from http://www.everyspec.com

Guidelines related to information accessibility can be found in the following documents:

Bergman, E. & Johnson, E. Towards Accessible Human-Computer Interaction. http://www.sun.com/access/developers/updt.HCI.advance.html

Chisholm, W., Vanderheiden, G., & Jacobs, I. Web Content Accessibility Guidelines 1.0. W3C Recommendation 5-May-1999. http://www.w3.org/TR/WAI-WEBCONTENT/

IBM Accessibility Center Guidelines. http://www-3.ibm.com/able/guidelines.html

IBM Guidelines for Writing Accessible Applications Using 100% Pure Java. Version 2.1, 2000. http://www-3.ibm.com/able/snsjavag.html

Microsoft Corporation. Guidelines for Keyboard User Interface Design, 2002. http://msdn.microsoft.com/library/?url=/library/en-us/dnacc/html/ATG_KeyboardShortcuts.asp

Thatcher, J., Bohman, P., Burks, M., Henry, S.L., Regan, B, Swierenga, S., Urban, M.D., & Waddell, C.D. Constructing Accessible Web Sites. Birmingham, UK: Glasshaus Ltd., 2002.

Theofanos, M.F. & Redish, J. Bridging the Gap Between Accessibility and Usability. Interactions, Nov-Dec 2003, 36-51.

Downloaded from http://www.everyspec.com

2. Input Devices

2.1 Pointing Device Input

2.1.1 Pointers

The pointing device is used to move a pointer on the screen and select and manipulate objects on which the pointer is placed. The pointer provides a graphical representation of the location of the pointing device on the screen, with the hot spot of the pointer (i.e., the active point) indicating the precise location where pointing device operations occur (e.g., the object that will be selected when users execute a select action). The pointer may change shape depending on where it is located, but the hot spot does not move when a change in shape occurs.

In order to be considered on an object, the hot spot of the pointer is placed within the hot zone of the object. The hot zone is defined by the boundary of an object, which includes both the graphic for an object and its label (e.g., in radio buttons and check boxes). An application implements the pointing device model defined here (i.e., the hot spot of the pointer indicates the locus of user input) and does not modify the size of the hot zone for any interface component.

Users control the movement of the pointer by moving the pointing device and can move the pointer anywhere on the screen. When users move the pointing device, the pointer moves in the corresponding direction. The position of the pointer remains where it is placed on the screen until users move it. An application does not arbitrarily move (i.e., warp) the pointer or restrict pointer movement to part of the screen.

2.1.2 Pointer Shapes

The shape of the pointer provides feedback about the function of the area on which the pointer is placed.

GNOME, Motif, MS Windows: Figure 2-1 lists standard pointer shapes and the functions they perform.

Java: Java defines a set of pointer types that map to the native platform pointers shown in figure 2-1. When no pointer exists in the native platform toolkit, the pointer is supplied by the Java Foundation Class.

An application uses standard pointer shapes whenever the pointer is over an area where the function applies. If an application redefines the pointer shape, it does so only when the pointer is in an application window. The arrow pointer shown in figure 2-1 is used to perform many of the functions available in a window. The pointer is assumed to have this shape, unless otherwise indicated.

Downloaded from http://www.everyspec.com

GNOME	Motif	MS Windows	Pointer Name and Function
Selection Pointers			
			Arrow pointer (selects objects).
			I-beam pointer (selects text).
			Sighting pointer (makes fine position selections).
Move and Resize Pointers			
			Move pointer (moves an object).
			Resize pointer (adjusts the upper edge of an object).
			Resize pointer (adjusts the lower edge of an object).
			Resize pointer (adjusts the right edge of an object).
			Resize pointer (adjusts the left edge of an object).
			Resize pointer (adjusts the upper left corner of an object).
			Resize pointer (adjusts the upper right corner of an object).
			Resize pointer (adjusts the lower right corner of an object).
			Resize pointer (adjusts the lower left corner of an object).
			Resize column pointer (resizes a column).
			Resize row pointer (resizes a row).
			Split pointer (splits a window horizontally).
			Split pointer (splits a window vertically).

Figure 2-1. Standard pointer shapes.

Downloaded from http://www.everyspec.com

GNOME	Motif	MS Windows	Pointer Name and Function
Waiting and Cannot Pointers			
			Working pointer (indicates that an operation is in progress).
			Cannot pointer (indicates that input is not allowed, that a drop action is not allowed at a location).
Help Pointers			
			Help pointer (indicates help mode).

Figure 2-1. Standard pointer shapes (continued).

An application does not create new pointer shapes for functions that already have a shape and does not use existing shapes for functions they were not designed to represent. If an application-specific pointer shape is created for a function not listed in figure 2-1, the following guidelines apply:

- The pointer shape should be easy to see (e.g., have high contrast with the background, not obscure other information on the screen), with a hot spot that is obvious and easy to locate.

- The pointer shape should provide a hint to its purpose and not be easily confused with other objects on the screen.

2.1.3 Pointing Device Buttons

A pointing device can have two or three buttons[1] which are used to perform actions related to object manipulation. These actions include selecting and transferring objects and displaying pop-up menus. Table 2-1 indicates the default assignment of actions to pointing device buttons. An application assigns actions to pointing device buttons as indicated in table 2-1.

GNOME, MS Windows: If the pointing device has a wheel button, it is used to scroll vertically in the window or control with focus, if the window or control supports scrolling.

[1] This document refers to the left, middle, and right buttons on the pointing device as BLeft, BMiddle, and BRight, respectively.

Downloaded from http://www.everyspec.com

Table 2-1. Assignment of actions to pointing device buttons.

GNOME	
BLeft	Selects and transfers objects.
BMiddle	Pastes objects.
BRight	Displays pop-up menus.
Java	
BLeft	Selects and transfers objects.
BMiddle	Not used.
BRight	Displays pop-up menus.
Motif	
BLeft	Selects and transfers objects.
BMiddle	Transfers objects.
BRight	Displays pop-up menus.
MS Windows	
BLeft	Selects and transfers objects.
BMiddle	Not used.
BRight	Displays pop-up menus.

NOTE: In GNOME, BMiddle and BRight perform a select action in some interface components. The desired implementation is as described above.

An application supports the following basic pointing device operations:

Press	Press and hold down a button.
Release	Release a button after it has been pressed.
Click	Press and release a button without moving the pointing device.
Double click	Press and release a button twice in rapid succession without moving the pointing device.
Move	Move the pointing device without pressing any buttons.
Drag	Move the pointing device while pressing a button.

An application supports pointing device operations such as chording (i.e., pressing multiple buttons simultaneously) only to provide redundant or shortcut access to actions available elsewhere in the application.

Downloaded from http://www.everyspec.com

2.2 Keyboard Input

2.2.1 Text Entry

An application uses standard keys for text entry. SPACE inserts a space during text entry, while RETURN inserts a carriage return in multi-line text. If no text has been selected, BACKSPACE deletes the character to the left of the text cursor (i.e., backward deletion) and DELETE deletes the character to the right of the text cursor (i.e., forward deletion). If text has been selected, both BACKSPACE and DELETE delete the selection. TAB inserts a tab or moves to the next tab stop in multi-line text.

If an application supports both insert and replace modes for text entry, INSERT toggles between them.[2] In insert mode, when users begin to type, the new character is added at the position of the text cursor. In replace mode, when users begin to type, the new character replaces the one under the text cursor.

2.2.2 Function Keys

An application supports standard function keys defined by the interface to perform navigation and selection using the keyboard. At a minimum, an application supports using TAB and the arrow keys to navigate among controls in a window, SPACE and RETURN to activate controls, and other standard function keys to move focus between windows and perform other window management functions.

> **NOTE**: In GNOME, some function keys may not perform the same action in all interface components. The desired implementation is as described above.

An application provides mnemonics and shortcut keys for navigation and selection in menus. An application also provides mnemonics for the controls in a window if keyboard navigation using the TAB and arrow keys cannot be supported. It is recommended that an application provide mnemonics for these controls as a redundant method of keyboard interaction even if the TAB and arrow keys are supported. If an application does not support some of these keys, it provides feedback (e.g., a warning sound) if these keys are used. Subsequent sections of this document identify the fixed function keys to be supported for each interface component; the implementation of variable function keys is addressed later in this section.

2.2.3 Mnemonics

A mnemonic, shown in figure 2-2, is an underlined character in the label of an interface component. Typing ALT+ the mnemonic for a component navigates to the component and selects or activates it. If the component already has focus, typing the mnemonic selects or activates it.

[2] MIL-STD 1472F indicates that overstrike mode should not be used for data entry.

Downloaded from http://www.everyspec.com

NOTE: In GNOME, this behavior is supported in pull-down and pop-up menus. However, if a control (e.g., a check box) in a window already has focus, typing the mnemonic does not select it. The desired implementation is as described above.

MS Windows: A mnemonic is not case-sensitive; users can type it in either upper or lower case.

If a component is unavailable (i.e., its label is dimmed), the mnemonic has no effect (i.e., executes no action). Section 5.4.5 addresses the use of mnemonics in menus, and section 8.2.2.1 the use of mnemonics in dialog windows.

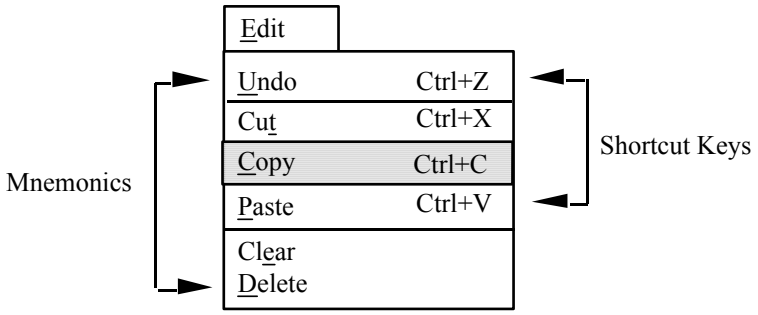


Figure 2-2. Example mnemonics and shortcut keys.

Appendix A lists the mnemonics assigned to common actions. If application-specific mnemonics are created, the following guidelines apply:

- Mnemonics should be (in order of preference) the first letter of the label, a distinctive consonant in the label, a distinctive vowel in the label, or a sequential number assigned to the label.

- Letters that are “thin”(e.g., “i” or “l”), are descenders (e.g., “g” or “y”), or are next to a letter with a descender should not be used as mnemonics because an underline is difficult to see on these letters.

2.2.4 Shortcut Keys

A shortcut key, shown in figure 2-2, consists of one or more keys that are used to activate an interface component in the window with input focus. If the component is unavailable or the window containing the option is minimized, the shortcut has no effect. A shortcut key identifies the keys to be typed, separated by a plus sign (e.g., Ctrl+C). Section 5.4.5 addresses the use of shortcut keys in menus.

Downloaded from http://www.everyspec.com

Appendix A lists the shortcut keys assigned to common actions. An application uses these shortcut keys only for common actions and does not assign these shortcuts to application-specific actions, even if it does not support the common actions.

If application-specific shortcut keys are created, the following guidelines apply:

- Shortcuts should be a single function key or special key (e.g., F4 or DELETE) or a CTRL+letter combination (e.g., CTRL+B) whenever possible.

- SHIFT+letter combinations should not be assigned as shortcuts since these keys are used for capitalization during text entry. In addition, SHIFT+symbol combinations where the symbol requires SHIFT (e.g., SHIFT+&) should not be assigned as shortcut keys. Finally, ALT+letter combinations should not be assigned as shortcuts since these keys are used as mnemonics.

- SHIFT should be included in shortcuts for actions that extend or complement the actions assigned to the unmodified key(s). For example, in GNOME, CTRL+W closes a window, and SHIFT+CTRL+W closes all windows; also, CTRL+A performs a “select all” action, and SHFT+CTRL+A performs a “deselect all” action.

- CTRL should be included in shortcuts for actions that represent larger-scale versions of the actions assigned to the unmodified key(s). For example, HOME navigates to the beginning of a line, and CTRL+HOME navigates to the beginning of the data.

2.2.5 Variable Function Keys

If desired, an application can implement variable function keys that perform different actions depending on the current state of the application. The actions are arranged in the same order as the portion of the keyboard being used (e.g., F1-F10, number pad) and displayed as a series of buttons in a toolbar or palette, as shown in figure 2-3. The button label changes whenever the action assigned to a key changes. An application limits the actions to no more than two per key and defines keys that return to the previous level and to the set of base-level actions. Users can execute the actions only when the window displaying the buttons has input focus; the buttons are grayed out (to indicate their unavailability) when the window does not have focus. In addition, the actions mapped to the keys do not conflict with the key mappings in appendix A.

Downloaded from http://www.everyspec.com

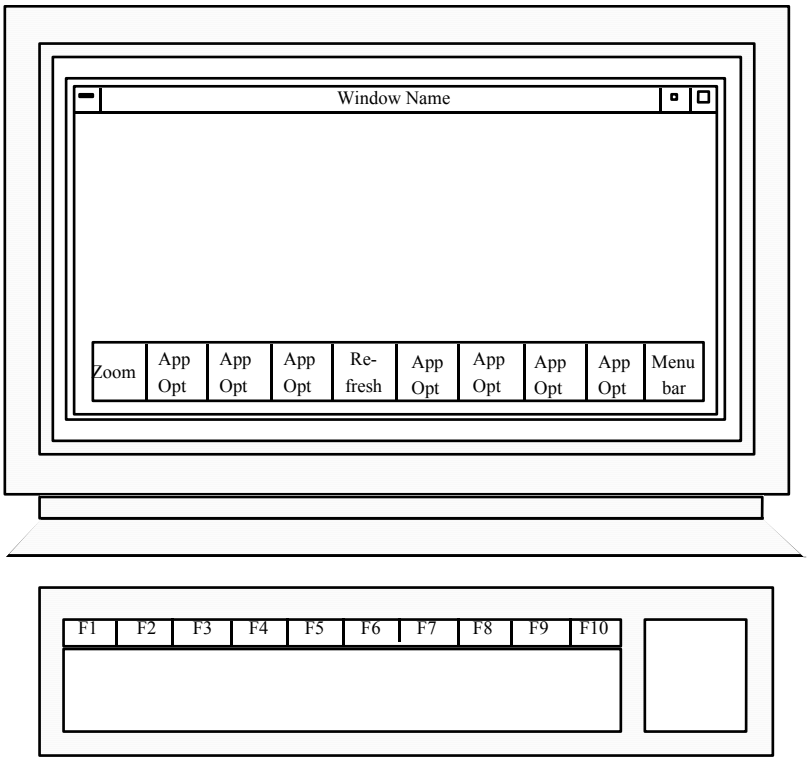


Figure 2-3. Example variable action keys.

Downloaded from http://www.everyspec.com

2.3 Interchangeability Between Input Devices

Users are able to interact with an application either with the pointing device or from the keyboard. An application supports interchangeability between input devices as defined in this document so that users can select the device that is most appropriate to the task being performed. An application provides redundancy for those functions that are visible as text (e.g., menu options, push button actions); redundancy is not required for those functions that can only be performed efficiently with a pointing device (e.g., drawing a geometric figure, placing a tactical symbol on a map).

Downloaded from http://www.everyspec.com

2.4 Alternate Input Devices

Developers considering the use of a hardware configuration that includes an input device other than a mouse or trackball should submit their requests to the appropriate configuration management board for approval prior to implementation. If the use of an alternate input device is approved, the manner in which users interact with the device (e.g., for navigation and selection) is consistent with the interaction models presented in section 3 of this document.

Downloaded from http://www.everyspec.com

3. User-Computer Interaction

3.1 Navigation

Navigation refers to the ways users move the pointer or location cursor to the object(s) with which they want to interact. This section addresses navigation among the objects in a window; section 4.1.2 discusses navigation between windows.

3.1.1 Pointing Device Navigation

Users perform pointing device navigation by moving the pointer to an object. Navigating to an object and pressing BLeft moves focus to the object. Pressing BLeft on an object that is unavailable does not move focus to it.

> GNOME, Java, MS Windows: Pointing device navigation may be supplemented with mouse-over feedback that changes the appearance of an object (e.g., its background color) when users move the pointer over the object.

3.1.2 Keyboard Navigation

Users perform keyboard navigation by pressing individual or combinations of keys to move focus from one object to another. The object with focus is identified by a location cursor. Only one location cursor appears in a window at any time. The shape of the location cursor depends on the type of object with focus. Figure 3-1 provides examples of the location cursors in MS Windows. An element cursor is a rectangular border around controls such as radio buttons, check boxes, and push buttons. A text cursor is a vertical bar that is displayed in the text box with focus and indicates where typed text will be displayed. An application uses the standard location cursors provided by the interface and creates new cursors only when the standard ones do not support application requirements.

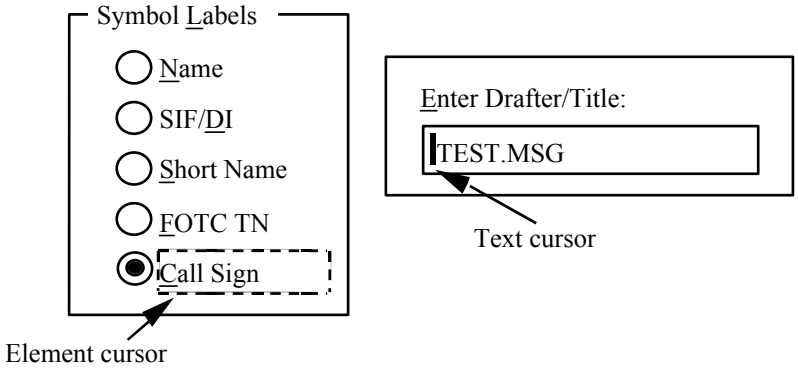


Figure 3-1. Example location cursors in MS Windows.

An application defines tab groups in a window based on the order in which users are expected to interact with the controls in the window. This sequence is left to right, top to

Downloaded from http://www.everyspec.com

bottom in the window, with wrapping between the last and first tab groups. When the window opens, the location cursor is placed on the top leftmost control in the window or on the control with which users are expected to interact first. When a window loses and then regains input focus, the location cursor should be placed on the control that last had focus, provided that the control remains available for selection; otherwise, the location cursor should be placed on the control users are most likely to select in the window.

TAB and SHIFT+TAB move the location cursor to the next and previous tab group in a window. UP, DOWN, LEFT, and RIGHT move the location cursor between the components within a tab group.

> **NOTE**: In GNOME, both TAB and the arrow keys may navigate between tab groups. The desired implementation is as described above.

GNOME, Java, and Motif: If a component that accepts tabs (e.g., a multi-line text box) has focus, CTRL+TAB and SHIFT+CTRL+TAB move the location cursor to the next and previous tab group.

The location cursor should remain visible as it moves within a window; i.e., there should be no “invisible” tab groups that cause the location cursor to temporarily disappear as users navigate in the window.

The location cursor skips a tab group if none of the controls in the group can have focus. Moving focus to a control does not activate any other action associated with the control. If a separate label is included with a control such as a text box or list box, focus is assigned to the control, not the label. Focus can be assigned to a conditional control only after the parent control is selected and the child control becomes available.

Downloaded from http://www.everyspec.com

3.2 Selection

Selection refers to the ways in which users identify the object(s) with which they want to interact. Select state is indicated by changing the appearance of the selected object (e.g., by highlighting, displaying "handles"). Select state is shown whenever the window containing the selected object(s) has focus.

3.2.1 Pointing Device Selection

An application supports single, range, and disjoint selection models using the pointing device listed in table 3-1. Sections 4 through 7 describe how selection is performed in individual interface components.

GNOME, MS Windows: Click-through selection enables users to assign focus to a window and select an object in it with one click, rather than having to click on the window to assign focus and then click on the object to select it. If an application supports click-through selection, it should do so for interface components whose actions are not destructive or difficult to cancel or undo.

Table 3-1. Pointing device selection models.

Single Selection	Click BLeft on an object selects the object.
Range Selection	SHIFT+click BLeft on an object extends a selection to the object.
Disjoint Selection	CTRL+click BLeft on an object toggles the select state of the object.

NOTE: In GNOME, BMiddle and BRight also perform a single selection in some interface components, and range selection may not be supported in some interface components. In Java, BRight also performs a single selection in some interface components. The desired implementation is as described above.

3.2.2 Keyboard Selection

An application supports single, range, and disjoint selection models using the keyboard listed in table 3-2. Sections 4 through 7 describe how selection is performed in individual interface components.

Motif, MS Windows: Two modes, normal and add, are available for performing a disjoint selection. In normal mode, focus and select state move together when the arrow keys are used. In add mode, focus moves independently of select state, and users have to perform an explicit select action. SHIFT+F8 toggles between normal and add modes if both modes are supported.

Downloaded from http://www.everyspec.com

Table 3-2. Keyboard selection models.

Single Selection	SPACE selects an object.
Range Selection	SHIFT+SPACE extends a selection to an object.
Disjoint Selection	CTRL+SPACE toggles the select state of an object.

NOTE: In GNOME, range and disjoint selection may be performed differently or may not be supported in some interface components. The desired implementation is as described above.

Downloaded from http://www.everyspec.com

3.3 Activation

Activation refers to the ways users invoke an action on the currently selected object(s). An application supports the standard methods for invoking an action and executing a default action.

Placing the pointer on an object that performs an action (e.g., an interface component such as a push button or menu option) and pressing BLeft moves focus to the object; releasing BLeft activates the object. If the pointer is moved off the object while BLeft is pressed, the object is not activated. When an object that performs an action has focus, SPACE activates it.

ESC cancels the action being executed and returns the object to its state prior to the action. For example, ESC in the window with input focus is equivalent to activating the Cancel push button in the window.

If a default action is assigned to an object, double clicking BLeft on the object selects the object and executes the action. For example, double clicking on a folder icon selects the folder and opens a window displaying its contents. If a default action is assigned to an object in a window, RETURN invokes the action when the window has focus. The default action for an object is available as a redundant method for performing the basic action on the object; e.g., users can open a file either by double clicking on its icon or by selecting the Open option in the pop-up menu for the icon.

Downloaded from http://www.everyspec.com

3.4 Transfer

3.4.1 Clipboard Transfer

Clipboard transfer allows users to move, copy, and link objects by transferring them from their current location (i.e., source) to a temporary clipboard and then from the clipboard to a new location (i.e., destination). An application supports clipboard transfer for objects that can be duplicated or inserted into other objects.

Users execute Cut and Paste commands to perform a clipboard move, and Copy and Paste commands to perform a clipboard copy or a clipboard link. Cut removes an object from the source and places it in the clipboard. Copy duplicates an object and places the duplicate in the clipboard. Paste copies the contents of the clipboard to the destination. A Paste command can be executed only when there are contents in the clipboard and they can be pasted at the destination. Pasting an object from the clipboard does not select the object. An object remains in the clipboard until another object is cut or copied into it. A Paste command transfers the object in its native form to the destination.

> GNOME, Java, and MS Windows: A Paste Special command transfers the contents of the clipboard in a user-selected form.

An application uses the vocabulary, mnemonics, and shortcut keys for clipboard transfer commands listed in appendix A. Access to these commands can be provided in a pull-down or pop-up menu, in a toolbar, and/or using shortcut keys. In addition, an application provides the capability to undo the results of a clipboard transfer.

3.4.2 Drag Transfer

Drag transfer allows users to move, copy, and link objects by dragging them from one location to another. An application supports drag transfer as a redundant method for transferring objects that users can manipulate. Users should be able to drag an object to any destination that accepts the type of data contained in the object. This section describes how to perform drag transfer; section 15 provides additional information on the topic.

Drag operations. To perform a default drag transfer, users drag an object using BLeft and drop it at the destination. The destination determines whether the operation performed by a default transfer is a drag copy or drag move.

To perform a nondefault drag transfer, users drag one or more objects using SHIFT and/or CTRL with BLeft. SHIFT+BLeft forces a drag move, CTRL+BLeft a drag copy, and CTRL+SHIFT+BLeft a drag link.[1]

[1] A drag link transfers an object to a destination and maintains a link to the original object so that changes to the object are reflected at the destination.

Downloaded from http://www.everyspec.com

MS Windows: BRight can also used to perform a nondefault drag transfer. When users release BRight at a destination, a pop-up menu listing available transfer options (e.g., Move Here, Copy Here, Link Here) is displayed. If the menu contains a default option, it is the transfer operation that would have occurred if BLeft had been used.

Users cancel a drag operation by pressing ESC, by releasing BLeft when the pointer is not over a valid destination, or by dragging an object back onto itself. In addition, an application provides the capability for users to undo the results of a drag transfer. If a drag transfer cannot be undone (e.g., it moves data between different applications), an application requests that users confirm the transfer before completing the action.

Drag transfer and select state. Drag transfer can be performed on a single object or a collection of objects. Dragging a collection of selected objects drags the entire collection. If users transfer a collection of objects to a destination that is valid for only some of the objects, an application displays a message giving users the option to cancel the operation or complete it for those objects that can be dropped. If desired, an application can allow users to specify the default behavior they want to take in this situation. Dragging an object within or into a scrollable area automatically scrolls the area when the object is dragged to the edge of the area.

Drag icons. During a drag transfer, the pointer changes appearance (i.e., into a drag icon) to indicate that a drag operation is in progress. An application uses the standard drag icons and creates new icon shapes only when the standard ones do not support application requirements. The drag icon shows the type of drag operation being performed and the type of object being dragged. If the modifier key changes during the drag (e.g., from SHIFT to CTRL), the drag icon changes accordingly and the new action is executed when BLeft is released.

Transfer feedback. A representation of the object being transferred remains at its original location until the transfer operation is completed. If the transfer is successful, the object is displayed at the destination, the drag icon is removed, and the data associated with the object is transferred from the source to the destination. The object may be placed where it is dropped, or its placement may be based on the current context at the destination. If the transfer does not occur immediately, the pointer changes to a “working” shape until the transfer is complete. If the transfer fails, the object remains at the source and the drag icon is removed; if desired, an application can display a message window explaining why the transfer failed.

3.4.3 Primary and Quick Transfer (Motif)

If desired, an application can support primary and quick transfer. Primary transfer allows users to transfer a selectable object (e.g., editable text) directly to a destination without using the clipboard for intermediate storage. When BMiddle is used to perform a primary transfer, the default is a copy operation. SHIFT and CTRL are used with BMiddle to perform a nondefault primary transfer. SHIFT+BMiddle forces a primary move,

Downloaded from http://www.everyspec.com

CTRL+BMiddle forces a primary copy, and CTRL+SHIFT+BMiddle forces a primary link. If primary transfer commands (i.e., Primary Move, Primary Copy, Primary Link) are included in a pull-down or pop-up menu, they execute the actions and use the shortcut keys listed in appendix A. Transferring an object by performing a primary copy or a primary link does not select the object; however, transferring an object via a primary move does.

Quick transfer allows users to temporarily select an object and immediately transfer it to a new location. Quick transfer is available in text components that are normally not selectable (e.g., the label for a text box). When ALT+BMiddle is used to perform a quick transfer, the default is a copy operation. ALT, SHIFT, and CTRL are used with BMiddle to perform a nondefault quick transfer. ALT+SHIFT+BMiddle forces a quick move, ALT+CTRL+BMiddle forces a quick copy, and ALT+SHIFT+CTRL+BMiddle forces a quick link. When a quick transfer is performed, the object being transferred is not selected when it is displayed at the destination location.

Downloaded from http://www.everyspec.com

3.5 Interactive Control

3.5.1 Object-Action Paradigm

GNOME, Java: An application bases user interactions on an object-action paradigm, except in "instant apply" windows where changes to values or settings are applied immediately, without executing an action to commit them.

Motif, MS Windows: An application bases user interactions on an object-action paradigm. Windows are "explicit apply" (i.e., users first select an object so that it has focus and then perform an action on that object).

3.5.2 Minimizing User Error

An application is designed to minimize the possibility of user error. The application provides visual and behavioral cues that prevent users from performing incorrect actions. The application encourages users to perform the correct actions in a window by graying out and disabling controls whenever they are inappropriate to the task being performed, rather than relying on error messages to inform users after they have made an error.

3.5.3 Immediate Feedback

An application provides users with an immediate indication that their action has been accepted. The following guidelines apply to response time:

- Some visible response (e.g., a button is highlighted, the pointer changes shape) should be made within 0.1 seconds of any user action.

- When a user action takes more than 2 seconds but less than 10 seconds to process, the pointer should change to a "working" shape immediately after the user initiates the action and remain this shape until the action is complete.

- When a user action takes more than 10 seconds to process, a message window should be displayed immediately after the user initiates the action and remain displayed until the action is complete.

These guidelines are recommended response times; it is expected that actual times will be determined by factors such as the hardware configuration being used and the amount of processing being performed.

An application informs users when they can and cannot take an action. An application provides visual cues indicating when it can accept input (e.g., the arrow pointer is displayed), when it is temporarily unavailable (e.g., the pointer changes to a "working" shape), and when it is unavailable during extended processing (e.g., a message window is displayed). In addition, the appearance of the objects displayed by an application provides an indication of their availability. For example, menu options provide feedback

Downloaded from http://www.everyspec.com

concerning the actions that can be performed on selected object(s). If a menu option is not appropriate for an object that has been selected or if a menu option cannot be executed for some reason, a visual cue (e.g., graying the option) is provided to indicate its unavailability.

When an operation requires several actions to complete, an application prompts users with the actions that need to be performed. For example, when users select a Zoom option from a pull-down menu, an application provides feedback (e.g., in the status bar of the window) explaining how to select the area to be zoomed.

An application ignores user actions made during periods when input cannot be accepted. An application disables the pointing device and/or keyboard when input may have destructive effects (e.g., the "working" or "cannot" pointer is displayed when an application is temporarily unavailable or input cannot be accepted at the location of the pointer). Disabling is particularly important so that any input made by users is not stored and then executed when an application becomes available again. Although an application does not allow users to override disabling, users are able to stop a process if desired (e.g., by activating a Cancel push button).

3.5.4 Error Detection

If users attempt to execute an invalid action, an application displays a message describing the action that is invalid. Error feedback is immediate (i.e., occurs within 2 seconds of error detection), is visual and/or auditory, and explains the nature of the error made. The following guidelines apply to error feedback:

- If users make multiple errors within a single action, they should be notified of each occurrence of an error.

- If desired, a Help push button that navigates to context-sensitive help can be included in the window that presents an error message so users can obtain additional information about how to respond to the error.

- If users repeat an error, the feedback provided by an application should be different (e.g., auditory with diagnostics or help also provided) to show users that their attempted corrective action was, in fact, processed.

- Users should only have to correct an invalid action and not repeat the entire action sequence. After correcting the error, users should then execute the same action (e.g., activate a push button) for re-entry that was used for the original entry.

3.5.5 Explicit Destruction

An application requires users to confirm a destructive action (i.e., one with irreversible negative consequences such as deleting a data file) before the action is executed. Users do not have to confirm an action that closes a window unless the action will cause

Downloaded from http://www.everyspec.com

significant data loss. In general, closing a primary window is a potentially destructive action because this type of window usually generates data that should not be lost. Closing a dialog window is usually nondestructive because these windows request user input that does not have to be saved. Section 4.1.1 provides definitions of primary and dialog windows.

When a confirmation window is displayed, the window in which the destructive action is about to be taken remains open until users confirm the action. The window is not dismissed when the prompt is presented and then redisplayed if users fail to confirm the destructive action.

3.5.6 General Undo Capability

An application allows users to undo the action most recently made unless the action was one requiring explicit destruction. Users can undo drag transfer and clipboard transfer actions and can recover information that was deleted from a window (e.g., when Delete in an Edit menu was selected). The application can support an undo capability by including Undo/Redo options in an Edit menu or by providing Reset/Revert push buttons in a window. If an undo capability cannot be provided, the application labels irreversible actions as such and clearly separates actions that are reversible from those that are not.

An application can provide a single-level (i.e., reverse the most recently executed action) or multi-level (i.e., reverses a series of previous actions) undo capability as appropriate to the function performed. If a multi-level capability is provided, an application should also support a Redo action so users can redo the most recently undone action.

3.5.7 Use of Processing Modes

A processing mode is a state where a user action has different results depending on the mode currently in effect. An application uses processing modes when defining how users interact with some interface components. For example, the mode assigned to a dialog window determines the extent to which users can interact with other windows while the dialog window is displayed (see section 4.2.2).

An application is designed so that whenever possible, the same action has the same effect whenever it is executed by users. If an application-specific processing mode is required for a particular function, users are provided with a visual cue to indicate when the mode is in effect. For example, when users select a drawing tool from a toolbar, the tool button remains highlighted to indicate the mode in which users are operating. An application provides feedback showing the actions that can be taken (e.g., displays as available only the menu options applicable to the currently selected object) and includes an easy and obvious means for exiting from the mode.

Downloaded from http://www.everyspec.com

This page intentionally left blank.

Downloaded from http://www.everyspec.com

4. Window Management

4.1 Window Components

4.1.1 Appearance

Window types. Windows differ in the types of information they present to users. A primary window provides the work area where users view and interact with information. A dialog window presents supplementary or supporting data or operations and is used for short-term interactions related to subtask or function control. A message window is used to present messages to users. Dialog and message windows are secondary windows that are dependent on a primary window (or other secondary window). GNOME, Java, Motif, and MS Windows support several variations in these basic window types.

GNOME, Java: A utility window is a dialog window that remains open when the primary window to which it relates closes (see section 8.2.3).

Java, MS Windows: A document window is a primary window that opens within another primary window rather than on the desktop (see section 10.1.2).[1] A pop-up window, which has no title bar or window controls, is used to display a splash screen and present context-sensitive help such as tooltips (see section 9.1.3).

Motif: A menu window is a dialog window that displays the contents of a torn-off menu (see sections 5.1.1, 5.1.2, and 5.1.3).

A window contains a frame with components for accessing window management functions and a client area where users view and interaction with the information in an application.

GNOME, Motif, and MS Windows: A window manager on the platform controls the size, placement, and operation of windows and defines the appearance and behavior of window components in each type of window in an application. This section describes the components of primary, dialog, and message windows provided by these interfaces.

Java: Window management components are provided by the operating system of the platform on which an application is installed. The platform supplies the window frame and title bar in primary and secondary windows and controls behaviors such as resizing, dragging, minimizing, positioning, and layering. However, the platform does not automatically close or minimize a secondary window when the associated primary window is closed or minimized (see section 4.2.1). As a result, an application keeps track of secondary windows and supplies these behaviors whenever they are not provided by the platform. The Java Look and Feel Guidelines provide additional information on window management in

[1] A document window in MS Windows is considered to be a primary window unless otherwise indicated.

Downloaded from http://www.everyspec.com

primary and secondary windows and address the construction of utility, document, and pop-up windows.

Components by window type. The availability of window management functions varies by window type.[2] Table 4-1 lists the window components in primary, dialog, and message windows, and figures 4-1, 4-2, and 4-3 show the placement of these components in a primary window. An application supports the window components and functions as defined here.

GNOME: The Control menu button is on the left in the title bar, and the Minimize, Maximize, and Close buttons, if present, are on the right.

Motif: The Control menu button is on the left in the title bar, and the Minimize and Maximize buttons, if present, are on the right.

MS Windows: The title bar icon, if present, is on the left in the title bar, and the Minimize, Maximize, What's This?, and Close buttons, if present, are on the right.

Table 4-1. Window components in primary, dialog, and message windows.

GNOME	
Primary Window	Title bar, Control menu button, Minimize button, Maximize button, Close button, resize border, size grip
Dialog Window	Title bar, Control menu button, Close button
Message Window	Title bar, Control menu button, Close button
Motif	
Primary Window	Title bar, Control menu button, Minimize button, Maximize button, resize border
Dialog Window	Title bar, Control menu button, Maximize button (optional), resize border (optional)
Message Window	Title bar, Control menu button
MS Windows	
Primary Window	Title bar, title bar icon, Minimize button, Maximize button, Close button, resize border, size grip (optional)
Dialog Window	Title bar, What's This? button (optional), Close button
Message Window	Title bar, Close button

NOTE: In GNOME, dialog and message windows differ in the window management functions they support. The desired implementation is as described above.

[2] This document describes GNOME window components assuming that a default theme such as Bluecurve or Crux has been selected.

Downloaded from http://www.everyspec.com

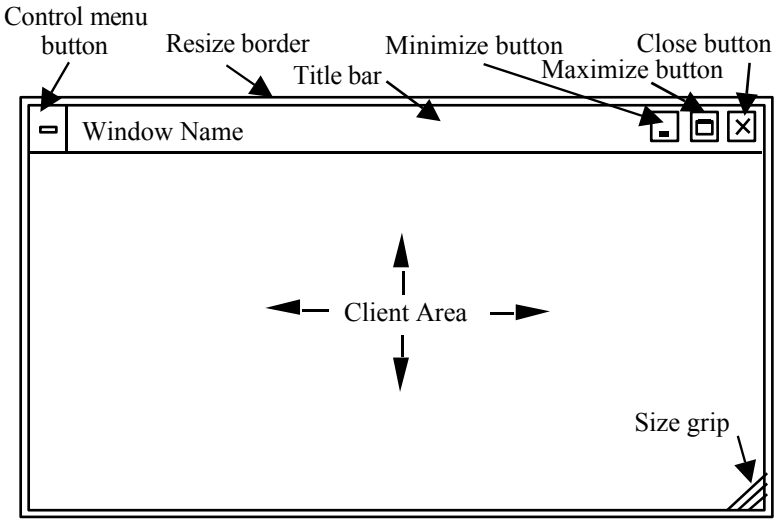


Figure 4-1. Primary window components in GNOME.

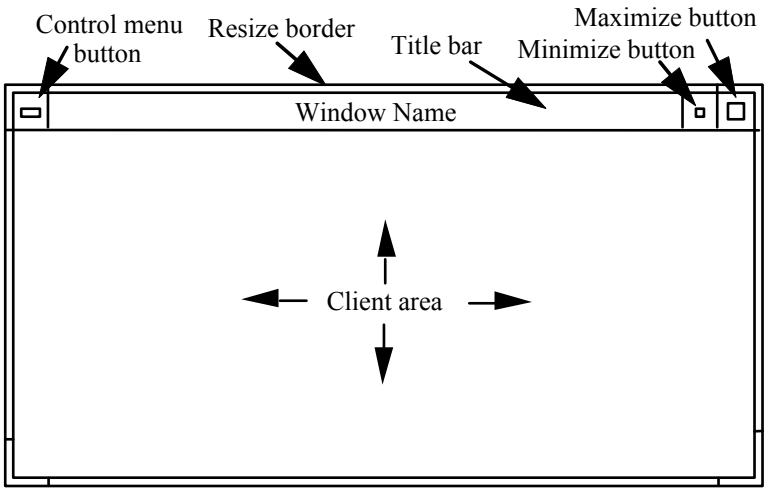


Figure 4-2. Primary window components in Motif.

Downloaded from http://www.everyspec.com

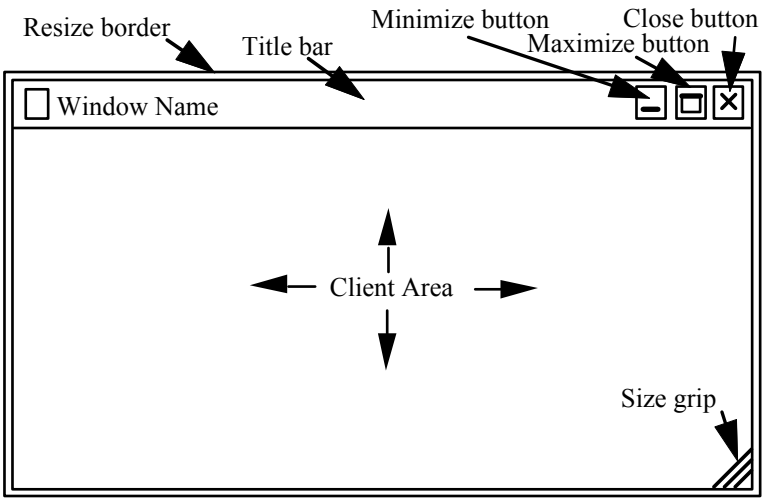


Figure 4-3. Primary window components in MS Windows.

Control menu. A window includes a Control menu that provides access to window management functions for each type of window.[3] Table 4-2 lists Control menu contents in primary, dialog, and message windows. An option may be included in a Control menu but be unavailable for selection based on window type (e.g., a message window cannot be minimized) or window state (e.g., an open window cannot be restored). An application includes window management functions in a Control menu as defined here and uses the vocabulary, mnemonics, and shortcut keys (if present) listed in appendix A.

GNOME, MS Windows: The options in a Control menu are left-aligned based on their text labels, with sufficient space preceding the options to display an icon.

Motif: The options in a Control menu are left-aligned based on their text labels.

[3] GNOME, Motif, and MS Windows support a Window menu that provides access to window management functions. The specifications presented here refer to this window component as a Control menu in order to minimize confusion with the Window pull-down menu that may be included in the menu bar of a window.

Downloaded from http://www.everyspec.com

Table 4-2. Control menu contents.

	GNOME	Motif	MS Windows
Primary Window	Minimize	Restore (unavailable)	Restore (unavailable)
	Maximize/Unmaximize	Move	Move
	Roll Up/Unroll	Size	Size
	Move	Minimize	Minimize
	Resize	Maximize	Maximize
	<separator>	Lower,	<separator>
	Close	<separator>	Close
	<separator>	Occupy Workspace	
	Put on All Workspaces	Occupy All Workspaces	
	Move to Workspace 1	Unoccupy Workspace (unavailable)	
	Move to Workspace 2	<separator>	
	Move to Workspace 3	Close	
	Move to Workspace 4		
Dialog Window	Minimize (unavailable)	Restore (unavailable)	Move
	Maximize (unavailable)	Move	Close
	Roll Up/Unroll	Size (optional)	
	Move	Maximize (optional)	
	Resize (unavailable)	Lower	
	<separator>	<separator>	
	Close	Occupy Workspace	
	<separator>	Occupy All Workspaces	
	Put on All Workspaces	Unoccupy Workspace (unavailable)	
	Move to Workspace 1	<separator>	
	Move to Workspace 2	Close	
	Move to Workspace 3		
	Move to Workspace 4		
Message Window	Minimize (unavailable)	Restore (unavailable)	Move
	Maximize (unavailable)	Move	Close
	Roll Up/Unroll	Lower	
	Move	<separator>	
	Resize (unavailable)	Occupy Workspace	
	<separator>	Occupy All Workspaces	
	Close	Unoccupy Workspace (unavailable)	
	<separator>	<separator>	
	Put on All Workspaces	Close	
	Move to Workspace 1		
	Move to Workspace 2		
	Move to Workspace 3		
	Move to Workspace 4		

NOTE: In GNOME, dialog and message windows differ in the window management functions they support. The desired implementation is as described above.

Downloaded from http://www.everyspec.com

Input focus and stacking order. Input focus identifies the window or object within a window to which keyboard input is directed. Two modes are available for assigning focus to a window. In explicit mode, focus is assigned by explicit user action, either with the pointing device or from the keyboard. In implicit mode, focus moves with the pointer, and keyboard events are sent to the window that the pointer is in. If stacking order is automatic, assigning focus also surfaces the window (i.e., raises it to the top of the window stack). If stacking order is manual, clicking on the window frame is required to surface it. These specifications assume that focus is explicit and stacking order is automatic, unless otherwise indicated.

Regardless of the focus policy that may be in effect, only one application window has input focus at any time. Within that window, only one object or collection of objects (e.g., resulting from a range or disjoint selection) has focus at any time. The window with focus is indicated by highlighting (e.g., changing the color of the title bar).

4.1.2 Behavior

Tables 4-3 and 4-4 list pointing device and keyboard interaction methods for performing window management functions, respectively. An application uses these methods whenever the functions described here are available in a window.

Downloaded from http://www.everyspec.com

Table 4-3. Pointing device interaction methods for windows.

	GNOME	Motif	MS Windows
Displaying/Dismissing a Control Menu			
Clicking BRight on the title bar displays the Control menu.	X	X	X
Clicking BLeft or BRight on the Control menu button displays the Control menu.	X	X	
Clicking BRight on the window frame displays the Control menu.	X	X	
ALT+clicking BRight inside a window displays the Control menu.	X		
Clicking BLeft on a title bar icon displays the Control menu.			X
Clicking BLeft or BRight outside the Control menu dismisses it.	X	X	X
Clicking BLeft on the Control menu button a second time dismisses the Control menu.	X		
Clicking BLeft on the title bar icon a second time dismisses the Control menu.			X
Assigning Focus to a Window			
Clicking BLeft on a window gives it focus.	X	X	X
Moving a Window			
Dragging the title bar using BLeft moves the window.	X	X	X
ALT+dragging inside a window using BLeft moves the window.	X	X	
Resizing a Window			
Dragging the resize border using BLeft resizes the window.	X	X	X
Dragging the size grip using BLeft resizes the window.	X		X
ALT+dragging near the corner inside a window using BMiddle resizes the window.	X		
Maximizing a Window			
Clicking BLeft on the Maximize button maximizes the window.	X	X	X
Double clicking BLeft on the title bar maximizes the window.		X	X
Clicking BLeft on the Maximize button restores a maximized window to its size and location before being maximized.	X	X	X
Double clicking BLeft on the title bar restores a maximized window to its size and location before being maximized.		X	X
Minimizing a Window			
Clicking BLeft on the Minimize button minimizes the window.	X	X	X
Closing a Window			
Clicking BLeft on the Close button closes the window.	X		X
Double clicking BLeft on the Control menu button closes the window.		X	
Double clicking BLeft on the title bar icon closes the window.			X

Downloaded from http://www.everyspec.com

Table 4-3. Pointing device interaction methods for windows (continued).

	GNOME	Motif	MS Windows
Rolling Up a Window			
Double clicking BLeft on the title bar reduces the size of a window so only the title bar is visible.	X		
Double clicking on the title bar restores a rolled up window to its previous size.	X		
Invoking Help Mode			
Clicking BLeft on the What's This? button invokes context-sensitive help in a window.			X

Downloaded from http://www.everyspec.com

Table 4-4. Keyboard interaction methods for windows.

	GNOME	Motif	MS Windows
Displaying/Dismissing a Control Menu			
ALT+SPACE displays the Control menu in the window with focus.	X	X	X
UP and DOWN navigate between options in the Control menu, with wrapping from the bottom to the top option.	X	X	X
RETURN activates an option in the Control menu and dismisses the menu.	X	X	X
SPACE activates an option in the Control menu and dismisses the menu.	X	X	
ESC dismisses the Control menu without activating an option.	X	X	X
Assigning Focus to a Window			
ALT+ESC moves focus forward among primary and secondary windows.	X		
ALT+ESC and SHIFT+ALT+ESC move focus forward and backward among window families.		X	X
ALT+TAB displays a pop-up window for switching among currently running applications.	X		X
ALT+TAB and SHIFT+ALT+TAB move focus forward and backward among primary windows.		X	
ALT+F6 and SHIFT+ALT+F6 move focus forward and backward among secondary windows.		X	
Moving a Window			
Activating the Move option in a Control menu and then using the arrow keys moves a window; RETURN sets the window at its new location.	X	X	X
Activating the Move option in a Control menu and then using the arrow keys moves a window; SPACE sets the window at its new location.	X		
Resizing a Window			
Activating the Resize/Size option in a Control menu and then using the arrow keys resizes a window; RETURN sets the window at its new size.	X	X	X
Activating the Resize option in a Control menu and then using the arrow keys resizes a window; SPACE sets the window at its new size.	X		
Maximizing a Window			
Activating the Maximize option in a Control menu maximizes the window.	X	X	X
Activating the Unmaximize/Restore option in a Control menu restores a maximized window to its size and location before being maximized.	X	X	X

Downloaded from http://www.everyspec.com

Table 4-4. Keyboard interaction methods for windows (continued).

	GNOME	Motif	MS Windows
Minimizing a Window			
Activating the Minimize option in a Control menu minimizes a window.	X	X	X
Closing a Window			
Activating the Close option in a Control menu closes the window.	X	X	X
Rolling Up a Window			
Activating the Shade option in a Control menu reduces the size of a window so only the title bar is visible.	X		
Activating the Unshade option in a Control menu restores a rolled up window to its previous size.	X		
Moving Windows Between Workspaces			
Activating the Put on All Workspaces option in a Control menu places a window in all workspaces.	X		
Activating the Move to Workspace # option in a Control menu moves a window to that workspace.	X		
Activating the Only on This Workspace option in a Control menu places the window in that workspace.	X		
Activating the Occupy Workspace option in a Control menu places a window in the workspace(s) selected by the user.		X	
Activating the Occupy All Workspaces option in a Control menu places a window in all workspaces.		X	
Activating the Unoccupy Workspace option in a Control menu removes a window from the current workspace.		X	
Lowering a Window			
Activating the Lower option in a Control menu lowers the window at the top of the stacking order to the bottom of the stack.		X	

Downloaded from http://www.everyspec.com

4.2 Window Families

4.2.1 Parent-Child Relationships

A window family consists of a main or parent window and one or more subordinate or child windows. A primary window is the parent for all other windows in the family. A secondary window is the child of its parent primary window and may also be the parent of other secondary windows. These specifications assume that a dialog window can have children, but a message window cannot.

When a primary window is minimized, it and all of its child windows are minimized, and processing in the window continues. When the window is restored, it and all of its child windows that were open when the window was minimized are restored. Each primary window in an application is minimized separately.

When a primary window is raised or lowered, it and all of its child windows are raised or lowered with it. When a primary window is closed, it and all of its child windows are closed, and processing in the primary window stops. When the last primary window in an application is closed, processing by the application ends.

When a secondary window is opened, it is placed in front of its parent window, and the parent remains open. When a secondary window is closed, it and all of its child windows are closed but its parent window is not affected.

4.2.2 Window Mode

The mode assigned to a child window determines the extent to which users can interact with other windows while the window is open. If a window is modeless, users can interact with other windows. If a window is modal, users are restricted from interacting with other application and/or system windows. An application modal window does not allow interaction with any window created by the same application even if the application has multiple primary windows, and a system modal window does not allow interaction with any other open window. Figure 4-4 illustrates the interaction restrictions for a message window in Application A under each type of modality.

Downloaded from http://www.everyspec.com

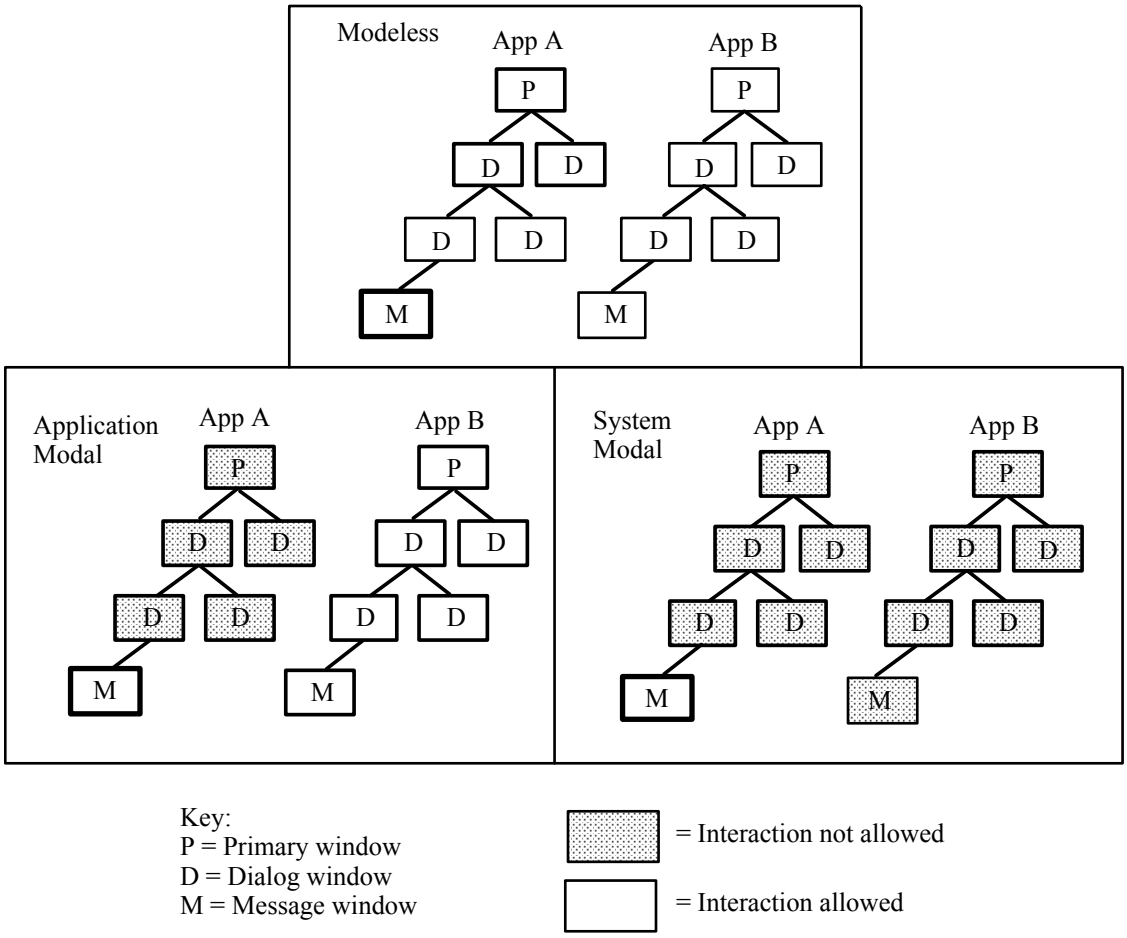


Figure 4-4. Interaction restrictions in modeless and modal windows.

A secondary window is modeless whenever possible so users can continue to interact with an application while the window is open. A secondary window is application modal only when an application requires user input in order to continue processing. A secondary window is system modal only when a critical error is detected or an unrecoverable condition exists.

Downloaded from http://www.everyspec.com

4.3 Window Buttons and Window Icons

4.3.1 Window Buttons (GNOME, MS Windows)

4.3.1.1 Appearance

A window button provides a visual representation (e.g., in the window list or on the taskbar) of each application window that is open on the desktop. The button contains a graphic image and a label, as shown in figure 4-5. The button has the same graphic as that used for the application icon and the same name as the associated window. A window button includes a Control menu with the same options in the same order as the Control menu for a window.

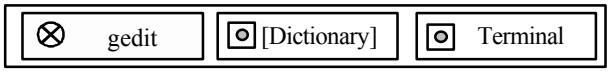


Figure 4-5. Example window buttons.

GNOME: A window button is displayed in the window list located in the bottom edge panel on the desktop when a window that can be minimized opens. The button is removed from the window list when the window is closed.

MS Windows: A window button is displayed in the taskbar on the desktop when a primary window in an application opens. The button is removed from the taskbar when the window is closed.

The window button has a recessed appearance when the associated window has focus and a normal appearance when the window does not have focus.

GNOME: The button has a normal appearance and the window name is enclosed by square brackets when the window is minimized.

MS Windows: The button has a normal appearance when the window is minimized.

4.3.1.2 Behavior

Clicking BRight on a window button displays its Control menu. Clicking BLeft on a window button restores the associated window to its size and location before being minimized. Clicking BLeft on the button for the window with focus minimizes the window. Clicking BLeft on the button for a window that is not minimized and does not have focus gives it focus.

Activating the Unminimize/Restore option in the Control menu for a window button restores the window to its previous size and location.

Downloaded from http://www.everyspec.com

4.3.2 Window Icons (Motif)

4.3.2.1 Appearance

A window icon provides a visual representation of an application window on the desktop. The window icon consists of a graphic image and a label, as shown in figure 4-6. The icon has the same graphic as that used for the application icon and the same name as the associated window. A window icon includes a Control menu with the same options in the same order as the Control menu for a window.

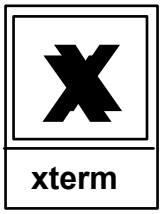


Figure 4-6. Example window icon.

A window icon can be displayed either in a window icon box or directly on the desktop. If an icon box is present, a window icon is displayed in the box when a primary window in an application opens.

The icon is dimmed (e.g., grayed out) when the associated window has focus. The icon has a normal appearance when the window is minimized. The icon is removed from the icon box when the window is closed. If an icon box is not present, a window icon is displayed on the desktop only when the associated window is minimized. The icon is removed from the desktop when the window is restored.

4.3.2.2 Behavior

Clicking BLeft or BRight on a window icon displays its Control menu. Double clicking BLeft on a window icon restores the associated window to its size and location before being minimized.

When a window icon has focus, ALT+SPACE displays its Control menu. Activating the Restore option in the menu restores the window to its size and location before being minimized.

Downloaded from http://www.everyspec.com

4.4 Window Management Considerations

4.4.1 Window Size

A primary window is large enough to display a typical amount of data in the client area but it does not fill the entire screen (so users are able to access the windows in other applications). A dialog window is smaller than its parent primary window and does not fill the entire screen. When a primary or dialog window is resized, its contents remain visible so users can view the effect of the change in size on the amount of information that can be seen.

4.4.2 Window Location[4]

When a window opens, it is positioned on the screen so it is fully visible. If the window is placed relative to other objects on the screen, its position is adjusted as needed to ensure that the window is fully visible. If users can open multiple instances of a window, each successive window should be offset to the right and below the previous one; the windows should not all open at the same location (i.e., at the same screen coordinates). A window is positioned correctly when opened on displays that vary in size or resolution, with window location adjusted as needed so the title bar is visible.

The following guidelines define the initial location of a dialog window relative to its parent window when window placement is overlapping:

- If a dialog window can be stacked below its parent window (e.g., when the parent receives focus), the dialog should be placed so it is not completely obscured by the parent.

- If users do not need to see an object or information in the parent window, the dialog window can be centered on the screen or over the window to which it relates. In both cases, the window should be placed close to but not obscuring the title bar or menu bar in the parent window. If users need to see an object or information in the parent window, the dialog window should be placed so it does not obscure the object or information about it (e.g., its label).

If a dialog window is obscured by other windows, re-choosing the command that opened the window raises it to the foreground without affecting its position on the screen.

When a message window opens, it is placed at a consistent location that will attract user attention. Options include displaying the window at the center of the screen, at the current pointer position, or near the object where the action occurred that caused the message to open.

[4] These specifications apply to a workstation configuration with a single monitor.

Downloaded from http://www.everyspec.com

Users can move a window anywhere on the screen. However, they cannot move a window to a location from which they cannot reposition it (e.g., place the window so its title bar is completely off the screen).

Downloaded from http://www.everyspec.com

5. Menus

5.1 Pull-Down Menus

5.1.1 Appearance

A pull-down menu, shown in figure 5-1, consists of a title and a set of options from which users choose. The menu is accessed from a menu bar located at the top of a window.

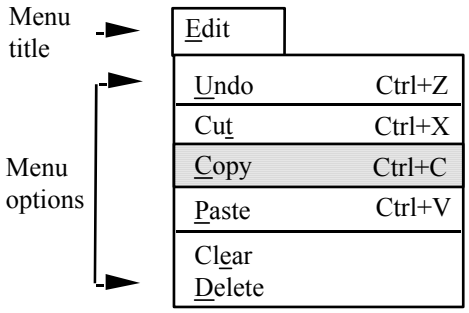


Figure 5-1. Example pull-down menu.

The title of a pull-down menu describes the category or type of options presented in the menu. The title is different from the other menu titles in the menu bar and does not appear as an option in the menu itself.

A pull-down menu includes options that perform actions, display a window (to request more information from users) or submenu, or specify a setting. A pull-down menu contains at least two options. Additional specifications on menu design are provided in section 5.4.

> Motif: If a pull-down menu can be torn off, it has a tear-off button with a dashed-line or other graphic that is the first option in the menu. When users activate the tear-off option, the contents of the menu are displayed in a menu window. The options in the window perform the same functions, are in the same order, and have the same availability as the options in the menu.
>
> MS Windows: If a pull-down menu is expandable, only the most frequently selected options are visible when the menu opens. Users can select one of these options in the shortened menu or view the hidden options by placing the pointer on a down-pointing arrow displayed at the bottom of the menu. After the menu is open for a few seconds, the hidden options are automatically displayed, and users can select an option from the full menu.

Downloaded from http://www.everyspec.com

5.1.2 Behavior

Pressing BLeft on a menu title displays a pull-down menu; releasing BLeft on a menu option activates it and dismisses the menu. In addition, clicking BLeft on a menu title displays the menu; clicking BLeft on a menu option activates it and dismisses the menu. Releasing BLeft off the menu, clicking BLeft off the menu, or clicking BLeft on the menu title a second time dismisses the menu without activating an option.

> GNOME, Java, Motif: Pressing BRight on a menu title also displays a pull-down menu; releasing BRight on a menu option activates it and dismisses the menu. In addition, clicking BRight on a menu title displays the menu; clicking BRight on a menu option activates it and dismisses the menu. Releasing BRight off the menu, clicking BRight off the menu, or clicking BRight on the menu title a second time dismisses the menu without activating an option.

Table 5-1 lists keyboard interaction methods for moving focus to a menu bar, displaying a pull-down menu, and activating a menu option.

Downloaded from http://www.everyspec.com

Table 5-1. Keyboard interaction methods for pull-down menus.

	GNOME	Java	Motif	MS Windows
F10 moves focus to a menu bar and displays the first menu.	X	X		
F10 moves focus to a menu bar.			X	X
ALT moves focus to a menu bar.				X
When a menu bar has focus, LEFT and RIGHT navigate between menu titles and display the associated menu, with wrapping from the last to the first title.	X	X		
When a menu bar has focus, LEFT and RIGHT navigate between menu titles, with wrapping from the last to the first title.			X	X
When a pull-down menu title has focus, DOWN and RETURN display the menu.			X	X
When a pull-down menu title has focus, SPACE displays the menu.			X	
When a pull-down menu title has focus, UP displays the menu.				X
When a pull-down menu is displayed, UP and DOWN navigate between menu options, with wrapping from the bottom to the top option.	X	X	X	X
When a pull-down menu is displayed, RETURN activates a menu option and dismisses the menu.	X	X	X	X
When a pull-down menu is displayed, SPACE activates a menu option and dismisses the menu.	X	X	X	
When a pull-down menu is displayed, ESC dismisses the menu without activating a menu option and removes focus from the menu bar.	X	X	X	
When a pull-down menu is displayed, ALT dismisses the menu without activating a menu option and removes focus from the menu bar.				X
When a pull-down menu is displayed, ESC dismisses the menu without activating a menu option; ESC a second time removes focus from the menu bar.				X

Downloaded from http://www.everyspec.com

5.2 Pop-Up Menus

5.2.1 Appearance

A pop-up menu, shown in figure 5-2, consists of a set of options that provide redundant access to frequently executed actions in a window. A pop-up menu is hidden from view until users choose to display it.

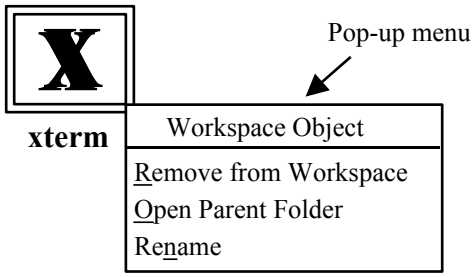


Figure 5-2. Example pop-up menu in Motif.

Motif: If a pop-up menu includes a title, it describes the function performed by the menu or the object to which the menu relates. The title is centered at the top of the menu and separated from the first option by a separator. The title does not include a mnemonic.

Motif: If a pop-up menu can be torn off, it has a tear-off button with a dashed-line or other graphic that is the first option in the menu. When users activate the tear-off option, the contents of the menu are displayed in a menu window. The options in the window perform the same functions, are in the same order, and have the same availability as the options in the menu.

A pop-up menu contains options that apply to the object under the pointer and its current context. A pop-up menu contains at least two options. Additional specifications on menu design are provided in section 5.4.

When the pointing device is used to display a pop-up menu, the left corner of the menu is positioned under the hotspot of the pointer.

GNOME, Motif: When the keyboard is used to display a pop-up menu, the menu is positioned at the pointer location.

Java, MS Windows: When the keyboard is used to display a pop-up menu, the menu is positioned next to the object.

Downloaded from http://www.everyspec.com

5.2.2 Behavior

GNOME, Java, Motif: Pressing BRight on an object displays a pop-up menu; releasing BRight on a menu option activates it and dismisses the menu.

Clicking BRight on an object displays a pop-up menu; clicking BLeft or BRight on a menu option activates it and dismisses the menu. Clicking BLeft off the menu dismisses it without activating an option.

GNOME, Java, Motif: Clicking BRight off a pop-up menu also dismisses it without activating an option.

GNOME, Motif: Releasing BRight off a pop-up menu also dismisses it without activating an option.

SHIFT+F10 displays a pop-up menu. When a pop-up menu is displayed, UP and DOWN navigate between menu options, with wrapping from the bottom to the top option. RETURN activates an option and dismisses the menu.

GNOME, Java, Motif: SPACE also activates an option and dismisses the menu.

When a pop-up menu is displayed, ESC dismisses the menu without activating an option.

Downloaded from http://www.everyspec.com

5.3 Submenus

5.3.1 Appearance

A submenu, shown in figure 5-3, presents groups of related menu options hierarchically. A submenu is accessed from a parent option in a pull-down or pop-up menu.

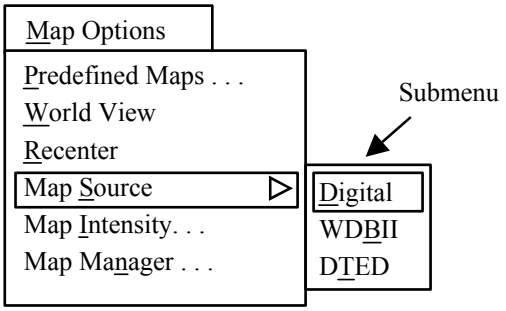


Figure 5-3. Example submenu.

Motif: If a submenu can be torn off, it has a tear-off button with a dashed-line or other graphic that is the first option in the submenu. When users activate the tear-off option, the contents of the submenu are displayed in a menu window. The options in the window perform the same functions, are in the same order, and have the same availability as the options in the submenu.

A submenu contains options that modify or are related to the parent menu option. A submenu contains at least three options; if there are only two options, they are placed in the parent menu rather than in a submenu. Additional specifications on submenu design are provided in section 5.4.

When a submenu is displayed, it is positioned to the right of the parent menu or above, below, or to the left if space to the right is limited. The first option in the submenu is aligned with the right-pointing arrow in the parent option for the submenu. A submenu does not repeat the parent option as the first option in the submenu.

5.3.2 Behavior

GNOME, Java, MS Windows: Placing the pointer on a parent option displays the submenu; clicking BLeft or BRight on a submenu option activates it and dismisses the submenu and parent menu. Moving the pointer to a different menu option or clicking BLeft or BRight off a submenu dismisses the submenu and parent menu without activating an option.

Motif: Pressing BLeft on a parent option displays the submenu; releasing BLeft on a submenu option activates it and dismisses the submenu and parent menu. Clicking BLeft on a parent option displays the submenu; clicking BLeft on a

Downloaded from http://www.everyspec.com

submenu option activates it and dismisses the submenu and parent menu. Releasing BLeft off a submenu or clicking BLeft off a submenu dismisses the submenu and parent menu without activating an option. Pressing BRight on a parent option also displays the submenu; releasing BRight on a submenu option activates it and dismisses the submenu and parent menu. In addition, clicking BRight on a parent option displays the submenu; clicking BRight on a submenu option activates it and dismisses the submenu and parent menu. Releasing BRight off a submenu or clicking BRight off a submenu dismisses the submenu and parent menu without activating an option.

GNOME: Moving focus to a parent option displays the submenu; RIGHT navigates to the first option in the submenu.

Java, Motif, MS Windows: When a parent option has focus, RIGHT and RETURN display the submenu, with focus on the first option in the submenu.

Java, Motif: When a parent option has focus, SPACE also displays the submenu, with focus on the first option.

When the submenu is displayed, UP and DOWN navigate between menu options, with wrapping from the bottom to the top option. RETURN activates an option and dismisses the parent menu and submenu.

GNOME, Java, Motif: When a submenu is displayed, SPACE also activates an option and dismisses the parent menu and submenu.

GNOME: When a submenu is displayed, ESC dismisses the submenu and parent menu without activating an option.

Java, Motif, MS Windows: When a submenu is displayed, LEFT and ESC dismiss the submenu without activating an option and return focus to the parent option.

Downloaded from http://www.everyspec.com

5.4 Menu Design[1]

5.4.1 Types of Menu Options

Menu options can be actions that are executed immediately, routings that display a window or submenu, or settings that define parameters or specify a state. Figure 5-4 illustrates these types of options.

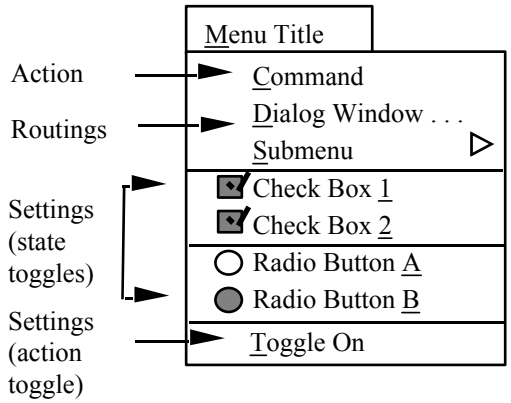


Figure 5-4. Examples of menu option types.

A routing option that requires additional information from the user before it can be completed is followed by an ellipsis (i.e., ". . ."); there are no spaces between the label and ellipsis or between the periods in the ellipsis. When a routing option causes a dialog window to be displayed, the option should remain available (i.e., not be dimmed) while the window is open. A routing option that displays a submenu is followed by a right-pointing arrow positioned at the right edge of the menu.

An option that is a setting can be an action toggle (e.g., turn on/turn off) or a state toggle (e.g., select a font size). An action toggle is worded to describe the action that is implemented when the option is activated. A state toggle can be an exclusive or nonexclusive setting. A radio button menu option is used to represent an exclusive setting, and a check box menu option a nonexclusive setting. An action toggle is preferred when space in a menu is limited. However, state toggles are easier to understand, given that users do not have to consider whether the option indicates the current state of a setting or the state the setting changes to when the option is selected. If space within the menu is limited, a check box menu option should be used for the state toggle; otherwise, a radio button menu option is preferred because it identifies each of the states that can be chosen.

Both radio button and check box menu options are preceded by a graphic when the setting is applied. When a radio button menu option is selected, a dot is displayed to the

[1] The specifications presented in this section apply to the design of pull-down menus, pop-up menus, and submenus. When the term "menu" is used, it applies to all three types of menus, unless otherwise indicated.

Downloaded from http://www.everyspec.com

left of the option or its indicator is filled; when the menu option is not selected, the dot is removed or the indicator is empty. When a check box menu option is selected, a check mark is displayed to the left of the option or its indicator is filled; when the menu option is not selected, the check mark is removed or the indicator is empty. If the state of the setting is indeterminate (i.e., has a mixed-value state), the graphic for the option is empty, shaded, or removed.

5.4.2 Available and Unavailable Options

A menu option that is never available to users (e.g., system administrator commands) is not included in a menu. Similarly, an option that does not apply to the currently selected object is not included in a pop-up menu. An option that is temporarily unavailable is included in the menu but dimmed (as in figure 5-5) to indicate that it cannot be selected. While the wording of options may change (e.g., when an option is an action toggle), options are not added to or deleted from a menu to indicate their availability.[2]

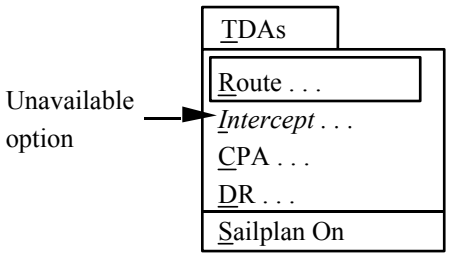


Figure 5-5. Example of an unavailable menu option.

A menu title is shown as available and the associated menu can be displayed even if all of the options in the menu are unavailable for selection. Similarly, a parent option for a submenu is shown as available and the submenu can be displayed even if all of the options in the submenu are unavailable.

5.4.3 Format and Wording

The options in a menu are left-aligned based on their text labels, with sufficient space preceding the options to display any graphics (e.g., toolbar icons, dot or check indicators) included in the menu (see figure 5-4). An option includes an icon only if the icon is used elsewhere in an application (e.g., in a toolbar) to represent the action performed by the option. A menu is wide enough to display the longest option and a graphic and shortcut key (if they are available) on a single line. If shortcut keys are included in a menu, they are aligned in a column to the right of text for an option. Long menu options are accommodated by making the menu wider or by replacing portions of the text with an ellipsis.

[2] The inclusion of unavailable options in a menu is consistent with Motif and MS Windows but not with MIL-STD 1472F.

Downloaded from http://www.everyspec.com

Java, MS Windows: Some objects (e.g., an application icon) are assigned a default action that is executed when users double click on the object. If the action is included in the pop-up menu for the object, the action is the first option in the menu and is displayed in bold font. A default action may also be assigned to an operation such as drag transfer. In this case, if the action is designated as the default in a pop-up menu, it is displayed in bold font and is not a destructive action (e.g., Exit, Delete). For example, if the default transfer operation for an object is to copy the object at a drop site, then the default option in the pop-up menu for the object is Copy.

An application uses the vocabulary in appendix A when the actions listed are included in menu options. If application-specific menu options are created, the following guidelines apply:

- A menu option should be a verb, noun, or verb phrase that describes an action and/or object on which the action is performed. For example, an Edit menu might include actions such as Cut, Copy, and Paste, nouns such as Links and Objects, and verb phrases such as Copy as HTML and Paste Special.

- If a pull-down menu performs the same action on different objects, the menu title should describe the action (e.g., Insert), with individual menu options identifying the objects to which the action can be applied (e.g., Page Break, Footnote, File, Picture). If the menu performs different actions on the same object, the menu title should be the name of the object (e.g., Table), with the menu options identifying the actions (e.g., Insert, Delete). The name of the object should not be repeated in each option; for example, the options in a File menu are New, Open, and Save, not New File, Open File, and Save File.

- A menu option should be worded as a command to the application (rather than as a question to the user), in the vocabulary of the user.

- If desired, tooltip help (see section 9.1.2) can be provided for a menu option.

The following guidelines apply to menu options that are action or state toggles:

- The wording of an action toggle should be semantically congruent with natural usage; for example, if one action is worded “Move Object Up,” the other action is “Move Object Down,” not “Move Object Back.”

- When users activate an action toggle (e.g., Turn On), the wording of the menu option should change (e.g., Turn Off) to reflect the action that will be implemented when the action is executed again.

- Only one of the action toggles should appear in a menu at any time.

Downloaded from http://www.everyspec.com

- If a multi-level Undo capability (see section 3.5.6) is provided, the wording of the option should change dynamically to reflect the action that can be undone. For example, if the most recently executed option is Cut, the option should be worded "Undo Cut." If the action cannot be undone or redone, the option should be worded "Can't Undo" or "Can't Redo" and should be shown as unavailable for selection.

- Options that are state toggles should be worded to describe the state being set (e.g., a series of font names).

- When users activate a state toggle, its select state should change but the wording of the option remains the same.

5.4.4 Grouping into Menus and Submenus

Menu options that are related are grouped together and delimited from other options by a separator. The following guidelines apply to grouping menu options:

- The options should be organized into groups based on function. If the options cannot be organized by function, they should be ordered by frequency of usage, with the most frequently executed at the top of the menu and least frequently executed at the bottom. If an organization based on function or frequency of use is inappropriate, then the options should be ordered alphabetically or in numerical order.

- Menu options that perform opposing actions (e.g., Save and Delete) should not be placed adjacent to each other in order to reduce the likelihood of users inadvertently selecting an incorrect action.

- If the same options are included in different menus, the options should be ordered in a consistent manner in each menu.

- If a group of radio button or check box menu options is included in a menu, the group should be delimited by separators to distinguish it from other options in the menu.

If submenus are included in a menu, they are limited to one level below the parent menu and organized as shown in figure 5-6; if the number of submenus will exceed two levels, new menus are created or a dialog window is used. The following guidelines apply to grouping submenu options:

- If an option is executed frequently, it should be placed in the parent menu rather than in a submenu.

- If a submenu contains a set of actions, the label for the parent option should be the object on which the actions can be performed.

Downloaded from http://www.everyspec.com

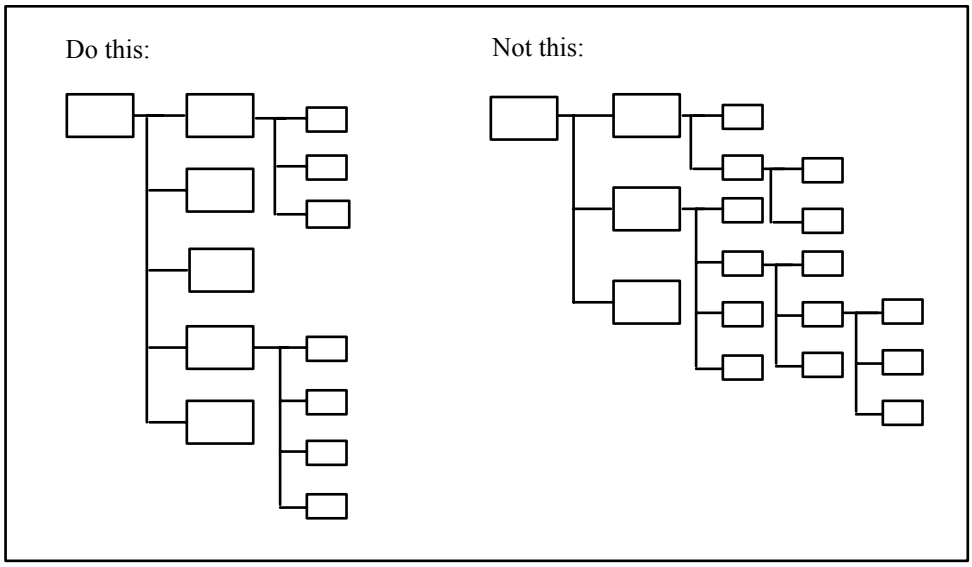


Figure 5-6. Organization of options in submenus.

5.4.5 Mnemonics and Shortcut Keys

An application uses the mnemonics in appendix A when the actions listed are included in a menu. Mnemonics are assigned to the titles of pull-down menus and to the options in pull-down and pop-up menus. Each pull-down menu title is assigned a unique mnemonic, and each option in a pull-down or pop-up menu is assigned a unique mnemonic. The following guidelines apply to assigning mnemonics in menus:

- A menu option should have the same mnemonic whenever it appears in a menu. For example, the mnemonic assigned to an option in a pop-up menu should be the same mnemonic assigned to the option in a pull-down menu.

- Mnemonics should be selected to produce minimal interference with each other. For example, the same character should not be assigned as the mnemonic for options performing opposite or contradictory actions in different menus (e.g., C is not used as the mnemonic for Continue in one menu and Close in another).

- If a menu option changes based on the state of an object (e.g., Show Details, Hide Details), the same mnemonic should be used in both versions of the option so users do not have to know the state of the object before they type the mnemonic.

An application uses the shortcut keys in appendix A when the actions listed are included in a menu. Shortcut keys are assigned to the frequently executed options in a pull-down menu (e.g., actions that may also be available as toolbar buttons). Shortcut keys are used to provide redundant access to actions that are available as menu options and are not the sole method for accessing any actions in an application. If shortcut keys are included in a

Downloaded from http://www.everyspec.com

pop-up menu, they are the same as the ones in the corresponding pull-down menu. The following guidelines apply to assigning shortcut keys in menus:

- A menu option should have the same shortcut key whenever it appears in a menu.
- The characters used in the mnemonic and shortcut key for a menu option should be selected to be coordinate with each other. For example, if S is the mnemonic for Save, Ctrl+S is used as the shortcut key.

Downloaded from http://www.everyspec.com

This page intentionally left blank.

Downloaded from http://www.everyspec.com

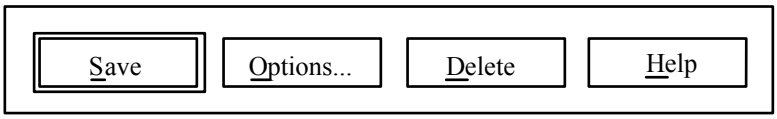
6. Controls

6.1 Push Buttons

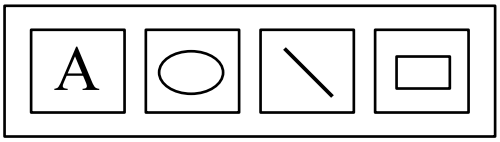
6.1.1 Appearance

A push button is used to initiate an action. The button has a text and/or graphic label (as shown in figure 6-1) that identifies the action executed when the button is activated. The label is centered in the button. If a push button contains both a text and graphic label, the graphic is placed either above the text or to the left of it. The label is grayed out if a push button is unavailable. The push button designated as the default has a dark border or an extra outline around it. Sections 8.2.2.2 and 8.3.2.2 address the placement of push buttons in dialog and message windows, respectively.

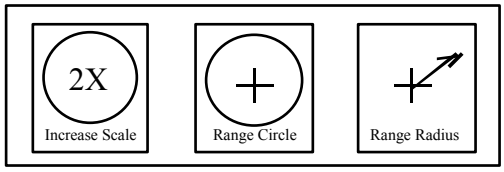
NOTE: In GNOME, some themes do not provide a visual cue that identifies the default push button. The desired implementation is as described above.



Push Buttons with Text Labels



Push Buttons with Graphic Labels



Push Buttons with Graphic and Text Labels

Figure 6-1. Example push buttons.

A push button with text label includes an ellipsis if additional information is needed from users before the action can be executed. A push button with a graphic label does not contain an ellipsis, even if a window opens when the push button is selected.

Motif, MS Windows: A push button with a text label includes “>>” if its action expands the window to display additional information.

Downloaded from http://www.everyspec.com

Java, Motif, MS Windows: A push button includes a menu indicator (i.e., down-pointing arrow) if its action is to display a menu.

An application uses the vocabulary in appendix A when the actions listed are included in push buttons. If application-specific vocabulary is created, the following guidelines apply:

- A push button label should be a verb and stated in active voice; the label should describe the results of activating the button and identify the action that will be taken by the application rather than the user. The label should not include extraneous words (e.g., "Now" in "Find Now").

- The names of push button actions should be congruent (e.g., Save/Delete, On/Off, In/Out), with the same vocabulary used to describe an action throughout an application.

- Terms such as "All" should be used in labels (e.g., Select All, Delete All) only when there is no ambiguity as to the objects to which All refers; if a label can have multiple referents, the name of the object should be used instead of All.

- If the action performed by a push button is to open a dialog window, the button label should be match the title of the window, followed by an ellipsis (e.g., Defaults...).

An application uses standard graphics provided by the interface when the actions represented by the graphics are included in push buttons. If application-specific graphics are created, the following guidelines apply:

- A push button graphic should be designed so users can easily identify the action performed when the button is selected.

- A push button graphic should be unambiguous and easily distinguished from the other graphics with which it is displayed.

- Push button graphics that represent opposite functions (e.g., Save, Delete) should be designed to mirror each other.

- Push button graphics should provide a symbolic, rather than a photo-realistic, representation of an action. For example, the graphics can depict a before-and-after representation of the action (e.g., a small and large version of an object, connected by an arrow, to indicate Magnify), the tool that would accomplish the action (e.g., a pair of paperclips to indicate Attach), or the action itself (e.g., a paintbrush filling an object with color to indicate Paint).

Downloaded from http://www.everyspec.com

6.1.2 Behavior

Clicking BLeft on a push button activates it. Releasing BLeft outside the push button does not activate it.

> Java, Motif, MS Windows: If the push button action is to display a menu, the methods for activating a menu option are the same as those used in a pull-down menu (see section 5.1.2).

When a push button has focus, SPACE activates it. If a push button is designated as the default, RETURN activates it even if it does not have focus; if a control such as a multi-line text area has focus, CTRL+RETURN activates the default action. If there is no default push button, RETURN has no effect.

Downloaded from http://www.everyspec.com

6.2 Radio Buttons

6.2.1 Appearance

Radio buttons are used to make a one-of-many selection. Radio buttons are appropriate when the states being set are related but not necessarily opposites (e.g., Full Duplex and Half Duplex).

A radio button has a circle-shaped indicator and a label describing the state being set, as shown in figure 6-2. The label is placed to the right of the indicator and has no ending punctuation (e.g., a period, question mark). The label is usually text but may be graphic. The label is worded as a phrase (not a full sentence), with parallel construction applied to all of the radio buttons in a group. The label is grayed out if a radio button is unavailable. If a radio button parents another control (e.g., a conditional text box), the radio button is placed above or to the left of the control, and the radio button label is followed by a colon.

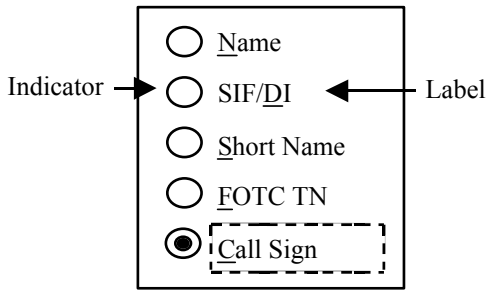


Figure 6-2. Example radio button group.

The indicator for a radio button is filled (e.g., contains a dot) if the button is selected and is empty if the button is deselected.

> GNOME: If the state of a radio button group is indeterminate (i.e., has a mixed-value state), all of the relevant indicators have a shaded pattern.
>
> Java, Motif, MS Windows: If the state of a radio button group is indeterminate (i.e., has a mixed-value state), all of the indicators are empty.

A group of radio buttons contains at least two radio buttons. The radio buttons in the group represent all possible states that can be set; if necessary, a radio button labeled "None" can be included in the group to cover any remaining states. The radio buttons are arranged in one or more rows and columns and left-aligned. If radio buttons are arranged horizontally, space should be sufficient (at least twice the distance between the indicator and its label) so the indicator is paired with the label on the right, not the left.

Downloaded from http://www.everyspec.com

6.2.2 Behavior

Clicking BLeft on a radio button selects it. When the state of a group of radio buttons is indeterminate, clicking BLeft on a button selects it and deselects all of the other buttons in the group.

When a group of radio buttons has focus, the arrow keys navigate between the buttons, with wrapping from the bottom to the top button.

GNOME, Java, MS Windows: Moving focus to a radio button selects it.

Motif: When a radio button has focus, SPACE selects it.

Downloaded from http://www.everyspec.com

6.3 Check Boxes

6.3.1 Appearance

Check boxes are used to make a many-of-many selection. A check box is appropriate when the states being set are opposites (e.g., "Disable auditing" is assumed to be the opposite of "Enable auditing"); otherwise, two radio buttons should be used.

A check box has a square-shaped indicator and a label describing the state being set, as shown in figure 6-3. The label is placed to the right of the indicator and has no ending punctuation (e.g., a period, question mark). The label is usually text but may be graphic. The label is worded as a phrase (not a full sentence), with parallel construction applied to all of the check boxes in a group. The label is phrased as a positive statement about the state being set (e.g., "Enable auditing," rather than "Do not disable auditing"). The label is grayed out if a check box is unavailable. If a check box parents another control (e.g., a conditional text box), the check box is placed above or to the left of the control, and the check box label is followed by a colon.

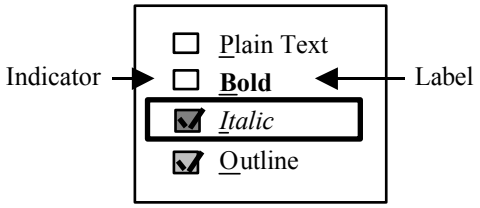


Figure 6-3. Example check box group.

The indicator for a check box is filled (e.g., contains a check mark or an X) if the button is selected and is empty if the button is deselected. If the state of a check box is indeterminate (i.e., has a mixed state), the indicator has a shaded pattern.

Check boxes can be presented individually or in a group. The check boxes in a group are arranged in one or more rows or columns and left-aligned. If the check boxes are arranged horizontally, space should be sufficient (at least twice the distance between the indicator and its label) so the indicator is paired with the label on the right, not the left.

6.3.2 Behavior

Clicking BLeft on a check box selects it. When the state of a check box is indeterminate, clicking BLeft on it changes its state to selected, clicking on it a second time changes its state from selected to unselected, and clicking on it a third time changes its state from unselected to indeterminate.

When a check box has focus, SPACE selects it.

Downloaded from http://www.everyspec.com

6.4 Toolbar Buttons

6.4.1 Appearance

A toolbar button is used to initiate an action or apply a state. Toolbar buttons, shown in figure 6-4, are displayed in groups in the toolbar of a window and provide redundant access to functions available elsewhere in the window (e.g., in a pull-down menu). Specifications on toolbar design are provided in section 8.1.1.2.

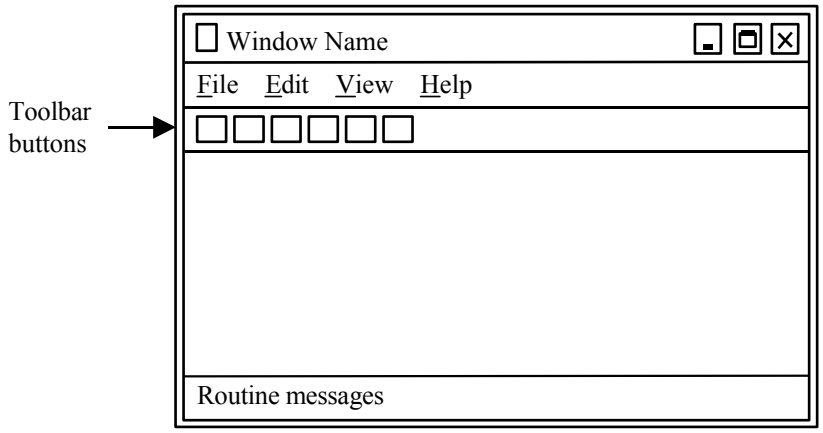


Figure 6-4. Example toolbar buttons.

A toolbar button has a graphic label identifying its action or state. The label for a toolbar button is centered in the button. If a toolbar button contains both text and a graphic, the graphic is placed either above the text or to the left of it. The text is a single word and does not include a mnemonic. The label is grayed out if a toolbar button is unavailable. A toolbar button does not contain an ellipsis, even if a window opens when the button is selected. A toolbar button can have a normal border (similar to that of a push button), or the border can be invisible until the pointer moves over the button (i.e., provides mouse-over feedback).

Java: A toolbar button can include an indicator or badge that provides additional information about the action performed by the button. The button has a menu indicator (i.e., a down-pointing arrow) if it parents a menu containing additional choices related to the button. A toolbar button has a new object indicator (i.e., an asterisk) if it creates a new object. The button has an add object indicator (i.e., a plus sign) if it adds an object to a group. The button has a properties indicator (i.e., a check mark) if it opens a Properties window for the object represented in the button.

GNOME, MS Windows: A toolbar button can be split to both support a default action and provide access to a menu of other actions. The left portion of the button has a label that describes the default action, and the right portion of the button contains a menu indicator button (i.e., a down-pointing arrow) that

Downloaded from http://www.everyspec.com

displays the menu. For example, the Back button in a browser might navigate to the previous Web page as its default action and open a menu of recently viewed Web pages that users can choose.

An application uses the vocabulary in appendix A when the actions listed are included in toolbar buttons. An application uses standard graphics provided by the interface when the actions represented by the graphics are included in toolbar buttons. If application-specific vocabulary and/or graphics are created, they follow the guidelines for push buttons in section 6.1.1.

Toolbar buttons that apply a state can control a set of related properties, with the same behavior as a set of exclusive or nonexclusive settings. Toolbar buttons can be used in groups to make a one-of-many selection (e.g., Align Left, Align Center, Align Right) or in groups or individually to make a many-of-many selection (e.g., Bold, Underline, Italic). In both cases, when a toolbar button is selected, the setting is applied immediately; the background of the button changes (to indicate select state) but the button label remains unchanged.

Toolbar buttons that apply a state can also be used to invoke a mode (e.g., in a set of drawing tools). The toolbar button remains selected as long as the mode is in effect, and the pointer shape changes to indicate the type of operation performed in the mode. Either the button automatically returns to an unselected state after an operation is executed, or a second toolbar button is provided for returning to an unselected state (e.g., exiting the mode).

6.4.2 Behavior

Clicking BLeft on a toolbar button activates or selects it. If a toolbar button performs a many-of-many selection, clicking BLeft on a selected button deselects it. If a toolbar button parents a menu, pressing or clicking BLeft on the button displays the menu.

GNOME, Java, Motif: If a toolbar has focus, LEFT and RIGHT navigate between the buttons in the toolbar. If a toolbar button has focus, SPACE activates or selects it.

GNOME: If a toolbar button has focus, RETURN also activates or selects it.

MS Windows: A toolbar does not support keyboard interaction. However, users can access the action performed by a toolbar button from the keyboard by activating the control (e.g., menu option) for which it provides redundant access.

Downloaded from http://www.everyspec.com

6.5 List Boxes

6.5.1 Appearance

A list box, shown in figure 6-5, is used to present a set of items from which to select. A list box can display one or more columns of information for each item in the list and can support selection of one or more items in the list. A label is included with the list box that describes its contents.

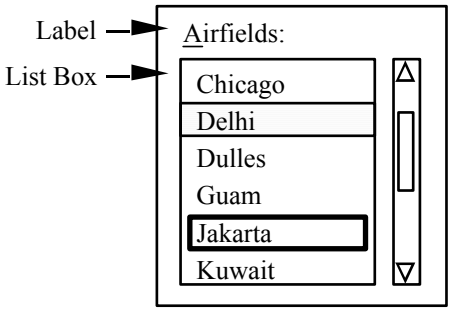


Figure 6-5. Example list box.

A list box includes vertical and/or horizontal scroll bars when its contents exceed the space available. The vertical scroll bar should remain displayed to the right of the list but be disabled when the content of the list box no longer requires scrolling. Whenever possible, a list box should be wide enough to view its contents without scrolling horizontally. If space is limited or the items in the list box vary in length, the items can be truncated and followed by an ellipsis, or a horizontal scroll bar can be used.

The normal size of a list box is large enough to display up to eight items at a time without scrolling, or all of the items if there are fewer than six. The list box may be larger if the number of items is very large. If the window containing the list box can be resized, the list should increase in size as the window is resized larger and decrease in size as the window is resized smaller.

The items in a list box are left-aligned and are presented in a logical order (e.g., alphabetical) based on the nature of the items. For example, a list of port names is ordered alphabetically, and a list of messages by precedence, date-time group, or a combination of the two (e.g., date-time group within precedence).

If an item in a list box is temporarily unavailable, it is omitted from the list (rather than grayed out). Selecting an item in a list box does not affect item order. When an item is added to a list, it appears in its correct position (e.g., in alphabetical order) rather than at the end of the list. If the content of a list box is updated through an automatic process, the list does not scroll automatically to the item that was added; the list box scrolls only in response to a user action (e.g., clicking on a scroll arrow).

Downloaded from http://www.everyspec.com

Multiple-column list boxes. If a list box has multiple columns, each column has a heading with a label that describes the content of the column, as shown in figure 6-6. The heading includes the unit of measurement (e.g. knots, miles) where appropriate and is not followed by a colon. The headings are placed outside the list box so they remain visible when the list is scrolled.

GNOME, Java, MS Windows: If a column is too narrow to display the entire heading, the text is truncated and may be followed by an ellipsis (to indicate the presence of additional text) or provide a tooltip showing the full heading.

Name	Size	Type	Modified
Test1	50KB	Text Document	5/26/02 8:15 AM
Test2	100KB	Text Document	4/23/01 2:30 PM
Test3	250KB	Text Document	10/2/01 10:48 AM
Test4	5KB	Text Document	7/15/00 1:04 PM

Figure 6-6. Example multiple-column list box.

The information in each column of the list box is aligned based on the nature of the content. Text is left-aligned, graphics are centered, numeric values right-aligned, and decimal values aligned on the decimal point. The list box should include blank space between the content of a column and the left and right edges of the column and uses grid lines and row striping as appropriate to improve readability. The items are presented in a logical order, with the default based on the content of the first column of the list. If desired, a list box can support sorting based on the content of other columns, with ascending order being the default. A list box that supports sorting in both ascending and descending order includes a sort indicator (e.g., an up- or down-pointing arrow graphic) in the column heading to identify the sort column and indicate the direction of the sort. The sort indicator is displayed as long as the sort order is valid.

Multiple-selection list boxes. If desired, a list box that supports range and/or disjoint selection can include a text label either above or below the list indicating the number of items currently selected (e.g., “4 Selected”).

GNOME, MS Windows: A multiple-selection list box includes a check box indicator preceding each item (as a visual cue that multiple items can be selected).

Downloaded from http://www.everyspec.com

6.5.2 Behavior

Clicking BLeft on an item in a list box selects the item. If a column in a list box can be resized, dragging its edge using BLeft resizes the column in the direction of the drag. If a column in a list box can be sorted, clicking BLeft on its heading sorts the items in the list in ascending or descending order. If a list box supports range and/or disjoint selection, they are performed as defined in table 3-1.

When a list box has focus, UP and DOWN navigate between items, with no wrapping from the bottom to the top item. If the list supports other keyboard navigation actions, the key combinations used are the standard ones supported by the interface. The headings in a multiple-column list box do not support keyboard access. However, if desired, an application can attach a pop-up menu to the headings that contain options for sorting column contents.

Moving focus to a list item selects it, or SPACE selects the item with focus. If range and/or disjoint selection are available in the list, they are performed as defined in table 3-2.

> Java, MS Windows: If speed search is supported in a list box, typing a character scrolls the list to the first item that begins with the character and selects it. Typing the character a second time scrolls to the next item that starts with the character and selects it. When the search reaches the bottom of the list, it wraps to the top and continues. If the character does not match an item in the list, the currently selected item remains selected, and feedback (e.g., an auditory signal) may be provided to indicate that no match was found.

Downloaded from http://www.everyspec.com

6.6 Drop-Down List Boxes

6.6.1 Appearance

A drop-down list box, shown in figure 6-7, is used to select from a set of items that is displayed on demand. A label is included with a drop-down list box that describes its contents.

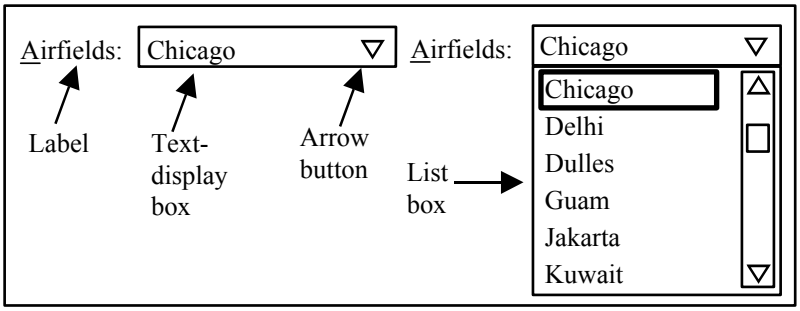


Figure 6-7. Example drop-down list box.

A drop-down list box consists of a text-display box and a down-pointing arrow button which, when activated, displays a list box. The text-display box shows the currently selected item. The text-display box and the list box are wide enough to display all of the items in the list.

The list box is sized to display up to eight items at a time but may be larger if the number of items is very large. The list box includes a vertical scroll bar when the number of items exceeds the space available. The items are presented in a logical order (e.g., alphabetical) based on the nature of the items. If an item is temporarily unavailable, it is omitted from the list box (rather than grayed out). If users need to select none of the items in a drop-down list box, an item labeled None should be included in the list box; this item is displayed in the text-display box if selected.

6.6.2 Behavior

Pressing BLeft on the text-display box or arrow button displays the list box; releasing BLeft on an item selects it, displays the item in the text-display box, and dismisses the list box. In addition, clicking BLeft on the text-display box or arrow button displays the list box; clicking BLeft on an item selects it, displays the item in the text-display box, and dismisses the list box. Releasing BLeft outside the list box, clicking on the text-display box or arrow button a second time, or clicking outside the drop-down list box dismisses the list box without selecting an item.

NOTE: In GNOME, this behavior is supported only for the arrow button. In addition, BMiddle and BRight also perform a selection in a

Downloaded from http://www.everyspec.com

drop-down list box. The desired implementation is as described above.

GNOME: When a drop-down list box has focus, DOWN cycles through items in the list box and displays them in the text-display box without opening the list box.

When a drop-down list box has focus, SPACE and ALT+DOWN display the list box.

GNOME: When a drop-down list box has focus, RETURN also displays the list box.

Java: When a drop-down list box has focus, DOWN and ALT+UP also display the list box.

Motif: When a drop-down list box has focus, DOWN also displays the list box.

MS Windows: When a drop-down list box has focus, DOWN, UP, ALT+UP, and F4 also display the list box.

When the list box is displayed, UP and DOWN navigate between items, with no wrapping from the bottom to the top item.

GNOME, MS Windows: When the list box is displayed, RETURN selects the highlighted item, displays the item in the text-display box, and dismisses the list box. ESC dismisses the list box without changing the selection in the text-display box.

Java, Motif: When the list box is displayed, navigating to a list item selects it and displays it in the text-display box; RETURN dismisses the list box. ESC dismisses the list box and maintains the latest selection in the text-display box.

Downloaded from http://www.everyspec.com

6.7 Option Menus (GNOME, Motif)

6.7.1 Appearance

An option menu, shown in figure 6-8, is used to select from a set of options that is displayed on demand. A label is included with an option menu that describes its contents.

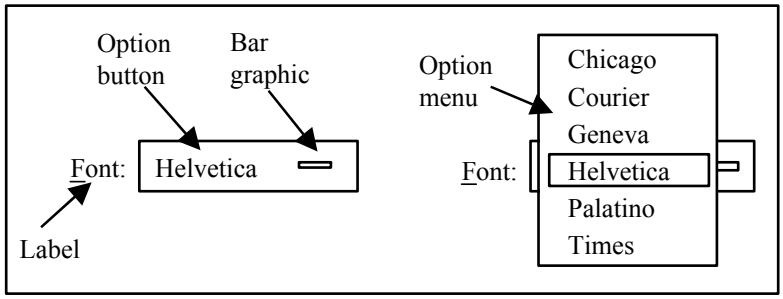


Figure 6-8. Example option menu.

An option menu consists of an option button which, when activated, displays an option menu. The option menu button shows the menu option that is currently selected and includes an option menu graphic (e.g., a horizontal bar). The button is wide enough to display the longest menu option and the graphic; the graphic does not obscure any text in the button.

The option menu contains no less than two or more than 12 options, all of which are visible when the menu opens. The option menu is positioned so it does not obscure the graphic in the option button. The options are presented in a logical order (e.g., alphabetical) based on the nature of the options. The menu does not include a separator or any options that are settings (i.e., action toggles) and does not have any submenus. If an option is temporarily unavailable, it is omitted from the menu (rather than grayed out).

6.7.2 Behavior

Pressing BLeft on an option button displays the menu; releasing BLeft on an option selects it, displays the option in the button, and dismisses the menu. In addition, clicking BLeft on an option button displays the menu; clicking BLeft on an option selects it, displays the option in the button, and dismisses the menu. Releasing BLeft off the menu or clicking BLeft off the menu dismisses the menu without selecting an option.

> Motif: Pressing BRight on an option button also displays an option menu; releasing BRight on an option selects it, displays the option in the button, and dismisses the menu. In addition, clicking BRight on an option button displays the menu; clicking BRight on an option selects it, displays the option in the button, and dismisses the menu. Releasing BRight off the menu or clicking BRight off the menu dismisses the menu without selecting an option.

Downloaded from http://www.everyspec.com

When the option button has focus, SPACE displays the menu; the currently selected option is highlighted. When an option menu is displayed, UP and DOWN navigate between menu options, with wrapping from the bottom to the top option. RETURN and SPACE select the highlighted option, display the option in the button, and dismiss the menu. When an option menu is displayed, ESC dismisses the menu without selecting an option.

Downloaded from http://www.everyspec.com

6.8 Tree Controls

6.8.1 Appearance

A tree control is used to display a hierarchical list of items (e.g., the folders and files in a file system). If an item in a tree control contains other items, it is preceded by a display control button, shown in figure 6-9, that toggles between showing and hiding the contents of the container. A container that is empty (e.g., folders 2 and 3 in the left panel of figure 6-9) does not have a display control button.

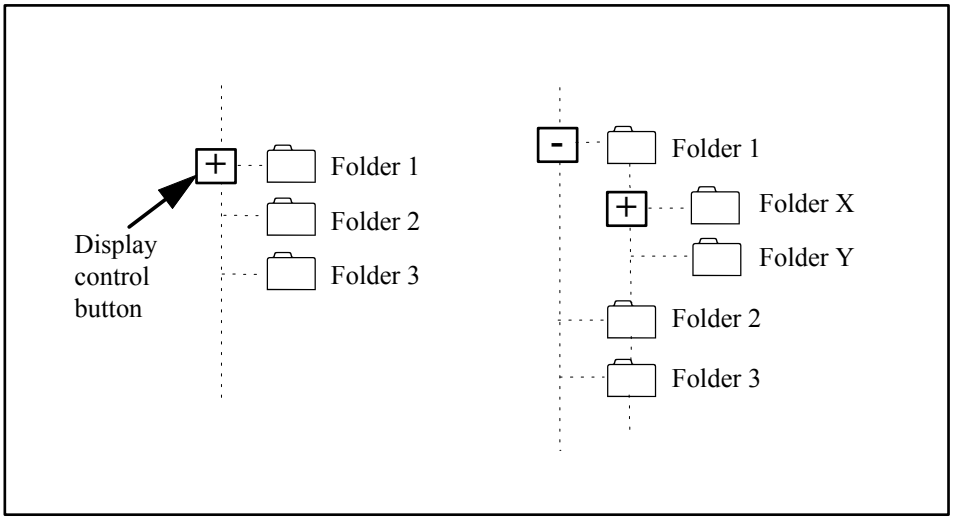


Figure 6-9. Example collapsed and expanded tree hierarchy.

A display control button contains a graphic that indicates when the contents of the container are hidden and when they are visible. When the contents are visible (e.g., folders X and Y in the right panel of figure 6-9), they are listed vertically below and indented to the right of the container. Vertical lines may be included with the display buttons to provide an additional indication of the hierarchical relationship of the items in the list. The items have a text label and may be preceded by a graphic (e.g., a small-size icon) to the left of the item. If other information is included with the item, it is presented in columns to the right of the label for the item.

6.8.2 Behavior

Clicking BLeft on the display control button for an item toggles between showing and hiding its contents.

> GNOME, Java, MS Windows: When a tree control has focus, UP and DOWN navigate between items, with no wrapping from the bottom item to the top item. Moving focus to an item selects it.

Downloaded from http://www.everyspec.com

Java, MS Windows: When a container has focus, RIGHT and LEFT toggle the display control button between showing and hiding the container contents.

Motif: When a tree control has focus, UP and DOWN navigate between items in the arrow direction, with wrapping from the bottom to the top item. When an item has focus, SPACE selects it.

Downloaded from http://www.everyspec.com

This page intentionally left blank.

Downloaded from http://www.everyspec.com

7. More Controls

7.1 Text Boxes

7.1.1 Appearance

A text box, shown in figure 7-1, is used to enter and edit text.[1] A label is included with a text box that describes its contents.

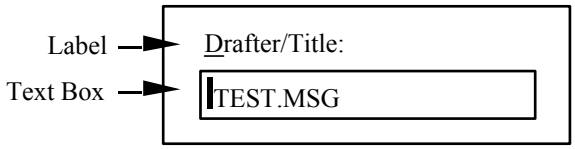


Figure 7-1. Example text box.

If the text being entered is a fixed length, the text box is the same length as the text string. If the text varies in length, the text box is as long as the longest text string so that the text does not scroll out of view when users type to the end of the text box. In cases where the text being entered is excessively long (e.g., descriptors or directory paths) and/or space within a window is extremely limited, the text box can be sized so that the text string is longer than the width of the box. In this case, text scrolling is enabled, with HOME and END scrolling to the beginning and end of the text string. The following guidelines apply to the design of text boxes:

- If entries such as date, time, latitude, and longitude are broken into chunks and entered in separate text boxes, the format used should be meaningful to users (e.g., year, month, and day in a date) and consistent with their expectations. The same formats should be used throughout an application, unless special or unique requirements dictate that multiple formats be supported.

- If desired, entries such as the ones above can be included in a single “smart” text box that traps user keystrokes and provides feedback to users that they are entering the correct characters. A “smart” text box may optionally provide a format template that guides users to enter the correct sequence of characters. For example, a text box for entering Social Security number may be displayed initially as “XXX-XX-XXXX;” as the user types, only numeric characters would be allowed, and the text cursor would automatically jump the hyphens.

- Text boxes can be used in combination with other controls to reduce the amount of text entry required by users; for example, drop-down lists can be used for selecting state names, and drop-down lists or radio buttons can be used for entering the time of day.

[1] The specifications presented here apply to a single-line text box unless otherwise indicated.

Downloaded from http://www.everyspec.com

- Routine or default data, data already known by an application, and data that can be computed by an application should be automatically entered whenever possible. For example, if text boxes are provided for the start date, end date, and duration of a mission, users should enter only two of the values and an application should calculate the third.

When a text box has focus, cues about the format of text to be entered are presented with the text box or displayed in the status bar (if one is present). If a unit of measurement (e.g., feet, miles) is always associated with a text box, it can be displayed as part of the label (i.e., preceding the text box) or placed to the right of the text box, as shown in figure 7-2. Cues about whether text entry is mandatory or optional are provided with the text box (e.g., in the label or group heading, color coded in the text box itself, or displayed in the status bar if the window has one). If the text being entered is limited to specific values, feedback (e.g., auditory signal) is provided when users enter an invalid value.

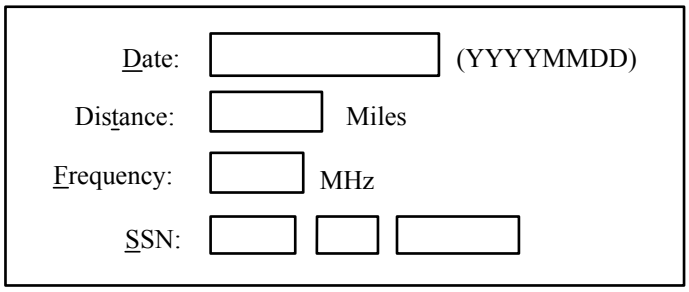


Figure 7-2. Example text box labels providing format cues.

Conditional text boxes. A conditional (or dependent) text box is placed to the right of or below the control to which it relates (e.g., a radio button or check box). The text box can be unavailable (with its label grayed out) or not displayed at all until the control to which it relates is selected.

Text-display boxes. An application is expected to display editable text in a text box and non-editable text in a label. If an application needs to display text that users can select and copy (e.g., to perform a transfer operation) but cannot change, it uses a text-display box to do so. A text-display box has a different appearance (e.g., flat border, a different outline or background color) than a text box to indicate that it is not editable.

7.1.2 Behavior

Text entry. Clicking BLeft on the text box moves focus to it and displays a text cursor in the box.

Downloaded from http://www.everyspec.com

GNOME, Java, MS Windows: When a text box receives focus, the text is highlighted, and the text cursor is placed at the end of the text. Typing in the text box replaces the highlighted text with the text being entered at the cursor location.

Motif: When a text box receives focus, the text cursor is placed at the same location where it was displayed when the text box lost focus. Typing in a text box inserts the text being entered at the text cursor location.

If desired, when users begin to type in a text box, the pointer can be removed from view so it does not obscure the text being entered; the pointer remains hidden until users move the pointer. When the window containing a text box loses focus, an application can either remove the text cursor from view or keep it visible but gray it out. If the text cursor is removed, it re-appears at the same location when the window containing the text box regains focus. If the cursor is grayed out, it stops flashing when the text box loses focus, and then returns to normal appearance and resumes flashing when the text box regains focus.

Text entry is possible only after the text cursor is visible at a location that can accept text. Text entry is not possible (i.e., is not accepted by an application) when the text cursor is not visible. An application supports insert mode as the default for text entry and may also provide access to replace mode when it is more efficient given the task being performed. For example, replace mode may be more appropriate for text entry in fields with predefined attributes (e.g., latitude and longitude and date-time group), but insert mode may be preferred for entering free text (e.g., a message). If an application supports both modes, it does not arbitrarily switch between them as users move from one text area to another. A text box supports the standard shortcut keys for Cut, Copy, and Paste actions.

When variable-length text is entered, it is automatically justified or truncated; users do not have to enter leading characters to fill the space available. Text is displayed as typed (with the same capitalization and punctuation) unless a special format is required (e.g., composing a message). If a text box accepts only certain characters (e.g., numbers), feedback (e.g., an auditory signal) is provided when users enter an invalid character. The following guidelines apply to text entry:

- The amount of text that has to be entered should be minimized whenever possible; for example, users should be able to enter an abbreviation rather than an entire word and not have to enter the unit of measurement associated with a number value.

- Users should be allowed to enter text (e.g., dates) in any of several formats, with an application converting the entry into the format it requires.

- If an application supports geographic position hooking (i.e., it allows users to click on a map location and have the coordinates of the position entered into a lat/long field), it should indicate when this feature is available (e.g., in the label of the text box, with a special symbol next to the text box, in the status bar of the window).

Downloaded from http://www.everyspec.com

Users can enter text or skip the text box as desired; users are not forced to enter mandatory text before moving to the next text box or to move through a series of text boxes in a fixed sequence. If a text box contains a default value, tabbing into and out of the box does not affect the default. If users modify the default but do not save the change, the text box reverts to the default value when the window containing the text box opens again.

Autotabbing is available only when related text boxes support entry of a long text string (e.g., date, time, latitude/longitude). In this case, autotabbing is appropriate since users consider the characters to be a single data value and expect to enter the data without the need to tab between the text boxes; while separate text boxes are intended to improve readability and minimize the opportunity for error, they should not interfere with efficient data entry by users.

Error checking. An application performs error checking on the data entered and provides feedback (e.g., an auditory or visual warning) when errors are detected. Users are informed of an error (e.g., data in an incorrect format or outside range requirements) when they move focus out of the text box. Error feedback is provided in the status bar of the window if one is present; otherwise, feedback is provided by changing the appearance of the text box with the error and/or in a separate message window.

Text input can be saved automatically or only in response to an explicit user action (e.g., activating a Save, Apply, or OK push button). In both cases, an application checks for errors, informs users if any are present, and does not commit (i.e., save) the data until the errors have been corrected. If desired, an application can provide feedback (e.g., a confirmation message) to indicate the data have been saved.

Downloaded from http://www.everyspec.com

7.2 Combo Boxes (GNOME, MS Windows)[2]

7.2.1 Appearance

A combo box, shown in figure 7-3, is used to type an item or select it from a list. A label is included with a combo box that describes its contents.

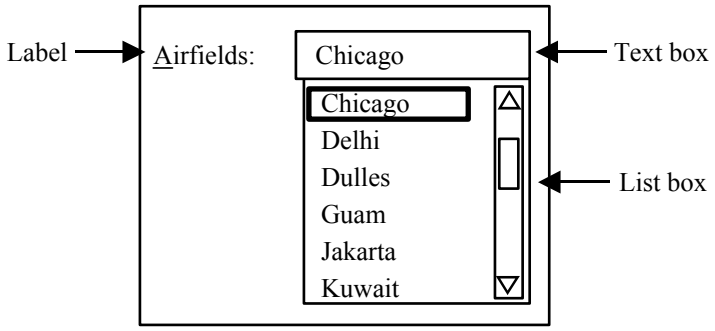


Figure 7-3. Example combo box.

A combo box consists of a text box and a list box that is attached to the text box. The text box shows the currently selected item. The text box and the list box are wide enough to display all of the items in the list. The text box may be constrained to accept entries that match items already in the list, or all entries in the text box may be accepted and added to the list. Text input is displayed as typed (with the same capitalization) but interpreted in a case-insensitive manner.

The list box is sized to display up to eight items at a time but may be larger if the number of items is very large. The list box includes a vertical scroll bar when the number of items exceeds the space available. The items are presented in a logical order (e.g., alphabetical) based on the nature of the items. If an item is temporarily unavailable, it is omitted from the list box (rather than grayed out). If users need to select none of the items in the combo box, an item labeled None should be included in the list box; this item is displayed in the text box if selected.

7.2.2 Behavior

Clicking BLeft on an item in the list box selects it and displays it in the text box.

> **NOTE**: In GNOME, BMiddle and BRight also perform a selection in a combo box. The desired implementation is as described above.

[2] Java documentation describes an editable and non-editable combo box, but these controls have the appearance and behavior of a drop-down combo box and a drop-down list box, respectively. Motif documentation shows a combo box with the appearance of the list and text box included in a Motif selection box; i.e., the list box is placed above and visually separated from the text box below.

Downloaded from http://www.everyspec.com

When a combo box has focus, typing in the text box replaces the existing text with the text being entered; if an item in the list matches the text, the item is selected. A combo box supports the standard shortcut keys for Cut, Copy, and Paste actions.

When a combo box has focus, UP and DOWN navigate between list items, with no wrapping from the bottom to the top item. Navigating to a list item selects it and displays it in the text box.

> **NOTE**: In GNOME, the list box and the text box are defined as separate tab groups, rather than the combo box behaving as a single integrated control. The desired implementation is as described above.

Downloaded from http://www.everyspec.com

7.3 Drop-Down Combo Boxes

7.3.1 Appearance

A drop-down combo box, shown in figure 7-4, is used to type an item or select it from a list that is displayed on demand. A label is included with a drop-down combo box that describes its contents.

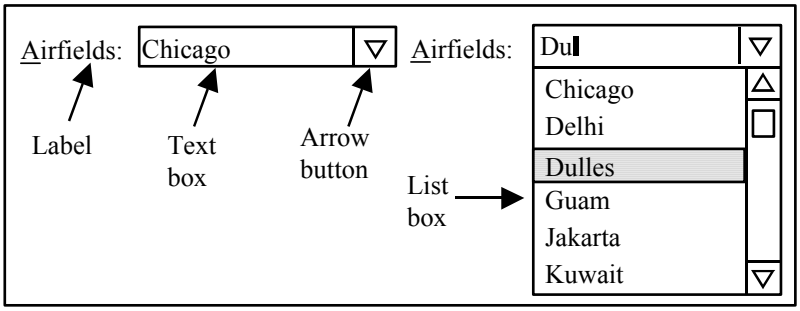


Figure 7-4. Example drop-down combo box.

A drop-down combo box consists of a text box and a down-pointing arrow button which, when activated, displays a list box. The text box shows the currently selected item. The text box and the list box are wide enough to display all of the items in the list. The text box may be constrained to accept entries that match items already in the list, or all entries in the text box may be accepted and added to the list. Text input is displayed as typed (with the same capitalization) but interpreted in a case-insensitive manner.

The list box is sized to display up to eight items at a time but may be larger if the number of items is very large. The list box includes a vertical scroll bar when the number of items exceeds the space available. The items are presented in a logical order (e.g., alphabetical) based on the nature of the items. If an item is temporarily unavailable, it is omitted from the list box (rather than grayed out). If users need to select none of the items in the drop-down combo box, an item labeled None should be included in the list; this item is displayed in the text box if selected.

7.3.2 Behavior

Pressing BLeft on the arrow button displays the list box; releasing BLeft on an item selects it, displays the item in the text box, and dismisses the list box. In addition, clicking BLeft on the arrow button displays the list box; clicking BLeft on an item selects it, displays the item in the text box, and dismisses the list box. Releasing BLeft outside the list box, clicking BLeft on the arrow button a second time, or clicking outside the drop-down combo box dismisses the list box without selecting an item.

NOTE: In GNOME, BMiddle and BRight also perform a selection in a drop-down combo box. In Java, BRight also performs a selection

Downloaded from http://www.everyspec.com

in a drop-down combo box. The desired implementation is as described above.

When a drop-down combo box has focus, typing in the text box replaces the existing text with the text being entered. A drop-down combo box supports the standard shortcut keys for Cut, Copy, and Paste actions.

GNOME, Motif: When a drop-down combo box has focus, DOWN cycles through items in the list box and displays them in the text box without opening the list.

When a drop-down combo box has focus, ALT+DOWN displays the list box.

GNOME: When a drop-down combo box has focus, RETURN also displays the list box.

Java: When a drop-down combo box has focus, DOWN and ALT+UP also display the list box.

Motif: When a drop-down combo box has focus, DOWN also displays the list box.

MS Windows: When a drop-down combo box has focus, DOWN, UP, ALT+UP, and F4 also display the list box.

When the list box is displayed, UP and DOWN navigate between items, with no wrapping from the bottom to the top item.

GNOME, MS Windows: When the list box is displayed, RETURN selects the highlighted item, displays the item in the text box, and dismisses the list box. ESC dismisses the list box without changing the selection in the text box.

Java, Motif: When the list box is displayed, navigating to a list item selects it and displays it in the text box; RETURN dismisses the list box. ESC dismisses the list box and maintains the latest selection in the text box.

Downloaded from http://www.everyspec.com

7.4 Spin Boxes (GNOME, Motif, MS Windows)

7.4.1 Appearance

A spin box, shown in figure 7-5, is used to select from a range of values that have an ordered or natural sequence (e.g., numeric, chronological, alphabetical). A spin box can be non-editable if it includes all possible values or editable if it does not include all possible values. A label is included with a spin box that describes its contents.

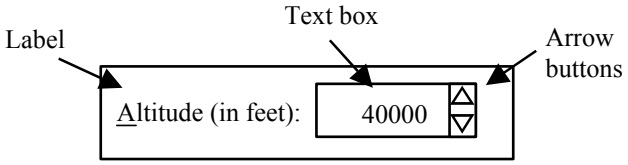


Figure 7-5. Example spin box.

A spin box that is editable consists of a text box and a pair of arrow buttons; a spin box that is not editable consists of a text-display box and a pair of arrow buttons. The arrow buttons are oriented vertically (i.e., one up-pointing and one down-pointing), and placed on the right edge of the spin box.

A spin box contains no more than 20 ordered values unless the data set naturally contains more values (e.g., the days in a month).

GNOME, Motif, MS Windows: The values in a spin box are left-aligned.

Java: The values in a spin box are right-aligned.

If spin boxes are used to enter data such as date/time group or latitude/longitude, separate spin boxes should be provided for each part of the entry. Spin boxes can be used with text boxes for data entry (e.g., separate spin boxes for month and day and a text box for typing the year).

7.4.2 Behavior

Clicking BLeft on an arrow button changes the value by one unit in the arrow direction. Pressing BLeft on an arrow button moves continuously through the values in the arrow direction.

GNOME: Clicking BRight on an arrow button moves to the first and last values.

When a spin box has focus, UP and DOWN change the value by one unit in the arrow direction. When an editable spin box has focus, typing in the text box replaces the existing value with the text being entered.

Downloaded from http://www.everyspec.com

7.5 Scales

7.5.1 Appearance

A scale, shown in figure 7-6, is used to set a value in a predefined ordered range. A label is included with a scale that describes its contents. A display-only scale can be used as a progress bar (see section 8.3.2.1) to present values that users cannot change.

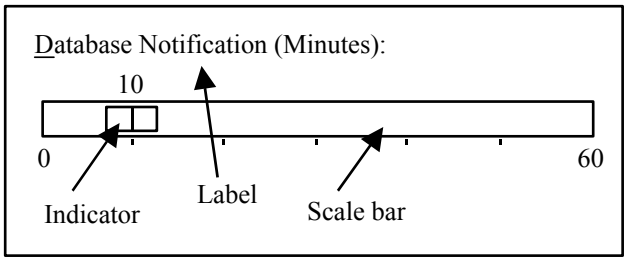


Figure 7-6. Example scale.

A scale consists of a scale bar representing the full range of values that can be selected and an indicator for setting a scale value. A scale may also include arrow buttons for moving the indicator, tick marks representing the range of available values, and labels indicating the minimum and maximum values for the scale. If users need to set an exact value, a text box for entering the value can be included with the scale. If the scale includes a text box, the scale indicator and text box should be linked so each is automatically updated when users select or enter a value. A scale can be oriented horizontally or vertically.

7.5.2 Behavior

Clicking BLeft on a scale bar moves the indicator one or more steps in the pointer direction. Dragging the indicator using BLeft moves it in the pointer direction. If the scale has arrow buttons, clicking BLeft on an arrow moves the indicator one step in the arrow direction; pressing BLeft on an arrow moves the indicator continuously in the arrow direction.

When a scale has focus, the arrow keys move the indicator one step in the arrow direction.

Downloaded from http://www.everyspec.com

7.6 Tab Controls

7.6.1 Appearance

A tab control is used to display multiple views of information within a single window. For example, a tab control, shown in figure 7-7, can be used in a Properties window to display the attributes for an object (e.g., the font and alignment properties of text). A tab control is also used to display groups of related controls when all of the groups do not need to be visible in the window at the same time.

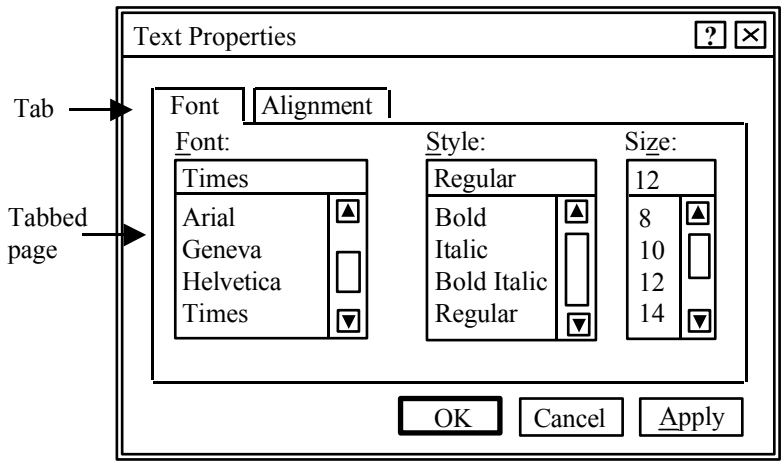


Figure 7-7. Example tab control.

A tab control consists of a set of tabbed pages, each identified by a tab. The tab has a label that describes the page contents. The label is centered in the tab. If the label contains text and a graphic, the graphic is placed to the left of the text. A tab is wide enough to display the label. If a tab has an unusually long label, the text can be truncated and followed by an ellipsis, or the tab can be sized to accommodate the label.

A tab control supports placement of tabs along the top, bottom, left, or right edges of the control; by default, the tabs are placed along the top edge. The control contains at least two but no more than six tabs, arranged in a single row. If space is insufficient to display all of the tabs in the row (e.g., because the window has been resized smaller), the tabs can be clipped based on the space available, and arrow buttons and/or a scroll bar displayed at the end of the row to scroll the visible tabs. A tab control should not contain multiple rows of tabs; instead, a window can include a list box with items that, when selected, display the “page” contents in the window. In addition, multiple tab controls are not nested within a window.

Each tab page presents a group of related controls, with a consistent arrangement of controls used across the pages. Controls such as push buttons that affect a single page are placed on that page; controls that affect every page are placed outside the page (e.g., on the window itself). A page is included in a tab control even if all of the controls on the

Downloaded from http://www.everyspec.com

page are unavailable. Selecting a different tab does not change the order of the tabs or automatically apply any changes made to controls on the previous page. Changing a control setting on one page does not affect the control settings on any other page.

7.6.2 Behavior

Clicking BLeft on a tab raises that page to the foreground.

> **NOTE**: In GNOME, BMiddle and BRight also perform a selection in a tab. In Java, BRight also performs a selection in a tab. The desired implementation is as described above.

GNOME: When a tab control has focus, LEFT and RIGHT navigate between tabs, with no wrapping from the last to the first tab.

Java, Motif, MS Windows: When a tab control has focus, LEFT and RIGHT navigate between tabs, with wrapping from the last to the first tab.

MS Windows: When a tab control has focus, CTRL+TAB and SHIFT+CTRL+TAB also navigate between tabs, with wrapping from the last to the first tab.

Moving focus to a tab raises it to the foreground. When a tab has focus, TAB navigates between it to the controls on the page, with wrapping from the bottom control to the tab.

Downloaded from http://www.everyspec.com

7.7 Scrolled Areas

7.7.1 Appearance

A scrolled area is used when the information being viewed exceeds the space available in a window. A scrolled area contains vertical and/or horizontal scroll bars. A vertical scroll bar controls backward and forward movement; a horizontal scroll bar controls left and right movement. A scroll bar is located to the right and/or at the bottom of the area. A scroll bar should remain displayed even if the information no longer requires scrolling.

A scroll bar consists of a scroll bar shaft, a scroll box, and scroll arrows, as shown in Figure 7-8. The scroll bar shaft represents the length of the information that users can scroll, while the scroll box represents the visible portion of the information. The position of the scroll box indicates the relative position of the information displayed in a scrollable area. The size of the scroll box is proportional to the amount of information that is visible; the scroll box fills the shaft if all of the information is visible. The scroll arrows scroll incrementally through the information and indicate the direction of the scrolling movement. The information in a scrollable area is updated continuously as it is scrolled. Scrolling occurs to the top or the bottom of the information but not beyond.

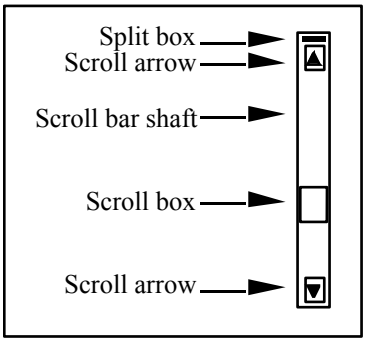


Figure 7-8. Example scroll bar in MS Windows.

7.7.2 Behavior

Clicking BLeft on a scroll arrow scrolls one unit (e.g., one line or column) in the arrow direction. Clicking BLeft on the scroll bar shaft scrolls one large unit at a time (e.g., one length or width of the visible area, minus one unit for overlap). Dragging the scroll box using BLeft scrolls the visible area in the pointer direction.

> GNOME: Clicking BRight on a scroll arrow scrolls to the top/bottom of a scrollable area.

Downloaded from http://www.everyspec.com

Scrolling occurs automatically when users drag an object to the edge of a scrollable area; scrolling is in the direction of the drag. Scrolling does not affect the select state of objects in the scrollable area or move the input focus.

When a scrollable area has focus, the arrow keys scroll one unit in the arrow direction. PAGEUP and PAGEDOWN scroll one large unit (e.g., a page).

Downloaded from http://www.everyspec.com

7.8 Paned Areas

7.8.1 Appearance

A paned area is used to display simultaneous views of the same data in a single window. Panes can be arranged horizontally or vertically and can be either fixed or adjustable by users. If desired, one paned area can be nested inside another. A window that contains paned areas should be resizable so users can increase the size of the panes in the window.

A split bar is a separator that defines the boundary between the paned areas in a window. When users split a window, the resulting panes include scroll bars if the information exceeds the space available in the pane. Users can scroll each pane (i.e., perpendicular to the direction of the split) independently within the window. Making one pane larger makes the other pane smaller but does not affect the overall size of the window.

GNOME, Motif: The split bar has a sash that adjusts the size of the panes. The sash is located near the right end of the split bar in a vertically-split paned window (as shown in figure 7-9) and near the bottom end of the split bar in a horizontally-split paned window.

Java: The split bar may include zoom buttons (i.e., up/down- or left/right-pointing arrows) that adjust the size of the panes. If the buttons are present, they are located near the left end of the split bar in a vertically-split paned window and near the top end of the split bar in a horizontally-split paned window.

MS Windows: A split box is used to adjust the size of the panes. A split box is a solid box located at the top end of a vertical scroll bar (as shown in figure 7-8) or at the left end of a horizontal scroll bar.

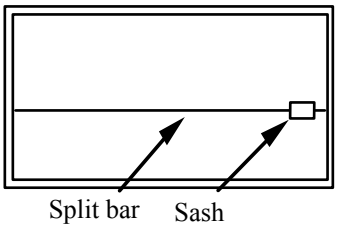


Figure 7-9. Example sash.

7.8.2 Behavior

GNOME, Motif: Dragging a sash using BLeft moves the split bar in the pointer direction. Dragging a sash to either end of the window reduces the pane to its minimum size but does not remove the split bar from the window.

Java: Dragging a split bar using BLeft moves the bar in the pointer direction. If zoom buttons are present in the split bar, clicking BLeft on a button moves the bar

Downloaded from http://www.everyspec.com

to its extreme (upper, lower, left, or right) position in the window. If the split bar is in this position, clicking BLeft on a zoom button restores the bar to its previous position within the window.

MS Windows: Dragging a split box using BLeft displays a split bar and divides the window into panes. Dragging a split box to either end of the window closes the pane in the direction of the drag.

GNOME, Java: F8 moves focus to the split bar.

Motif: TAB navigates to the sash on a split bar.

When a split bar has focus, the arrow keys move it in the arrow direction.

MS Windows: Focus cannot be moved to the split bar from the keyboard. However, an application can include a Split option in the Window or View menu. Activating this option splits the window (e.g., in the middle). Selecting this option can also enable split mode and move focus to the split box. When split mode is enabled, the arrow keys move the split bar in the arrow direction; RETURN sets the location of the split bar and exits split mode. ESC cancels the split operation and exits split mode.

Downloaded from http://www.everyspec.com

8. Window Design[1]

8.1 Primary Windows

A primary window includes a window title in the title bar and a client area where users perform the task(s) supported by the window. The client area normally contains a menu bar, one or more toolbars, and a status bar, as shown in figure 8-1. The remainder of the client area provides a workspace where users view and interact with information. The client area can include a scrolled area if space to display information is limited, or it can use a paned area or tab control to present multiple views of information. This section provides specifications on menu bars, toolbars, and status bars; specifications on scrollable areas, paned areas, and tab controls are included in section 7.

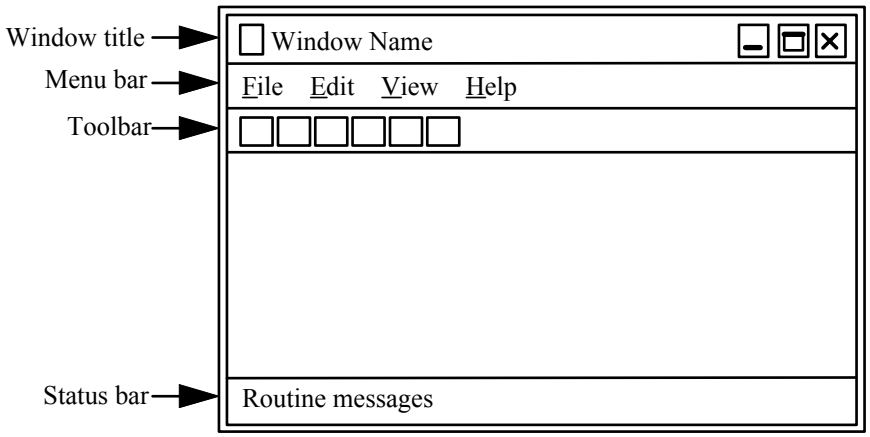


Figure 8-1. Example primary window.

8.1.1 Window Titles

GNOME: The window title is the application name. If an application is document-based, the title is the document name, followed by the application name if desired, using the format “Document Name – Application Name. “

Java: The window title is the document name, followed by the application name, using the format “Document Name – Application Name.”

Motif: The window title is the application name, followed by the document name if desired, using the format “Application Name – Document Name.”

MS Windows: The window title includes a small version of the application icon and the application name. If an application is document-based, the window title

[1] The specifications in this section describing Java style are taken from both the Java Look and Feel Design Guidelines and the Java Look and Feel Design Guidelines: Advanced Topics. The guidelines in Advanced Topics were used if there were differences in the direction provided by the two documents.

Downloaded from http://www.everyspec.com

includes a small version of the document type icon and the document name, followed by the application name if desired, using the format "Document Name – Application Name."

An application does not include extraneous information (e.g., the application version number) in a window title and does not place its own controls in the title bar. If users provide the name used in a window title (e.g., when they save a file), the title is displayed as entered by the user. If a window contains a new (i.e., unnamed) file, the window should be assigned a default name (e.g., Untitled), with each instance of the name numbered (e.g., Untitled 1, Untitled 2, etc.). If the name of the object being viewed in the window changes (e.g., the user renames the object), the window title is also updated.

8.1.2 Window Content

8.1.2.1 Menu Bars

A primary window contains a single menu bar. The menu bar can occupy a fixed position within the window, or it can be movable.

GNOME, Java, MS Windows: If a menu bar is movable (i.e., supports docking), it has a grip handle located at the left edge of the menu bar. Dragging the grip handle using BLeft detaches the menu bar; releasing BLeft either attaches the menu bar to another edge of the window or displays it in a separate window. Dragging the window to the edge of its parent window using BLeft reattaches the menu bar to the parent.

GNOME, Java, and MS Windows: The menu titles begin at the left margin of a menu bar and extend rightward.

Motif: The menu titles begin at the left margin of a menu bar and extend rightward, with Help at the right margin of the menu bar.

The space between menu titles should be sufficient (at least three character widths) so multi-word titles can be distinguished from single-word titles.

Each interface defines a set of common menus that are used frequently in applications and recommends that the menus be used when an application supports the associated functions. Table 8-1 lists the contents of the menus for each interface and the order in which they appear in a menu bar. An application includes common menus in a menu bar if they support the associated functions. Any application-specific menu options are added to the appropriate menu and grouped with the options to which they relate. The menus in table 8-1 perform the following functions:

- A File menu contains options that manipulate the object represented in a window. If the title of this menu is not File, it is an application-specific term describing the type of object(s) the window operates on.

Downloaded from http://www.everyspec.com

- An Edit menu contains options that change the content of an object and locate specific information in the object.
- A View menu contains options that change the view of the current object but do not change the object itself. This menu can also contain options for controlling the display of window components such as a toolbar or status bar.
- An Insert menu contains options that list the types of objects that can be inserted in a window.
- A Format menu contains options that change the visual appearance of an object (e.g., font, style) and are saved as part of the object.
- A Tools menu contains options that launch the tools provided in an application.
- A Go menu contains options that navigate within an object or collection of objects.
- A Bookmarks menu contains options that allow users to browse objects (e.g., files, Web pages).
- An Options menu contains options that control the behavior of an application.
- A Window menu contains options that apply to the open windows in an application.
- A Help menu contains options that access information about a window or application.

NOTE: Table 8-1 lists the options in a Help menu for each interface. Guidelines on these menu options can be found in the style documentation provided by the interface. The specifications presented in section 9.2 assume that an application provides access to user support resources using the Web and online documentation services provided by the COE.

GNOME: At a minimum, a menu bar includes a File and a Help menu. Any application-specific menus are inserted following the Format menu.

Java: At a minimum, a menu bar includes a File and a Help menu. Any application-specific menus are inserted following the Edit or the View menu.

Motif: At a minimum, a menu bar includes a File, a View, and a Help menu. Any application-specific menus are inserted following the View menu.

Downloaded from http://www.everyspec.com

Table 8-1. Common menus.

	GNOME	Java	Motif	MS Windows
File Menu	New Open… <separator> Save Save As… Save a Copy… Revert <separator> Page Setup Print Preview… Print… Send To… <separator> Properties <separator> <list of recently used files> <separator> Close Quit	New Open… Close <separator> Save Save As… <separator> Page Setup Print… <separator> Preferences <separator> <list of recently used files> <separator> Exit	New Open… Save Save As… <separator> Print… Print Setup <separator> Close Exit	New Open… Close <separator> Save Save As… <separator> Page Setup… Print Preview Print… <separator> Send To Properties <separator> <list of recently used files> Exit
Edit Menu	Undo Redo <separator> Cut Copy Paste Paste Special… Duplicate Delete <separator> Select All Deselect All <separator> Find… Find Next Find Previous Replace… <separator> Insert… <separator> Preferences	Undo Redo <separator> Cut Copy Paste Paste Special… Delete <separator> Find… Find Again Replace… Select All	Undo Redo <separator> Drag <separator> Cut Copy Paste <separator> Clear Delete <separator> Select All Deselect All	Undo Redo <separator> Cut Copy Paste Paste Special <separator> Delete Select All Duplicate <separator> Find… Replace… Go To… <separator> Links…

Downloaded from http://www.everyspec.com

Table 8-1. Common menus (continued).

	GNOME	Java	Motif	MS Windows
View Menu	Toolbar Status Bar <separator> Icons List Details <separator> Sort By… Filter… <separator> Zoom In Zoom Out Normal Size Best Fit <separator> Refresh	Large Icons Small Icons List Details <separator> Sort By… Filter <separator> Zoom In Zoom Out <separator> Refresh	Change View New Window	<list of available views> <separator> Toolbar Status Bar <separator> Zoom…
Insert Menu	Page Break Date and Time… Symbol… <separator> Sheet… Rows… Columns… <separator> Image… Graph… <separator> From File… External Link…			Break… Page Numbers… Date and Time… Symbol… <separator> Picture Diagram… File… Object… Hyperlink…
Format Menu	Style… <separator> Font… Paragraph… <separator> Bold Italic Underline <separator> Cells… List… <separator> Layer… Page…	Font Style Size <separator> Align Left Align Center Align Right		Font… Paragraph… <separator> Cells… Row Column <separator> Theme… Style…
Tools Menu				None specified.

Downloaded from http://www.everyspec.com

Table 8-1. Common menus (continued).

	GNOME	Java	Motif	MS Windows
Go Menu	In a document-based application: Previous Page Next Page Go to Page… <separator> First Page Last Page In a browser-based application: Back Forward Up <separator> Home Location…			
Bookmarks Menu	Add Bookmark Edit Bookmarks…			
Options Menu			None specified.	
Window Menu	<list of open windows> <separator> Save All Close All		New Window <list of window arrangements> <separator> <list of open windows>	New Window Arrange All Cascade Split <separator> <list of open windows>
Help Menu	Contents About	Contents Tutorial Index Search… <separator> About <application name>	Overview Table of Contents Tasks Reference <separator> On Item <separator> Using Help <separator> About <application name>	<application name> Help <separator> What's This? <separator> About <application name>

NOTE: The MS Windows style guide defines common menus but does not identify the options to include in the menus. The menu contents listed here are based on implementations in COTS products such as Microsoft Office.

Downloaded from http://www.everyspec.com

8.1.2.2 Toolbars

A toolbar provides access to frequently used actions available elsewhere in a primary window. The default location of the toolbar is immediately below the menu bar in the window. A window can contain a single toolbar, or it can provide multiple toolbars, each performing a related set of functions (e.g., drawing, editing). A toolbar can occupy a fixed position within a window, or it can be movable.

> GNOME, Java, MS Windows: If a toolbar is movable (i.e., supports docking), it has a grip handle located at the left edge of the toolbar. Dragging the grip handle using BLeft detaches the toolbar; releasing BLeft either attaches the toolbar to another edge of the window or displays it in a separate window. Dragging the window to the edge of its parent window using BLeft reattaches the toolbar to the parent.

> GNOME, MS Windows: If some of the buttons in a toolbar are hidden (e.g., because the window has been resized smaller), the toolbar includes a “>>” button at the right end of the toolbar that, when selected, displays a menu of the hidden buttons.

An application allows users to configure toolbar content and visibility. Users are able to show or hide individual toolbars, change the order and appearance of toolbar buttons, and save and restore a toolbar configuration.

A toolbar contains a set of toolbar buttons and may also include other controls such as drop-down list boxes and spin boxes. The toolbar buttons begin at the left edge of a horizontally oriented toolbar and extend rightward. All of the buttons in the toolbar are the same size. The buttons are arranged in the same order as the actions (e.g., menu options) to which they provide redundant access. Buttons for related actions are placed together and separated from other buttons by a separator or blank space.

> GNOME, Java, MS Windows: If a toolbar supports mouse-over borders, the button borders are invisible, and there is no space between the buttons in the toolbar. The button border is visible only when the pointer is placed on the button.

An application uses tooltip help (see section 9.1.2) to provide feedback about the function performed by toolbar buttons. The wording of the tooltip is the same as that used to describe the action elsewhere in the window. For example, if a toolbar button provides redundant access to the Cut option in an Edit menu, the tooltip for the button would be Cut. An application allows users to show or hide tooltips (e.g., from an option in the View menu).

> Motif: If an application does not support tooltips, it includes a status bar in a window with a toolbar and uses it to provide feedback about the function performed by toolbar buttons.

Downloaded from http://www.everyspec.com

At a minimum, a window includes a “main” or “standard” toolbar that performs some of the most common actions in the application. If this toolbar includes any of the following actions, they are ordered: New, Open, Save, Print, Cut, Copy, Paste, and Help. The buttons for any application-specific actions are inserted after Paste and arranged in an order expected by users.

The appearance of a toolbar button is linked to that of the menu option or control for which it provides redundant access; for example, if a Copy menu option becomes unavailable, the Copy button in the toolbar is also shown as unavailable. When a toolbar button is selected, its action is applied immediately; users do not have to activate a push button in the window to invoke the property. If a toolbar button is linked to a menu option or control whose action opens a dialog window (e.g., to request additional input from users), the toolbar button can either open the dialog or it can execute the action using the default values associated with the action.

8.1.2.3 Status Bars

A status bar presents information for which no user response is needed. The status bar is located at the bottom of a primary window and is separated from other window contents by empty space or a separator. The status bar can display a single line of text that extends the width of the window, or it can be divided into parts to display more than one type of feedback. An application allows users to show or hide the status bar (e.g., from an option in the View menu).

A status bar presents information for which no user response is needed (e.g., feedback about events that occurred as expected, information about the object under the pointer, the status or current state of an application. If the status bar provides feedback on the completion of a process, dots can be added dynamically to the message (e.g., Loading database…) to indicate that the process is still executing.

A status bar is blank if there is no status-related information to present. If information is displayed in the status bar, it is removed as soon as it is no longer relevant to the current state of the window or the current object with focus. For example, when the status bar is used to indicate the status of an action, a progress message is displayed when the action is initiated (e.g., “Drawing map...”), updated when the action is completed (e.g., “Drawing map...Done”), and removed within 5 seconds of action completion.

Downloaded from http://www.everyspec.com

8.2 Dialog Windows

A dialog window, shown in figure 8-2, includes a window title in the title bar and a client area where users interact with the controls needed to perform a subtask or function.

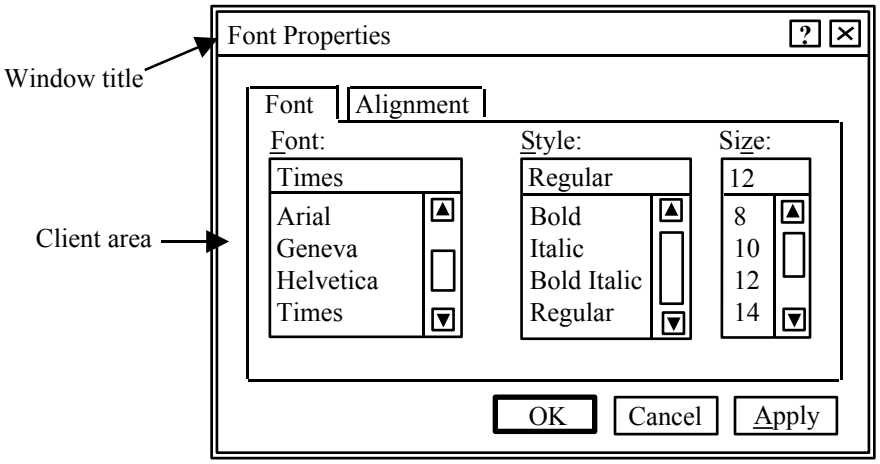


Figure 8-2. Example dialog window.

8.2.1 Window Titles

GNOME, Java: The window title is the name of the command that opened the window.

Motif: The window title contains the application name, then describes the purpose of the window, using the format “Application Name : Window Purpose.”

MS Windows: The window title describes the purpose of the window.

An application does not include extraneous information (e.g., words such as “dialog”) in a window title and does not place its own controls in the title bar.

8.2.2 Window Content

8.2.2.1 Arrangement of Controls

A dialog window contains all of the controls required by users to perform the function supported by the window; there are no “invisible” controls that are invoked only via a shortcut key, a pop-up menu, or drag and drop.

GNOME: A dialog window can include a menu bar, instead of push buttons, if the number of controls required in the window would be excessive. The specifications in section 8.1.1.1 apply to the design of menu bars in dialog windows. A dialog window does not have a toolbar or a status bar.

Downloaded from http://www.everyspec.com

Java, Motif, MS Windows: A dialog window does not have a menu bar, toolbar, or status bar.

Related controls are grouped together and arranged from left to right, top to bottom in a window in the order in which users are expected to interact with them. The window is designed so users can perform the function with a minimum amount of pointer movement and/or number of keystrokes and without having to refer to information not included in the window.

The following guidelines address how to select the controls to include in a dialog window:

- Use radio buttons, an option menu, or a list box when users need to select from discrete values or choices; use a scale or spin box when users need to select from a continuous range of values.

- Use radio buttons when users need to see all of the settings available in a group; use an option menu or drop-down list box when users need to see only the current setting.

- Use radio buttons or an option menu when the options from which users choose are not likely to change; use a list box when the options from which users choose may change.

- When users have to make a single selection, use a group of radio buttons if there are up to seven choices; use an option menu if there are up to ten choices; and use a list box or drop-down list box if there are more than ten choices.

- When users have to make multiple selections, use check boxes if there are up to seven choices; use a list box if there are more than seven choices.

- Use push buttons for frequently executed actions when space is available to display the buttons; use tear-off menus for these actions if space is limited.

- Use a scale when users need to enter approximate values; use a spin box or include a text box with a scale when users need to enter precise values.

- Use a list box when the set of choices from which users select is known; use a drop-down list box instead of a list box when space in the window is limited. Use a combo box when the set of choices from which users select is indeterminate and users need to enter a choice if it does not appear in the list; use a drop-down combo box instead of a combo box when space in the window is limited.

Spacing, headings, and group boxes. Spacing and headings are used to organize window contents. The heading is placed above or to the left of the group of associated controls

Downloaded from http://www.everyspec.com

and is not followed by a colon. The heading can be displayed in normal or bold font. The heading and associated controls are both left-aligned, with the controls indented below the heading if desired. The same spacing and alignment of headings and controls are used in all of the dialog windows.

Java, Motif, MS Windows: Group boxes are used in a window if spacing and headings are insufficient to organize its contents. A group box, shown in figure 8-3, is a frame drawn around related controls and may include a label that describes its contents. A group box is normally included with groups of radio buttons, check boxes, or text boxes and not used with push buttons or a single control (e.g., a list box) or in a window with only one tab group. A group box contains all of the controls in a tab group. Group boxes are not placed inside other group boxes in a window.

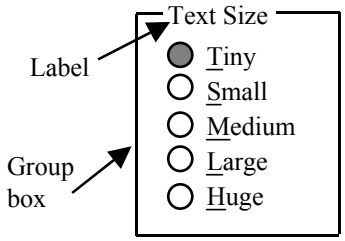


Figure 8-3. Example group box.

If the controls in a window include labels that describe their contents, each label is placed to the left of or above the control to which it relates and is followed by a colon. A label placed above a control is aligned with the left edge of the control. If a control is temporarily unavailable, its label is grayed out.

When the controls are arranged vertically in a window, the labels and controls are both left-aligned. If the labels vary greatly in length, the labels can be right-aligned and the text boxes left-aligned, as shown in figure 8-4, or the labels can be reworded to be more similar in length. When a text box or a control that contains a text box (e.g., a combo box) is placed in a window, the baseline of its label is aligned with the baseline of the text in the text box.

Downloaded from http://www.everyspec.com

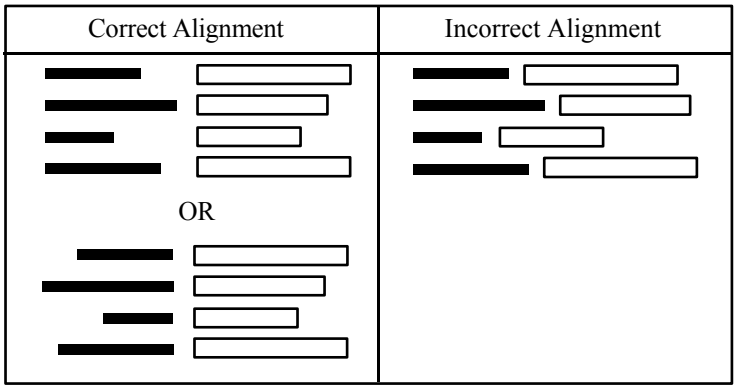


Figure 8-4. Correct and incorrect alignment of labels and controls.

Mnemonics. Table 8-2 lists interface-specific requirements related to the assignment of mnemonics to interface components in dialog windows. An application complies with these requirements. In general, mnemonics are assigned to all of the controls in a dialog window, except for the default push button (usually OK) and Cancel which are already mapped to RETURN and ESC. Mnemonics are included either in a control or in the label provided with a control. In the latter case, typing the mnemonic moves focus to the control, not the label. An application assigns a unique character as the mnemonic for each control within a context (e.g., the radio buttons in a group).

Table 8-2. Assignment of mnemonics to interface components in dialog windows.

	GNOME	Java	Motif	MS Windows
Check box labels	Yes	Yes	Yes	Yes
Combo box items	No	-----	No	No
Combo box labels	Yes	-----	Yes	Yes
Drop-down combo box items	No	No	No	No
Drop-down combo box labels	Yes	Yes	Yes	Yes
Drop-down list box items	No	No	No	No
Drop-down list box labels	Yes	Yes	Yes	Yes
Group box labels	-----	No	No	No
Headings	No	No	No	No
List box column headings	No	No	No	No
List box items	No	No	No	No
List box labels	Yes	Yes	Yes	Yes
Option menu options	Yes	-----	No	-----
Option menu labels	Yes	-----	Yes	-----
Push button labels	Yes	Yes	Yes	Yes
Radio button labels	Yes	Yes	Yes	Yes
Scale labels	Yes	Yes	Yes	Yes
Spin box labels	Yes	-----	Yes	Yes
Tab labels	No	Yes	Yes	Yes
Text box labels	Yes	Yes	Yes	Yes
Toolbar button labels	No	No	No	No

Downloaded from http://www.everyspec.com

Window settings. When a dialog window opens, all of the controls in the window are set to reflect the current state of the application. For example, a dialog window that allows users to change the font size is displayed with the current font size selected. If the window contains a list box, the currently selected item is visible in the box. If the window contains a tab control, the same page is visible when the window re-opens as when the window closed.

8.2.2.2 Push Buttons[2]

Push buttons are displayed individually or in groups in a dialog window. All of the push buttons in a group are the same size, with equal spacing between buttons, and use the same type of label (i.e., text, graphic, or both). If the push buttons have text labels, the buttons are wide enough to display the longest text. If the push buttons have graphic labels, the push buttons are large enough to display the largest graphic. Exceptions may occur in order to accommodate a button with a label that is significantly longer or larger than the others in the group, especially when space in a window is limited.

A dialog window contains at least one push button that performs an action in the window and dismisses it or that dismisses the window without taking any action. In a single-use window (i.e., one where users perform one operation and dismiss the window), OK applies the changes and closes the window, and Cancel ignores the changes and dismisses the window. In a multiple-use window (i.e., one where users perform several actions before dismissing the window), Apply applies the changes but does not dismiss the window, and Close ignores the changes and dismisses the window. If desired, the OK and Apply push buttons can be renamed to describe the action performed by the button (e.g., Print, Find), and multiple push buttons that perform OK-type and Apply-type actions can be included in the window.

Default push buttons. If a default push button is designated in a dialog window, it is the action that users are most likely to select (e.g., OK in a single-use window). When a window opens, the default push button is one that performs a nondestructive action. If a dialog window contains controls that require mandatory input, the default push button is shown as unavailable when the window opens and remains so until users have provided the mandatory input.

The default designation can move to a different push button depending on the control that has focus in a window. When focus moves to a push button, its action is the default, and it is shown with a default border. The default moves with the location cursor during keyboard navigation in the push button group and returns to the original button when focus leaves the group.

[2] These specifications assume that a dialog window is “explicit apply” (i.e., the choices made in a window are applied only after the user selects OK or an equivalent push button). Dialog windows that are “instant apply” (the choices made in a window are applied immediately) are addressed in section 8.2.3.

Downloaded from http://www.everyspec.com

Push button placement and order. The push buttons are separated from the other controls in the window by a separator or blank space.

GNOME: Push buttons are arranged horizontally along the bottom edge of a window; the buttons start in the lower right corner and extend leftward, with the Help button, if present, in the lower left corner.

Java: Push buttons are arranged horizontally along the bottom right edge of a window.

Motif: Push buttons are arranged horizontally along the bottom edge of a window.

MS Windows: Push buttons are arranged either horizontally along the bottom right edge of a window or vertically along the upper right edge of a window.

If push buttons can affect different controls in a window, they are placed to the right of the control(s) and aligned vertically.

GNOME: Push button order is OK on the right, preceded by Cancel, any other actions (e.g., Apply), and Help (if present) on the left. If a default action is identified, it is the rightmost push button in the group.

Java, Motif: Push button order is OK on the left, followed by any other actions, Cancel, and Help (if present) on the right. If a default action is identified, it is the leftmost push button in the group.

MS Windows: Push button order is OK on the left, followed by Cancel, any other actions, and Help (if present) on the right. If a default action is identified, it is the leftmost push button in the group.

8.2.3 Types of Dialog Windows

GNOME, Java: A utility window is a dialog window that remains open when its parent window is closed. This type of dialog is used to display a collection of tools, colors, or patterns (e.g., in a tools palette window). A utility window follows the same design guidelines as other dialog windows.

GNOME, Java: An instant apply window is a dialog window where user choices take effect automatically, without users selecting OK or an equivalent push button in the window. This type of dialog is used to select preferences or set properties (e.g., in a Preferences or Properties window). An instant apply window contains a Close push button that closes the window after user choices have taken effect. This push button is not the default in the window.

NOTE: An instant apply window violates the object-action paradigm (see section 3.5.1) which calls for user choices to take effect only

Downloaded from http://www.everyspec.com

after an explicit action. In addition, this type of window does not provide the capability to undo a change in settings, forcing users to remember what the original settings were if they want to revert to what was in effect before they made the changes.

GNOME, Java, MS Windows: An assistant window is a dialog window that guides users through the steps in a task. The window presents a sequence of pages, each containing information and/or controls concerning the current step in a task and push buttons for navigating between the pages. Each page has a default navigation button that allows users to move through the steps in the assistant. Any user input on a page is verified before moving to the next page.

Motif, MS Windows: An expandable window is a dialog window with an optional control area for accessing additional settings and/or actions. This type of dialog is used to manage window size when screen space is limited. For example, a Print window might include a common area for entering print parameters, with an optional area available for displaying application-specific print functions. An expandable window includes an expand-action button (e.g., More) that toggles between showing and hiding the optional control area.

Table 8-3 lists the pre-built dialog windows that are provided by each interface for performing common functions. An application uses these common windows whenever it performs one of the functions they support. The style documentation provided by each interface contains guidelines on the design of common dialog windows. If desired, an application can extend or customize these windows but should maintain consistency with the basic window design as defined by the interface.

Table 8-3. Common dialog windows.

GNOME	Java	Motif	MS Windows
Color window	About window	About window	Browse for Folder window
Font window	Color Chooser window	Command window	Color window
Open window	Find window	File Selection window	Find window
Print window	Login window	Open window	Font window
Save window	Preferences window	Print window	Open window
	Progress window	Properties window	Page Setup window
	Properties window	Save As window	Print window
			Replace window
			Save As window

Downloaded from http://www.everyspec.com

8.3 Message Windows

An application displays a message window when it needs to inform users about critical problems, to request clarification of a previous response, or to present non-critical information that requires acknowledgment by users.

GNOME: A message window, shown in figure 8-5, includes a client area where users view and respond to application messages.

Java, Motif, MS Windows: A message window, shown in figures 8-6, includes a window title in the title bar and a client area where users view and respond to application messages.

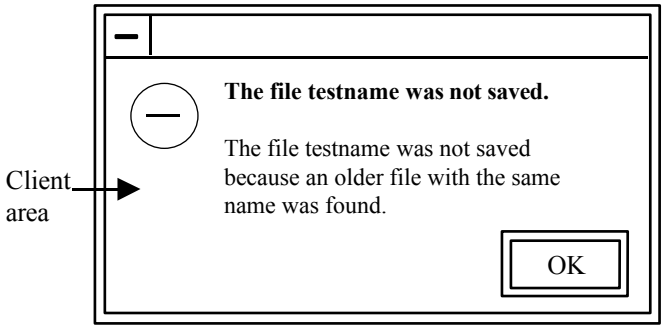


Figure 8-5. Example message window in GNOME.

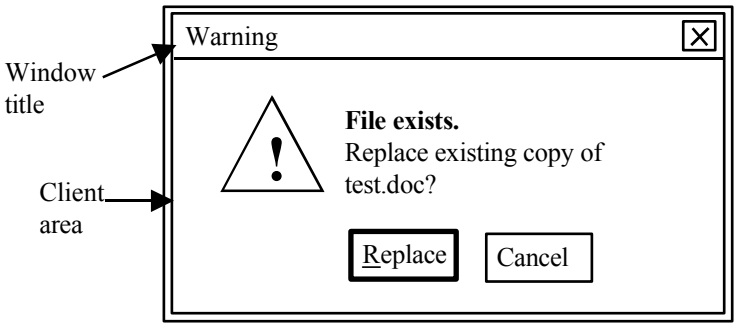


Figure 8-6. Example message window in Java.

8.3.1 Window Titles

GNOME: A message window does not have a title.

Java: The window title describes the purpose of the window, followed by the application name if desired, using the format “Window Purpose – Application Name.”

Downloaded from http://www.everyspec.com

Motif: The window title contains the application name, then describes the purpose of the window, using the format "Application Name : Window Purpose."

MS Windows: The window title contains the name of the object to which the message applies.

8.3.2 Window Content

8.3.2.1 Arrangement of Controls

A message window contains a symbol that identifies the message type, message text, and one or more push buttons, as shown in figure 8-7; the window may also include controls such as a text box or progress indicator.

GNOME: The message text consists of a one-sentence summary, in bold font, of the error or condition that caused the alert, followed by a more detailed description, in normal font, of the problem and recommended action. The top of the message symbol should align with the top of the message text, as shown in figure 8-5.

Java: The message text consists of a short heading, in bold font, that identifies the error or condition that caused the alert, followed by a description, in normal font, of the problem and recommended action, as shown in figure 8-6.

Motif, MS Windows: The message text describes the problem and recommended action.

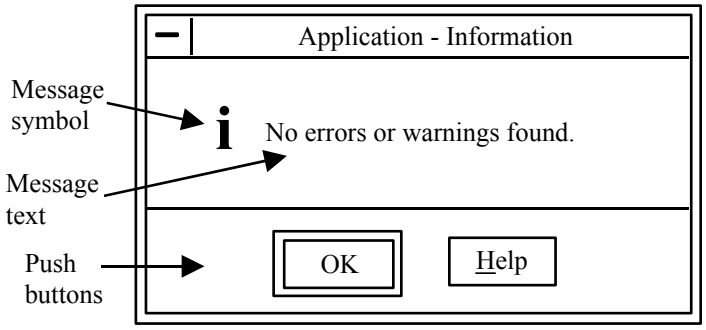


Figure 8-7. Example message window in Motif.

The message text is left-aligned, display-only (i.e., cannot be selected or edited), and positioned to the right of the message symbol in a message window. The following guidelines apply to message text:

- A message should use language that is meaningful to users and require no further documentation or translation.

Downloaded from http://www.everyspec.com

- When a message contains more than one sentence, the important information should be placed at the start of the message. The message should describe the problem first, followed by the action requested.

- If desired, a message can be worded so that the action users are asked to perform can appear as a push button in the window. For example, a window displaying the message "Confirm deletion of file" contains Delete and Cancel push buttons.

A message window can include controls such as a text box or drop-down list box if the window requests information from users. The control and message text are left-aligned and positioned to the right of the message symbol in the window.

Progress bars. A message window can include a progress bar in addition to or in place of the message text in the window. A progress bar, shown in figure 8-8, is a display-only scale that fills dynamically from left to right to indicate that an operation is in progress. A progress bar can be used to show the amount of time remaining, the proportion of a task remaining, or the number of objects remaining. If the progress can be accurately estimated, feedback (e.g., a label indicating the time, percentage, or number remaining) can be included with or in the progress bar, and both the bar and label are updated dynamically every few seconds. The bar does not fill completely until the operation is complete, and the fill never decreases.

GNOME, Java: If the progress cannot be accurately estimated, the progress bar contains a spinning pattern or fills back and forth to indicate an ongoing operation.

The message window containing a progress bar remains displayed until the operation is complete.

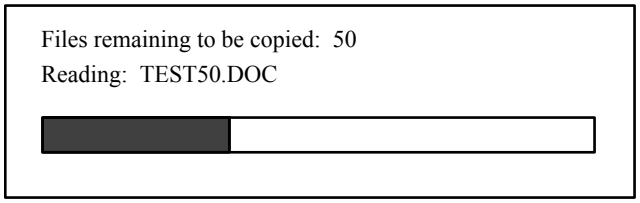


Figure 8-8. Example progress bar.

8.3.2.2 Push Buttons

Push buttons are displayed individually or in groups in a message window. The specifications for push buttons in dialog windows also apply here except as indicated below.

Downloaded from http://www.everyspec.com

GNOME, Motif: Push buttons are arranged horizontally along the bottom edge of a window.

Java: Push buttons are aligned with the left edge of the message text in a window.

MS Windows: Push buttons are centered along the bottom edge of a window.

8.3.3 Types of Message Windows

Table 8-4 lists the pre-built message windows that are provided by each interface for conveying information to users. An application uses these common windows whenever it presents the application messages they support. The style documentation for each interface contains guidelines on the design of common message windows. If desired, an application can extend or customize these windows but should maintain consistency with the basic window design as defined by the interface.

Table 8-4. Common message windows.

GNOME	Java	Motif	MS Windows
Authentication window	Error window	Error window	Critical window
Confirmation window	Information window	Information window	Information window
Error window	Question window	Question window	Warning window
Information window	Warning window	Warning window	
		Working window	

The following guidelines apply to the design of common message windows:

- The message text in an Error window should describe the error, why it happened, and what should be done to correct it. The window may include push buttons or other controls that allow users to correct the error that caused the window to be displayed.

- The message text in an Information window should convey non-critical information that requires acknowledgment (but no further action) by users.

- The message text in a Warning window should explain the possible consequences of an action and allow users to modify or cancel the action. The default push button should be the least destructive action in the window.

- A Working window should be displayed when processing time in response to a user request exceeds 10 seconds or when uses may want to cancel the process in progress.

A message window remains displayed until dismissed by users, except for a progress-type window which closes when processing is done. An application does not present

Downloaded from http://www.everyspec.com

timed-information message windows that close automatically after a fixed period of time without requiring a user response.

An application provides both visual and auditory notifications to users when critical messages are displayed. Users can select the type of notification desired and, when auditory feedback is provided, can set the signals at a very low intensity or disable them as required (e.g., for rig-for-quiet operations on submarines).

GNOME, Motif: An application displays a message window containing critical information in the current workspace even if the application is currently running in a different workspace.

MS Windows: An application displays a message window containing critical information only when an application window is open. If none of its windows is open, an application flashes its taskbar button or places an icon in the taskbar status notification area to alert the user and then displays the message after the user has restored the window.

Downloaded from http://www.everyspec.com

9. User Support Resources

9.1 Context-Sensitive Help

9.1.1 Contextual Help

An application supports access to contextual help using one or more of the techniques described here.

Instant help. Instant help provides help information about the object with focus. F1 activates instant help and displays a pop-up help window at the pointer location, opens a Help dialog window, or launches browser-based application help.

Help mode. Help mode provides help information about an object selected by the user. When Help mode is invoked, the pointer shape changes to a help pointer; clicking BLeft on an object exits the mode and displays a pop-up help window at the pointer location or opens a Help dialog window. SHIFT+F1 invokes help mode from the keyboard.

MS Windows: If a dialog window has a What's This? button, activating the button invokes help mode.

NOTE: If an application supports the common Help menu options as defined by Motif and MS Windows (see table 8-1), help mode can also be invoked by activating the On Item option (in Motif) and the What's This? option (in MS Windows).

Help push buttons. If a window contains a Help push button, it provides help information about the window. Activating the Help push button opens a Help window or launches browser-based application help.

9.1.2 Tooltip Help

GNOME, Java: An application provides tooltip help for all interface components in a window.

Motif, MS Windows: An application provides tooltip help for controls such as toolbar buttons that do not have labels. If desired, an application can provide this form of help for other interface components in a window.

Tooltip help provides help information about an object when the pointer rests on the object. The information is displayed in a pop-up help window at the pointer location.

9.1.3 Help Windows

Help pop-up windows. A help pop-up window, shown in figure 9-1, displays the name of the object about which help was requested and may include a short description of its

Downloaded from http://www.everyspec.com

function. The window provides additional information about the object and should not duplicate the label for the object. If an object is temporarily grayed out or disabled, the window should identify why the object is not available.

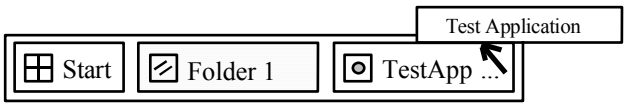


Figure 9-1. Example help pop-up window.

Help dialog windows. A Help dialog window, shown in figure 9-2, describes the object or window about which help was requested. The title of the Help window includes the name of the object or window to which it relates (e.g., "Help on WindowName"). A Help window should be large enough to display at least ten lines of text and wide enough to display its contents without scrolling horizontally. The window contains an OK push button which is the default action in the window. The window may include Previous and Next buttons if the information can be viewed in page increments and a More button if users can access other help information from the window (e.g., launch browser-based application help).

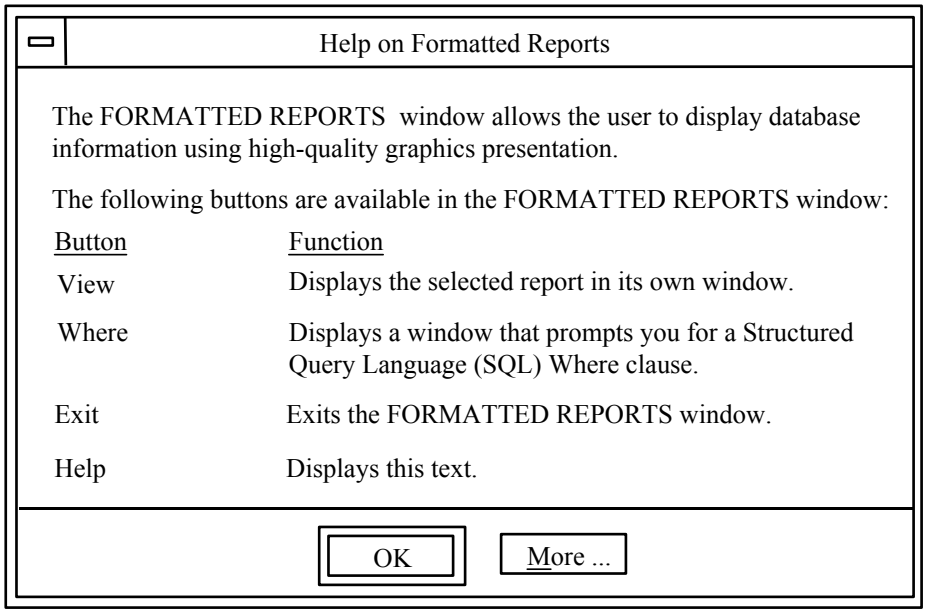


Figure 9-2. Example Help dialog window.

Downloaded from http://www.everyspec.com

9.2 Application Help

An application provides browser-based access to user documentation using online documentation services provided by the COE. Activating application help (e.g., from the <application name> Help option in a Help menu) launches a Web browser configured to display COE documentation. If an application is constructed so its component segments can be loaded or not as the system administrator sees fit, the help information should be similarly subdivided so it is not offered to users unless it is both appropriate and available.

> **NOTE**: If desired, an application can provide access to other user support resources as described in style documentation for the interface (see section 8.1.1.1). These resources can be configured so they use COE online documentation services, or they can be packaged in an application-specific format and made available using COE Web services.

The browser window displaying application help is divided into framed navigation and viewing areas, as shown in figure 9-3. The upper left frame contains a directory of available documents, with entries for each component segment in an application as appropriate. Selecting one of the documents displays its outline or table of contents in the lower left frame. Selecting one of the sections of a document displays the information in that section in the lower right frame, with navigation links provided in the upper right frame. Basic browser features such as bi-directional tracking, history lists, and bookmarks are available for interacting with each document and its content. An application uses the version of HTML supported by the COE and complies with specifications on Web user interfaces in sections 11 and 12.

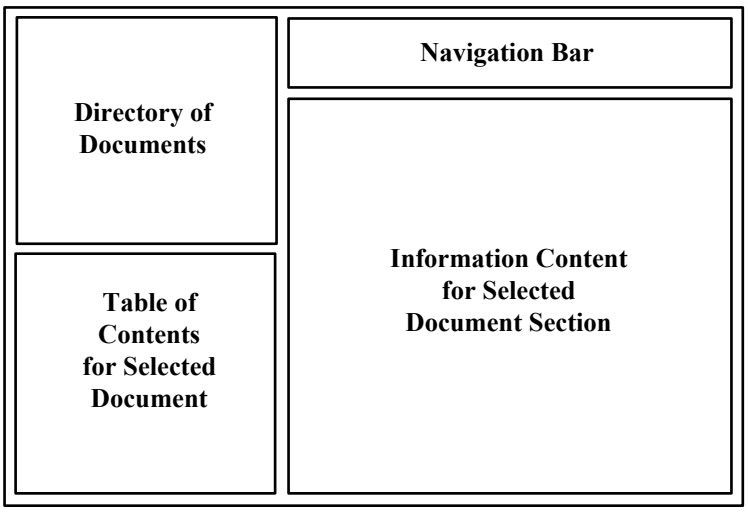


Figure 9-3. Example format for browser-based application help.

Downloaded from http://www.everyspec.com

This page intentionally left blank.

Downloaded from http://www.everyspec.com

10. Application Design and Configuration

10.1 Application Design

10.1.1 Single-Document Interface Applications

Task structure. In a single-document interface (SDI), each primary window in an application displays the content of a document. An SDI application can open a single primary window within which to view one document at a time, or the application can open multiple primary windows, each displaying a single document. If the application is task-oriented rather than document-oriented, it can provide a single primary window within which to perform one or more operational tasks, or it can open separate primary windows for each task, as shown in figure 10-1. Each primary window has its own menu bar, toolbar, and status bar and can parent multiple dialog windows, each of which performs a distinct function in the application. A dialog window may parent one or more additional dialog windows in order to solicit information from users.

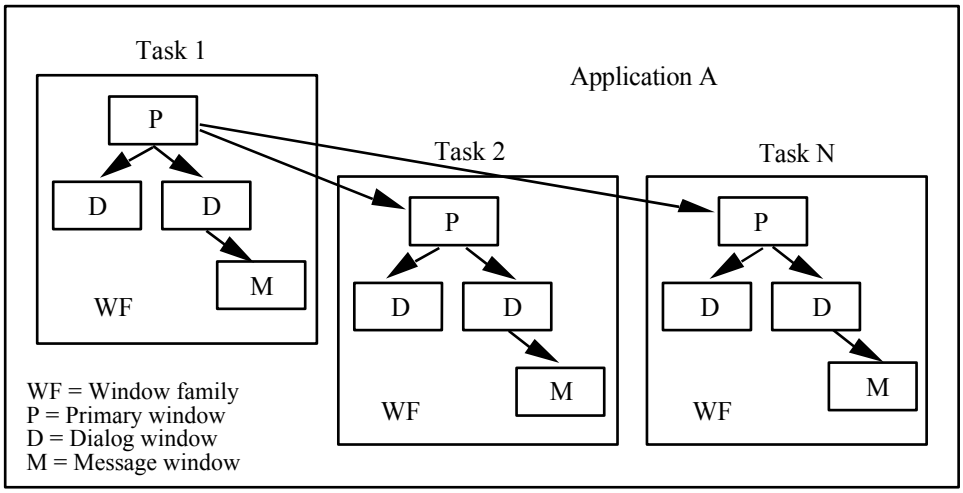


Figure 10-1. Example SDI application.

NOTE: A controlled SDI application is one where the menu bar and toolbar normally included in each primary window are displayed in their own windows. GNOME documentation indicates that this design should be used only when the space within the primary windows in the application is insufficient to support an SDI design.

Data representation. Double clicking on an application icon starts the application and opens a primary window which can be populated with default task data or empty (i.e., awaiting user input). Double clicking on a file icon starts the application, opens a primary window, and populates it with the contents of the file. If an application with a single

Downloaded from http://www.everyspec.com

primary window is already running, double clicking on a file icon closes the currently open file (after saving any unsaved changes) and populates the window with the contents of the new file. If an application with multiple primary windows is already running, double clicking on a file icon opens a new primary window and populates it with the contents of the file.

Window naming. If an application has a single primary window, its title is the application name.

> GNOME: If an application has multiple primary windows, each is identified by file or task name.
>
> Java, Motif, MS Windows: If an application has multiple primary windows, each is identified by application name and file or task name.

Exiting an application. Closing a primary window closes that window; the other primary windows in the application are not affected, and processing by the application continues. Exiting a primary window closes it and all open primary windows in an application and ends processing by the application.

10.1.2 Multiple-Document Interface Applications (MS Windows)[1]

Task structure. In a multiple-document interface (MDI), an application opens multiple documents or files within a single primary window. The contents of files created by the application are displayed in separate document windows that are managed as children of this parent. These windows are constrained to appear within the parent and share the functionality of its menu bar, toolbar, and status bar. Secondary windows are available as children of the parent application window and have the same appearance and behavior as in an SDI application.

Data representation. Double clicking on an application icon starts the application and opens the parent window; a default document window may open inside the parent, or the parent may be empty. Double clicking on a file icon starts the application, opens the parent window, and displays a document window populated with the contents of the file within it. If an application is running and the parent window is already open, double clicking on a file icon opens a new document window in the existing parent window.

Window naming. The title of a parent window is the application name. The title of a document window is the file name. If a document window is maximized, the title of the parent window is the application name, followed by the file name.

Exiting an application. Closing a document window closes that window; the other document windows in the application are not affected, and processing by the application

[1] GNOME and Java recommend that an application not have an MDI design because of the usability problems it poses. Motif does not support an MDI design.

Downloaded from http://www.everyspec.com

continues. Exiting the parent window closes it and all child document windows in an application and ends processing by the application.

10.1.3 Nested Applications[2]

Task structure. The software for performing the tasks in an application usually comes from a single source (i.e., one software segment). In the COE, the software in an application can be contributed by different sources. The result is a nested implementation where the window families in an application are "owned" by different segments, as shown in figure 10-2. An application is composed of a parent segment and one or more other segments that managed as children of the parent segment. For example, in figure 10-2, Segment X supports tasks 1 and 2 and serves as the parent for the application, while Segment Y supports task N and is available from within Segment X. A child segment can contribute an entire window family, as shown in figure 10-2, or it can add dialog windows to an existing window family in the parent segment.

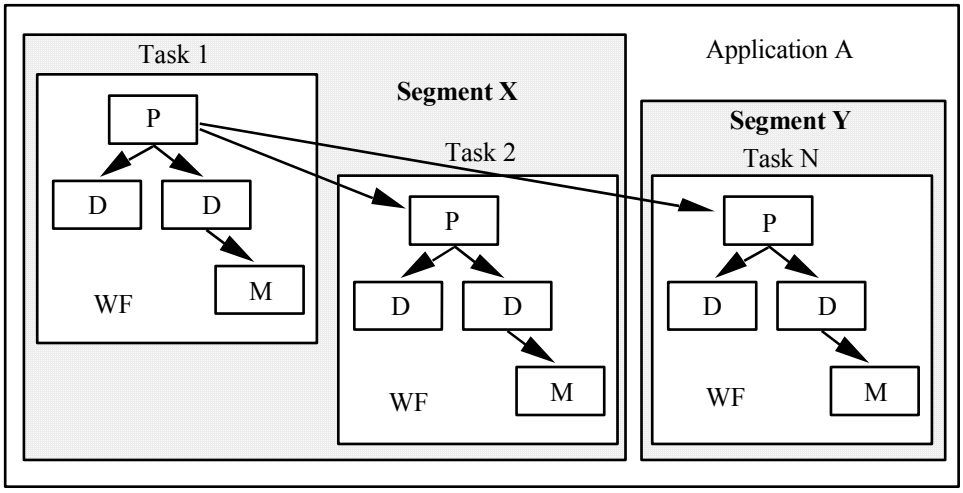


Figure 10-2. Example nested application.

Data representation. Double clicking on an application icon starts the parent segment and opens the primary window for the segment. The child segments in a nested application are accessed from the primary window(s) in the parent segment. The tasks can occupy all or part of a menu in a primary window, be available in a submenu, or be distributed across several menus, as shown in figure 10-3. If menus become excessively long (e.g., extend beyond the bottom of the screen), an application can reduce menu length by organizing options into submenus, including controls (e.g., arrow buttons) for scrolling the options that cannot be viewed, or including an option (e.g., in the View menu) for users to show/hide sets of options related to specific tasks they perform in the window.

[2] A nested application is a COE-specific design and not addressed in the style documentation provided by each interface.

Downloaded from http://www.everyspec.com

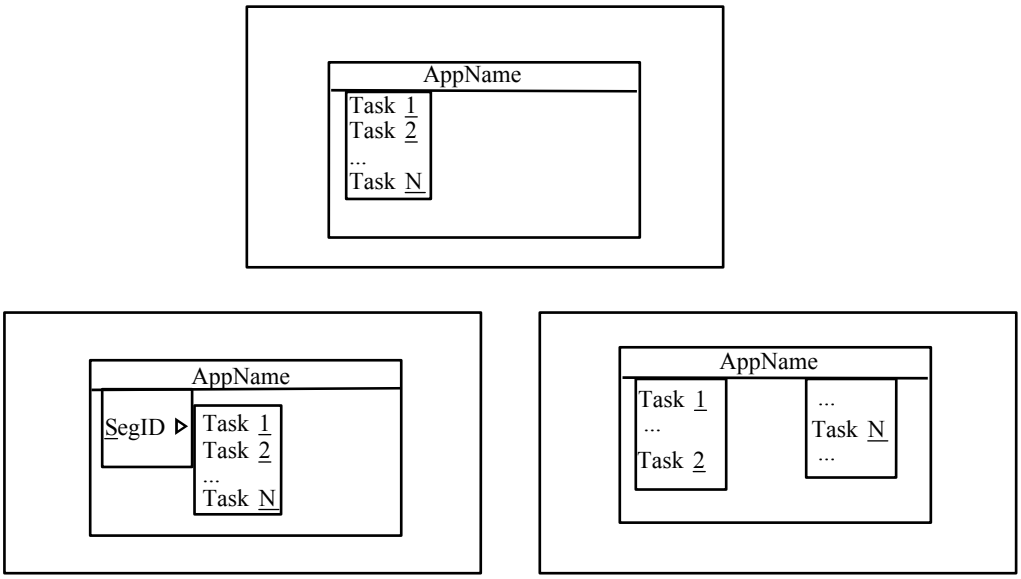


Figure 10-3. Options for accessing child segments in a nested application.

Window naming. The title of a primary window in the parent segment is the segment name. In a child segment, window titles match or refer to the wording of the menu option that displayed the window.

Exiting an application. Closing a primary window in a parent or child segment closes that window. Exiting a primary window in the parent segment closes all windows in the application and ends processing by the parent and all child segments.

Downloaded from http://www.everyspec.com

10.2 Application Configuration

10.2.1 Visual Design

10.2.1.1 Icons

An application complies with interface-specific requirements concerning the size and format of icons supplied by the application. Guidelines on icon design are provided in the style documentation for each interface. An application complies with interface-specific conventions related to icon design. An application uses the graphics for file and folder icons provided by the interface and ensures that application-specific icon images are rendered legibly in all required sizes and formats.

GNOME: An application supplies icons in multiple sizes, rendered in table and shelf perspectives and with an upper left and overhead light source, as indicated in table 10-1. Table perspective presents objects as if they were sitting on a table in front of the user; shelf perspective presents objects as if they were placed at eye level.

Table 10-1. Icon requirements in GNOME.

Icon Type	Size (in Pixels)	Perspective	Light Source
Object / Document Icons	24x24 48x48* 96x96	Table	Upper Left
Application Icons	24x24 48x48*	Table	Upper Left
Toolbar Icons	24x24* 48x48	Shelf	Overhead
Menu Icons	16x16 * Primary size	Shelf	Overhead

Java: An application supplies icon images in accordance with the size requirements of the platforms on which the application will be installed. Application icons are normally available in two sizes (16 x 16 pixels and 32 x 32 pixels), and toolbar icons in two sizes (16 x 16 pixels and 24 x 24 pixels).

Motif: An application supplies icon images in three sizes (16x16 pixels, 32x32 pixels, and 48x48 pixels) and in bitmap and pixmap formats (i.e., .xbm and .xpm), as indicated in table 10-2. Toolbar icons are available in 24 x 24 pixels.

Downloaded from http://www.everyspec.com

Table 10-2. Icon requirements in Motif.

Icon Type	Color 16x16	Color 32x32	Color 48x48	Mono 16x16	Mono 32x32	Mono 48x48
Application Icon	Required	Required	Required	Required	Required	Required
Document or File Icon	Required	Required	-----	Required	Required	-----
Folder or Container Icon	Required	Required	-----	Required	Required	-----
Minimized Window Icon	-----	-----	Required	-----	-----	Required

MS Windows: An application supplies icon images in three sizes (16x16 pixels, 32x32 pixels, and 48x48 pixels), in bitmap or icon file format (i.e., .bmp or .ico), and in 16- and 256-color versions. Toolbar icons are available in 16 x 16 pixels, 16- and 256-colors, and 20 x 20 pixels, 256 colors.

10.2.1.2 Layout and Visual Alignment

Section 8 provides specifications on the design of primary, dialog, and message windows, and section 13 addresses the presentation of text and graphic information. Detailed guidelines concerning the layout, visual alignment, and size of interface components in windows are available in the style documentation for each interface. An application complies with these interface-specific conventions related to window layout and visual alignment.

10.2.1.3 Capitalization

Window text can be displayed using either header-style capitalization (i.e., capitalize the first letter of each word, except for prepositions and articles) or sentence-style capitalization (i.e., capitalize the first letter of the first word and any other words normally capitalized in sentences). Table 10-3 identifies the capitalization style used in interface components and labels. An application complies with interface-specific requirements related to capitalization of window text.

Downloaded from http://www.everyspec.com

Table 10-3. Capitalization of interface components and labels.

	GNOME	Java	Motif	MS Windows
Check box labels	Sentence	Header	Header	Sentence
Combo box items	Sentence	-----	Header	Sentence
Combo box labels	Sentence	-----	Header	Sentence
Drop-down combo box items	Sentence	Header	Header	Sentence
Drop-down combo box labels	Sentence	Header	Header	Sentence
Drop-down list box items	Sentence	Header	Header	Sentence
Drop-down list box labels	Sentence	Header	Header	Sentence
Group box labels	-----	Header	Header	Sentence
Headings	Header	Header	Header	Sentence
List box column headings	Header	Header	Header	Header
List box items	Sentence	Header	Header	Sentence
List box labels	Sentence	Header	Header	Sentence
Message text	Sentence	Sentence	Sentence	Sentence
Option menu options	Sentence	-----	Header	-----
Option menu labels	Sentence	-----	Header	-----
Pop-up menu options	Header	Header	Header	Header
Pop-up menu titles	-----	-----	Header	-----
Pull-down menu options	Header	Header	Header	Header
Pull-down menu titles	Header	Header	Header	Header
Push button labels	Header	Header	Header	Header
Radio button labels	Sentence	Header	Header	Sentence
Scale labels	Sentence	Header	Header	Sentence
Spin box labels	Sentence	-----	Header	Sentence
Status bar texts	Sentence	Sentence	Sentence	Sentence
Tab labels	Header	Header	Header	Header
Text box labels	Sentence	Header	Header	Sentence
Toolbar button labels	Header	Header	Header	Header
Tooltips	Sentence	Header (if short); Sentence (if long)	-----	Header
Window titles	Header	Header	Header	Header

10.2.2 Style Management

10.2.2.1 Font and Color Attributes

The desktop provides a centralized style management capability that changes font and color attributes of user interface components dynamically when users select a different font or color scheme. The desktop may manage these attributes individually (e.g., in separate Color and Font dialog windows) or group them together in “themes” that define the visual appearance of interface components.

An application inherits the visual features of interface components from system-level settings and does not hardcode any of the attributes for these components. In addition, an application ensures that any application-specific coding related to these attributes is legible regardless of the font or color scheme selected by users. Finally, an application

Downloaded from http://www.everyspec.com

ensures that window size and layout respond correctly to changes in font attributes (e.g., an increase in font size).

If an application requires control over font attributes (e.g., for text in graphs or charts or on tactical displays), it uses a sans serif font of sufficient thickness and size to be readable when users are seated at a normal viewing distance from the screen. At a minimum, character height should be 1/200 of viewing distance (e.g., a viewing distance of 36 inches requires a .18 inch character height on the screen). If desired, an application can use a bold font to ensure that the text is readable when presented normally or grayed out even if screen resolution is degraded. An application should not use the italic or slanted style of a font since this style can be more difficult to read than the same font presented in its normal style.

> **NOTE**: MS Windows style documentation recommends that the text in windows be displayed in non-bold font and that bolding be used only in titles and headings and to call user attention to an area. The desired implementation is as described above.

If an application requires control over color attributes, it selects color values based on and consistent with the color model used by the interface. In general, a color scheme with a light background and black text is recommended, assuming an office-like operational environment with normal ambient lighting.

Additional specifications on the use of font and color in the presentation of text and graphic information are provided in section 13.

10.2.2.2 Input Focus

The desktop manages input focus policy and may allow users to select either explicit or implicit focus. An application complies with the system-level focus policy and ensures that any application-unique features behave correctly under the focus policy currently in effect.

10.2.2.3 Pointing Device Features

The desktop provides a centralized style management capability that allows users to customize pointing device features (e.g., reverse the actions assigned to BLeft and BRight, set the double-click speed, adjust the speed of pointer movement). An application ensures that any application-specific pointer shapes or pointer actions do not conflict with the system-level pointing device settings currently in effect.

Downloaded from http://www.everyspec.com

11. Web Page Design

11.1 Page Elements

11.1.1 Page Titles

A Web page has a title that describes the content of the page and is displayed in the title bar of the browser window. Each page on a Web site has a different title so users can identify the page if they navigate to it from outside the site or if they include the page in a Bookmarks or History list maintained by the browser. A page title should contain four to six words (40-60 characters), with keywords about the page content placed at the start of the title, so that an alphabetized list of titles is ordered by keyword and can be scanned easily by users.

11.1.2 Page Headers

A Web page includes a header with identifying information about the page and a set of links for navigation. The header displays the page name, a logo or "signature" graphic that identifies the site, and text and/or graphics links (e.g., a navigation bar, page path) that allows users to understand how the page is related to other pages within the Web site. The page name is the same as the page title.

The header, which is displayed on every page, is sized and its content arranged so it is visible within the browser window on a standard-size screen without scrolling horizontally. If desired, the logo in the header can serve as a link to the home page for the site so this page can be accessed directly from anywhere in the site. The page header can also include a search button so this capability is available from every page in the site.

11.1.3 Page Footers

A Web page includes a footer that identifies the page author, contact information, and page creation and/or revision dates, and a set of links for navigation. If desired, the author's name can be a link to a biography page with information about the author, including an e-mail address. Alternatively, the contact information can be a "mailto" link that allows users to send mail to the author. The actual address (rather than some other text) should be included in the link so users with browsers that do not support a "mailto" capability can copy the address for use in a separate mail application.

11.1.4 Links

A Web page provides consistent visual cues that indicate the presence of a link. A text link is underlined and changes color to indicate when a destination has been visited. Links to unvisited pages are blue, and links to visited pages are purple or red. A Web page uses these colors to support effective navigation by users. If users select a link and there are other links to the same destination (either on the same page or other pages), all of the links change color to indicate that the destination has been visited.

Downloaded from http://www.everyspec.com

Links can also be graphic or included in drop-down lists, or they can be made visible using mouse-over feedback (which changes the appearance of the text defined as a link when users move the pointer over it). However, underlined text is the preferred approach for identifying links because the text is visible to users (drop-down lists and mouse-over links hide content), and it changes color once the link is visited (graphic links do not).

A text link consists of a single word or short phrase (i.e., two to four words) that describes the destination so users can decide if they want to follow the link. If a graphic is included with the text in a link, the graphic is defined as part of the link and has the same destination as the text. The following guidelines apply to links:

- Links that are very similar to each other should include supplementary text (next to but not part of the link) that allows users to distinguish between the links and determine which one to select.

- A link with unexpected consequences (e.g., a very large file, an unusual file format) should be identified as such (e.g., in a parenthetical note or a link title) and include its size (especially if very large) and whether it is internal or external to the site (since there may be differences in loading delay and return path).

- Each link to a given destination should use the same URL when referring to the destination. Doing so ensures that the color coding assigned to visited and unvisited links to the destination remains consistent (e.g., when a destination is visited, all of the links to that destination change color).

11.1.4.1 Embedded Links

Embedded links are displayed in the body text of a Web page and provide access to parenthetical or supplementary information about a topic. When users select an embedded link, the action executed is to navigate within the same page, to a different page within the same site, or to a page at a different site. Because users expect that links will navigate to a different page within the same site, the use of within-page links should be minimized, and links that take users to a different site should be labeled as such.

Selecting an embedded link can replace the current page with the destination page in the same browser window, or it can display the page in a new browser window that opens in front of the current window. In the former case, users can select the Back button in the browser window to return to the previous page or site; in the latter case, users can view the destination page without leaving the original site.

11.1.4.2 Structural Links

Structural links provide access to the key parts of the Web site and allow users to move among them. While the specific links included on a particular Web page may vary, the same set of structural links to key parts of the site are available on all pages. Structural

Downloaded from http://www.everyspec.com

links appear in the same order on each page and can take the form of a button bar or set of tabs at the top of the page (e.g., in the page header), a left-side navigation bar in a pane or panel in the page body, or a set of text links (e.g., in the page footer). At a minimum, each page provides a text-only version of structural links so users with a graphics-disabled or text-only browser have access to the links.

11.1.4.3 Link Titles

A link title is a small pop-up window that is displayed when users rest the pointer on a link. The window provides supplementary information about a link such as the name of the destination (if it is different from the current site), a short description of the information found at the destination, and any alerts about the destination (e.g., password required). This information is included in the "title" attribute of the link anchor tag and is 60-80 characters in length. A link title should be provided whenever a link has unexpected consequences. The link titles included on a Web page are visible only if the browser understands how to display this information.

11.1.4.4 Page Paths

A page path is a navigation aid that identifies the sequence of pages that lead to the current page from the home page. Except for the current page, each name in the path is defined as a link that navigates to that page. For example, "Home Page > Menu Page > Content Page" defines the path from the home page, to a menu page, and then to the current content page, with the first two pages identified as links to those destinations. If a page path is included on a page, the page names in the path should be separated by arrows and shortened as needed so the path fits on a single line.

Downloaded from http://www.everyspec.com

11.2 Page Layout

11.2.1 Design Templates

A Web page is designed according to a design template that identifies common page elements and how they are to be displayed on the page. This template, which is applied to all of the pages on a site, includes a layout grid, such as the ones shown in figure 11-1, that specifies the placement of common page elements and indicates the number and location of major navigation links. The template also defines style features such as logos, fonts, color schemes, page backgrounds, and size and position of text and graphic elements and may include standard terminology to use in headings, titles, and navigation elements throughout the site. If desired, a Web site that is very large and complex can create several design grids tailored to address specific content within different parts of the site.

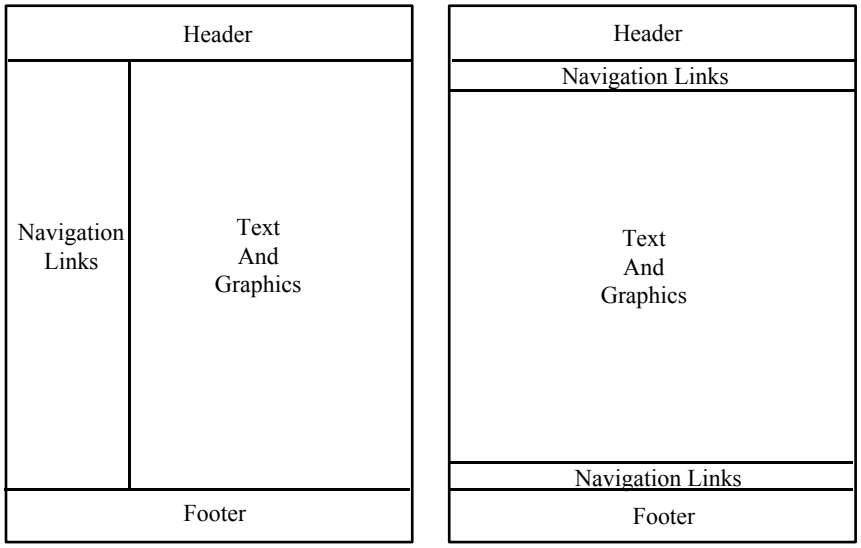


Figure 11-1. Example layout grids for Web pages.

11.2.2 Layout Tables

"Invisible" tables can be used to define the main areas of a Web page and to control page layout. A layout table is made "invisible" by setting the value of the "border" attribute in the table tag to 0. A table can be used to define a multicolumn layout for the body of the page, as in the example on the left in figure 11-1. Margins are included in the table to separate the areas so the content can be read and/or scanned easily. The empty space in a margin can be created by adding a cell to the table that functions as the margin or setting values for the "cell padding" or "cell spacing" attributes of the table tag.

An "invisible" table can also be used to position images on a page. Because links can be included in table cells, it is possible to create a composite image map by placing a portion

Downloaded from http://www.everyspec.com

of the image in each cell of the table. When displayed by a browser, the table presents the various images seamlessly and makes the composite behave like an image map (see section 11.3.2.1). If a layout table is used in this way, the "border," "cell padding," and "cell spacing" attributes of the table tag are set to 0 so the portions of the image in each cell join correctly.

A layout table can be either fixed-width or variable-width. In the former, the table dimensions are fixed regardless of the size of the browser window; in the latter, the table dimensions adapt to fill the browser window.

A fixed-width table can be created by specifying an absolute value as the width of each cell. While this type of table provides control over the position of page components and line length and spacing, it can result in large amounts of blank space if the table width is significantly smaller than the browser window size. In addition, if the content of a fixed-width table is printed, the table width may not fit the printable area of the paper used. The amount of blank space can be minimized by centering the table in the browser window, and printing problems can be mitigated by providing a separate file containing a printable version of the page content with margin widths that fit the size of the paper used.

A variable-width table can be created by specifying cell widths as percentages. When the size of the browser window changes, the cells resize but maintain their relative proportions. A variable-width table can also be created by allowing table cells to size themselves according to the size of the browser window.

If desired, a layout table can contain both fixed- and variable-width cells. For example, fixed cell widths can be used for columns that require absolute positioning (e.g., the navigation column in the example on the left in figure 11-1), and unspecified cell widths for the remaining column (e.g., text and graphics) so it adapts to fill the browser window.

11.2.3 Frames

Frames can also be used to define the main areas of a Web page and to control page layout. While frames provide another option for standardizing page layout, they can reduce the available space within the browser window, increase download time, and make a Web site more difficult to view and navigate. As a result, a Web page should use frames only when other layout options are inadequate or inappropriate. If frames are used to define page layout, the following guidelines apply:

- The number of frames and their size and placement should be based on the type of information being presented, with the same overall layout used throughout a site. An effective layout is one where users can view information without resizing the frame and can read headings, select links, and view images without scrolling frame contents repeatedly. By default, users can resize a frame, with the browser adding scrollbars only when the contents are too large to fit within the framed area. These defaults should be used when specifying if a frame can be resized and scroll bars are visible.

Downloaded from http://www.everyspec.com

- As with layout tables, frames can be either fixed-width or variable-width, depending on whether frame size is specified as an absolute or percentage value. A variable-width frame is preferred because it adjusts to changes in the size of the browser window.

- One of the disadvantages in using frames is that navigation actions such as bookmarking no longer work correctly. Nielsen in Designing Web Usability recommends that this problem can be fixed by including a "target=top" attribute in the anchor tag for each link, which forces all of the frames to be reloaded and provides the browser with a new URL for the destination.

- Because browsers differ in the extent to which they support frames and some users may choose to disable frames, a Web page provides a "no frames" alternative that displays page content without frames enabled.

11.2.4 Cascading Style Sheets

Cascading style sheets (CSS) can be used to control the presentation and placement of text and graphic information on a Web page. The style features of a Web site are described in a single style sheet that is applied to all of the pages in the site. If the site supports the needs of very different audiences, separate style sheets can be defined that are tailored to each audience. A linked style sheet (i.e., one that is maintained in a separate file) is preferred to an embedded style sheet (i.e., one that is part of the Web page) because a linked style sheet provides a centralized location for managing style changes and reduces the size (and download time) of individual Web pages by removing the style-related code.

A Web page that relies on CSS is able to present its contents legibly if the style sheet is disabled (e.g., by the user or the user's browser). This flexibility is required in order to support users who have older browsers or who choose to disable the style feature in their browser. In addition, because browsers differ in how they interpret style sheets, Web pages are viewed on different browsers to determine the extent of variability in how the pages are displayed, and style sheets are modified as needed to maximize consistency in appearance across browsers.

Downloaded from http://www.everyspec.com

11.3 Page Content

11.3.1 Text

11.3.1.1 Font Attributes

The font attributes used to display text on a Web page can be specified via CSS or inherited from browser-level settings. If style sheets are used, page layout should be defined with a variable-width table so page content adjusts correctly to changes in attributes such as font size.

Font typeface. The number of fonts on a page should be limited to two, one typeface for body text and one for headings, with the fonts selected from the same typographic family. A serif font such as Times New Roman or a sans serif font such as Arial is preferred. If the font size is very small (e.g., 9 point or less), a sans serif font should be used because it is easier to read. When a typeface is specified, several alternative fonts are included to maximize the likelihood that one of them is supported by the browser.

Font style. Regular font style is used in the body text on a Web page; bold style is reserved for text in headings and to focus user attention on important information. Text is not underlined (since underlining identifies a link), and entire sentences or blocks of text are not presented in italics (since this style in some fonts can be difficult to read when displayed by a browser).

Font size. Font size is defined as a percentage, rather than an absolute, value so it scales correctly relative to the base size defined in browser-level settings. To be legible, text needs to be displayed in at least a 10-point font, with a larger font size (12-point or 14-point) desirable under certain viewing conditions.

Font color. Black text on a white or very light gray background is recommended for maximum contrast and legibility when viewed in office-like environments with normal ambient lighting. White text on a black background also provides high contrast but can reduce users' reading speed (compared to black text on a white background).

11.3.1.2 Grammar and Editorial Style

The body text on a Web page is presented in mixed case, following standard rules for capitalization and punctuation. The content is written so it can stand alone if viewed independently of other pages on the Web site. For example, a page does not start with references to "the previous step" or "the next step." Text is worded so it is readable (and can be printed, if desired) despite the presence of links on the page. The following guidelines apply to editorial style:

- Each page should include opening information (e.g., in the first paragraph) that provides sufficient context for users to understand the content of the page and

Downloaded from http://www.everyspec.com

determine if the page contains what they are looking for, or the page includes a link to background information that provides this context.

- Each paragraph on a page should address a single idea or topic, with the first sentence containing the key information about the topic.

- The text should not contain references to online aspects of the content or to protocols or mechanisms used to find the information. For example, "Click here for information on XXX" is better stated as "See the section on XXX." Similarly, "You can read more about XXX in the tutorial that is linked to the home page" is better stated as "The tutorial contains more information on this topic."

The text used in a link describes what it is about (so it makes sense when read out of context) and does not consist solely of the word "here." Because a browser may include the text as a bookmark or hit list entry, using "here" as a link results in entries that contain only this word. Whenever possible, links are placed at the end of a sentence or paragraph so users can read the text before deciding if they want to follow the link.

11.3.1.3 Alignment and Spacing

The titles, headings, and body text on a Web page are left-aligned to increase readability, and white space is used to define the major areas of the page and to separate paragraphs. If separators are used to delimit different areas of a page, they are created as a tag, rather than as an image, so they can be displayed by all types of browsers. When delimiting the content on a page, white space is preferred over using a separator because it often looks better and downloads faster. If desired, the first line of a paragraph can be indented by setting the appropriate paragraph property in CSS or, if a style sheet is not used, by inserting several space characters or single-pixel transparent Graphics Interchange Format (GIF) graphics as spacers at the start of the paragraph. The approach taken to defining spacing and paragraph style is included in the design template that defines page layout and is applied consistently throughout the site.

Text is arranged in a single column on a Web page; text is not presented in side-by-side columns, which force users to scroll down and then back up the page to continue reading. To increase text readability, line length should be about 10-12 words per line. Lynch and Horton in Web Style Guide indicate that in fixed-width page layouts created using an "invisible" table, text can be limited to this length by making text cells no wider than 365 pixels. In variable-width page layouts where cell width is not fixed, the authors recommend using CSS to increase the vertical spacing in a block of text, which improves legibility when line length is longer.

11.3.1.4 Headings

A Web page uses headings to break up large blocks of text and make each section of the page stand out according to subject. The same heading tag is applied to the same level of content on all pages. Each heading should be worded so it is meaningful when it is

Downloaded from http://www.everyspec.com

viewed in isolation (e.g., when the heading appears as an item in a menu or "hit" list). The wording should uniquely identify the content that follows the heading so search engines that examine the keywords in heading text can determine if the page is a "hit" that matches the search parameters. CSS can be used to specify the typographic attributes of heading text so it has the same appearance on all pages that reference the style sheet.

11.3.2 Images and Multimedia

11.3.2.1 Images and Image Maps

An image is left-aligned or centered on a Web page and has a transparent background so it is the same color as the underlying page. Horizontal and vertical space is included around the image so it does not abut the text that is next to it. Whenever possible, the image should be sized so it can be viewed without scrolling and printed on standard-size paper. If desired, the height and width of the image can be specified in the image tag to speed up the rendering process as the image is downloaded.

The images on a Web page are usually provided in either GIF or Joint Photographic Experts Group (JPEG) file format. GIF is preferred for graphic images such as drawings, drawings, icons, and buttons, while JPEG is preferred for photographs and complex images. If a GIF image is used, it should be made available in interlaced, rather than non-interlaced, form. An interlaced GIF image becomes progressively clearer as the file is transferred to the browser, while a non-interlaced image is loaded from the top of the picture to the bottom. While both formats take the same amount of time to download, an interlaced one gives more control to users by allowing them to identify the image sooner and determine whether they want to view the entire image or stop it before the transfer is complete. The progressive form of a JPEG image provides a similar advantage compared to the standard form of this image.

If desired, "invisible" images can be used to arrange the elements of a page and provide consistent formating of layout tables in different browsers. For example, single-pixel transparent GIF images can be included as spacers in the table cells of layout tables so the cells do not appear empty to the browser.

An image map is an image that is divided into multiple regions, each of which is defined as a link that navigates to a different destination. Each region should have clearly marked boundaries and include a descriptive label that helps users identify the presence of a link. An image map is interlaced so users can select a link as soon as they recognize the region they want to select. If an image map is included on a home or menu page, it is sized so it downloads quickly (since these pages are likely to be visited frequently). While an image map can enhance the appearance of a page, it provides little information about the content of the destination and is a poor information look-up tool as far as speed and detail.

A short text description is included as part of the image tag for all images on a Web page except those that are purely decorative so users with a text-only or graphics-disabled

Downloaded from http://www.everyspec.com

browser are informed about what cannot be displayed. Alternative text is also provided for each link destination defined in an image map. The text consists of eight to ten words presented in mixed case, following standard capitalization and punctuation rules.

11.3.2.2 Animation and Multimedia

Animation can be included on a Web page to perform functions such as illustrating a concept or technique, showing changes over time, or alerting users to updated information. Most animation requires a special plug-in for viewing. However, animation can be simulated by loading a sequence of layered GIF images; this type of animation is handled by the browser itself and does not require a plug-in. Because animation effects such as marquees (i.e., words and phrases that scroll across a page) slow down page loading and can be distracting and difficult to read, they are used only when integral to the content of a Web page.

If a multimedia object such as an audio or video clip is incorporated into a Web site, it is provided as a supplement to the text and images on a page. Because these files can be large, they are not part of the basic page design but instead appear in a separate page that users can close when they finish with the content. If a multimedia object is made available on a Web page, the following guidelines apply:

- The page should describe the object and its content, including previews such as still shots from a video, and specify the run time. The page should also indicate the file size and download time and identify any required plug-ins along with a link to download them.

- The multimedia object should have defined beginning and end points (rather than looping continuously). Users should be able to play and stop an animated GIF file and to control playback of audio and video objects, including play, pause, and stop and volume adjustment as appropriate.

11.3.3 Applets

An applet can run in a browser window, or it can create a window that is separate from the browser. If an applet is tightly integrated with the content of a Web page, the applet is displayed in the browser window and designed in accordance with the specifications for Web interfaces presented here. If an applet performs a function or manipulates data separate from the Web page, the applet is displayed in its own window. The window does not have any browser controls, and the applet is designed in accordance with the specifications for graphical interfaces contained in this document.

Downloaded from http://www.everyspec.com

11.4 Page Design Considerations

11.4.1 Page Dimensions

Page width. The width of a Web page is designed to adapt to whatever screen size it is viewed on and to support printing directly from the browser. Users can be expected to have a 17- to 19-inch monitor set to display an 800 x 600 pixel screen area, although some systems may have a smaller monitor with less display area. A Web page accommodates various display sizes and resolutions by minimizing the amount of horizontal scrolling required to view page content. In addition, because a printed page is not as wide as most monitors, a Web page is arranged so its content can be printed without cropping words or using small type. Lynch and Horton in Web Style Guide provide the following recommendations concerning page dimensions assuming an 800 x 600 pixel screen area:

	Layout for Maximizing Screen Use	Layout for Proper Printing
Maximum width	760 pixels	560 pixels
Maximum height (visible without scrolling)	410 pixels	410 pixels

Page length. The length of a Web page depends on the nature and complexity of the content and the manner in which the information is expected to be used. While long pages require scrolling and can be disorienting to users, they collect related information on a single page and are often easier to download and print. It is recommended that home and menu pages and pages that need to be browsed and/or read online be limited to one or two screens of content and that all other pages contain no more than five screens of content.

Because a Web page is viewed as a sequence of vertical “chunks” of information, the first screen of content that users see should be the most information-rich. A Web page is designed to display the most important content and the highest density of navigation links at the top of the page, with less important content and lower priority links visible on subsequent screens as users scroll down the page.

11.4.2 Page Size

A Web page is sized to accommodate the bandwidth available to most users. Users are normally willing to wait about 10-15 seconds for a Web page of average size to download. Nielsen in Designing Web Usability provides the information in table 11-1 on the maximum allowable page size needed to achieve response times for various connection speeds. Per Nielsen, response times should be as fast as possible, with a ten-second response being the limit for maintaining user attention on navigating within a site. In addition, users should be informed whenever select a link to a page that is larger than the norm for pages at the site given the bandwidth available to most users.

Downloaded from http://www.everyspec.com

Table 11-1. Maximum allowable page size for response times at different connection speeds.

	One-Second Response Time	Ten-Second Response Time
Modem	2 kilobytes	34 kilobytes
ISDN	8 kilobytes	150 kilobytes
T1	100 kilobytes	2 megabytes

11.4.3 Device-Independent Page Design

While most users can be expected to have a graphics- and color-capable browser, some users may be limited to an all-text browser or choose to deactivate graphics or disable the style feature in their browser. In addition, users are likely to view page content on platforms with display devices that vary in size and resolution. A Web page accommodates different browsers by providing redundant access to graphic information. For example, text links are included on each page along with graphic-based button bars to support navigation, and text equivalents are available for those graphics that provide information required to understand the content of a page or to facilitate navigation.

A Web page anticipates the ways in which information will be used by making critical content available in multiple formats (e.g., in word processing format for downloading, in a single file for printing) and by providing information about and access to special plug-ins required to interact with page content. A Web page adapts to differences in screen size and resolution by specifying the page layout as percentages of available space and not using fixed-width values for any tables, frames, or other page elements. In addition, text attributes such as font size are defined in relative, not absolute, terms so users can resize text larger or smaller to accommodate the size and resolution of their monitors and optimize overall viewing and/or reading conditions.

11.4.4 Accessible Page Design

While many of the requirements for accessible Web page design[1] also increase usability for non-disabled users and users with text-only or graphics-disabled browsers, some are targeted to the specific needs of disabled users. If compliance with accessibility standards is required, the following guidelines apply:

- A Web page should not use a complex layout containing multiple nested tables. Accessibility software reads tables in a linearized manner, with the content of each cell read in sequence. A complex layout presents the content of a complex table in what seems to be random order and reduces the comprehensibility of the page. Layout tables should be kept simple, or CSS should be used instead of tables to create

[1] Appendix B describes federal standards for information accessibility by individuals with disabilities.

Downloaded from http://www.everyspec.com

columns and control spacing. If a page requires a complex layout, it should include a link to an alternative page that presents the same content in a more accessible format. In addition, if empty cells are used to define margins in a layout table or if transparent GIF graphics are used for spacing in page layout, a blank "alt" statement should be included in the appropriate tag so the cell and graphic are not read by accessibility software.

- A Web page that uses frames should include a meaningful title and name for each frame in the frame set.

- Users should be able to change text attributes such as font size and color. Font size should be defined in relative terms so users can display text in a large font, and text should be rendered as plain text and not in graphic form, given that images do not respond to changes in size and color in the same manner as plain text. CSS and a flexible page layout (e.g., a variable-width layout table) should be used so the page layout adjusts correctly when users apply their own style sheet to enlarge the font. Color should be used as a redundant code (e.g., with text bolding), with users able to select different foreground and background colors in order to maximize contrast.

- Alternate text should be provided for all graphic links on a Web page. For example, if the link is an image in a button bar, the text describes the purpose of the link and does not include phrases such as "click here" or "link to." If the same set of navigation links is repeated on each Web page, a "skip to main content" link is provided at the top of the page that skips these links and moves to the top of the content section of the page. Alternate text should also be provided for all images and image maps that convey information on a Web page. Longer descriptive text should be available that explains the content of images such as charts and graphs. If the page includes decorative images such as custom bullets or uses transparent GIF images for spacing, the alternative text in the image tag should be empty so the graphic is not read by assistive software.

- Marquees and special effects that blink, fade, or move should not be included on a Web page. Captions or a text transcript should be provided for all audio content that communicates page content; text equivalents are not needed for audio content that is decorative and does not convey meaning. Similarly, text or audio descriptions should be provided for all important visuals, including word-for-word transcripts of video content.

11.4.5 Support for Printing

A Web page is designed to support printing of page contents. Documents that are likely to be printed should appear on one page, and page width should be narrow enough to print correctly on standard paper sizes. Each page should identify its URL so the print version of a document can be linked back to its online source. Pages that contain long documents should support two versions of the document, one that is optimized for viewing online and the other with a layout that is optimized for printing.

Downloaded from http://www.everyspec.com

This page intentionally left blank.

Downloaded from http://www.everyspec.com

12. Web Site Design

12.1 Site Structure

12.1.1 Information Organization

A Web site provides a structure for organizing its content that reflects the users' view of the information space and is based on the tasks users want to perform at the site. If a site is organized sequentially, it arranges information in an ordered sequence (e.g., alphabetically, chronologically). A sequence-based site, which is appropriate if information needs to be reviewed in a linear fashion, is recommended only for the presentation of content such as web-based training. If a site is organized in a web structure, information is arranged according to unique associative patterns, with links both to information elsewhere in the site and at other sites. A web-linked site provides the flexibility to structure information in a non-linear manner but its complexity may be confusing to users.

If a site is organized hierarchically, the upper levels of the information structure address general topics, with lower levels providing progressively increasing depth in detail or specialization. Most Web sites use a hierarchical structure because it is the best approach for organizing complex bodies of information. When a hierarchical site is created, important content should be placed as close to the top of the hierarchy as possible. A "flatter" structure presents more information to users sooner and requires fewer jumps for them to find the content they are seeking. Some Web style guides recommend using a structure that is as shallow as possible. For example, Morris and Hinrichs in Web Page Design provide the following guidelines with regard to information structure:

- Users can reach important or frequently accessed information in a maximum of three jumps.
- Users can reach 80 percent of the information they may want to view in four to five jumps.
- Users can reach any information in an application in seven jumps.

Other style guides argue that the effectiveness of an information structure should not be defined in number of jumps but instead by whether users are successful at finding the target content they are seeking. As users explore a well-structured site, pages should become progressively more specific and provide a path to the information desired.

If a Web site is very large and complex or does not lend itself to a hierarchical structure, subsites can be defined within the information space to address specific content. A subsite is a collection of Web pages that provides its own home page and has a common style that may differ from that at the site as a whole. The pages within the subsite support local navigation back to the subsite home page as well as to the home page for the overall site.

Downloaded from http://www.everyspec.com

12.1.2 Site Structure

The logical point of entry into a Web site is the home page which presents an overview of the site structure and provides links to all major content areas and support tools, as shown in figure 12-1. These tools include various search and navigation aids for locating information in the site or in other related sites (e.g., branches (1) and (2) in figure 12-1). Other menu pages further subdivide the top-level structure and provide links to subordinate nodes in the information space. Content pages present detailed information about a topic and may include links to other pages in the site or in other related sites. Access to this content is provided at varying levels of nesting below the home page (e.g., branches (3), (4), and (5) in figure 12-1).

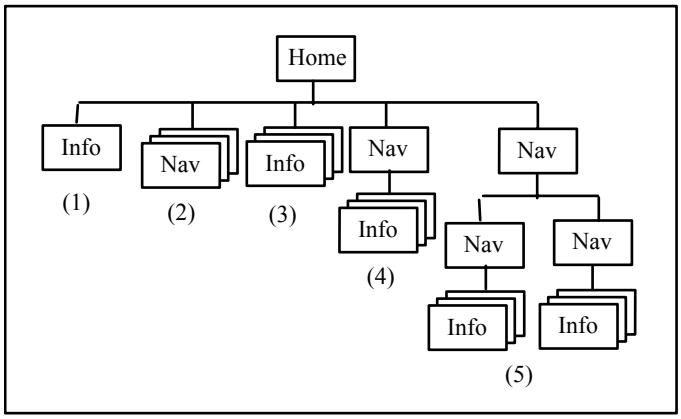


Figure 12-1. Example Web site structure.

Links provide the connectivity among the pages in a Web site, allowing users to navigate among the pages in nonlinear ways and to follow different paths at different times. A site defines links to complement the navigation mechanisms provided by the browser so users can easily explore the information space. At a minimum, a site supports navigation to the page(s) above and below the current one in the structure, navigation to sibling pages and other portions of the site, and navigation to support tools such as search, feedback, and help. The links are bi-directional, allowing the user to move up to the home page or other navigation pages as well as downward through chains of linked pages.

The content of a Web site is organized in terms of a metaphor or theme that is familiar to users and will assist them in locating and understanding the information they are seeking. For example, access to online documents can be organized in the form of a document library, and information about an organization can be represented graphically in an organization chart or office floor plan. If a site will be used by different audiences, it provides multiple paths through the information hierarchy that are tailored to the needs of each audience, with each path leading to the same content. A site also supports multiple methods for users to access the information they are seeking. Users can browse information efficiently as well as use search capabilities (e.g., index, keyword search) to locate information quickly. If the information structure is complex, a site includes a site

Downloaded from http://www.everyspec.com

map or directory that provides a global overview of the structure, with links that directly access its contents.

Downloaded from http://www.everyspec.com

12.2 Site Elements

12.2.1 Home Pages

A home page provides access to all major content areas and support tools in a Web site. Most home pages provide menu-like lists of text and/or graphic links that identify these content areas and serve as the "home base" for navigating the site. A home page can also include featured content (e.g., announcements, event calendars, alert messages) and links pointing users to specific parts of the site that are tailored to their needs or interests. Splash screens that display a graphic or animation should not be used as a home page because they require users to navigate to another page before they can access site content.

The layout of a home page is designed so it adapts to the width of the current viewing area and displays its content without scrolling horizontally. If possible, the length of the page should be short enough so users can view its content without scrolling vertically. If a home page requires scrolling, the density of links is highest in the top part of the page. Because a home page is visited frequently, the number of graphic images on the page should be minimized in order to reduce download time.

A home page contains a larger logo and a more prominently placed site name than other pages in the site. If the home page relies on links displayed as image buttons or embedded in image maps, it also provides alternative text-based links that remain visible if graphics are disabled by users. These links can be displayed in a small-size font at the bottom of the page (e.g., below the page footer) where they are accessible but do not disrupt the overall design of the page.

12.2.2 Menu Pages

A menu page provides a list of links that are related to one or more of the topics presented on the home page and serves as a mini-home page for that portion of the Web site. The menu page also includes links to other parts of the site and a link back to the home page.

As with a home page, menu pages are designed so they adapt to the width of the current viewing area and display their content without scrolling horizontally. If possible, the length of the page should be short enough so users can view its content without scrolling vertically. If a menu page requires scrolling, the density of links is highest in the top part of the page. In addition, the page should be designed so users are aware that some of the content is currently hidden; if the bottom of the first screen contains a separator or white space, users may assume that there is not more information on the page. Because menu pages are visited frequently, the number of graphic images on the page should be minimized in order to reduce download time.

Users prefer to interact with a few, more densely packed menu pages, rather than multiple pages each with only a few links. A menu page should contain at least five to seven links organized in a logical manner (e.g., alphabetically, hierarchically) based on

Downloaded from http://www.everyspec.com

the nature of the information. Nested lists can be used to group related links and show the structure and relationship among them. Each link contains 7-10 words that describe the destination in enough detail that users can determine where to navigate next. If desired, the menu page can include other information about the link such as information about the size or format of the destination page or if access is password-protected. The destination page should contain the same keywords included in the link plus more specific content.

12.2.3 Site Index Pages

A site index page provides an index of the information in a Web site. The page can be organized as a collection of text links that mirror the structure of major menu and submenu pages in the site, or the page can present a list of keywords identifying topics addressed in the site and organized alphabetically. If an index page contains a list of keywords, the entries should be formatted so users can scan them easily. Each keyword can serve as a link to the page with information on that topic, or the page name(s) can be included with the keyword and serve as a link to the information. A letter key is provided at top and bottom of the page; selecting one of the letters navigates to that part of the index.

12.2.4 Data Entry Pages

A data entry page is an electronic form with interface controls that allow users to interact with the information on the page. The page contains controls such as text boxes, radio buttons, check boxes, and list boxes for selecting and entering the information and at least one push button for submitting the completed form. The controls are defined as "form" elements, with as many of the appearance and behavior attributes described in sections 6 and 7 as can be supported by the form.

All of the controls needed to complete the data entry task are included on the same page so users do not have to remember information entered on one page while working on another. A data entry page can contain more than one form, but forms cannot be nested. Whenever possible, the controls are placed on the page so users can interact with them without scrolling. The controls are arranged on the page according to the design guidelines in section 8.

Error checking is performed when users select the "submit" push button to commit the data entered. If an error is found, the data entry page is displayed again, with all of the users' input shown along a description of the error. Users correct the error, then select the "submit" push button again. The error checking process is repeated until all errors are corrected.

12.2.5 Search Query and Results Pages

It is recommended that a Web site with more than 100 pages provide a search capability, either using a commercial tool or creating one that is tailored to site requirements. In the latter case, the query page provides a data entry area where users type a keyword and

Downloaded from http://www.everyspec.com

then activate the search engine that locates each instance of that word in the site. Alternatively, the query page can support concept-based searching where users type a word and the engine locates instances that are related to or "about" the keyword. If desired, the query page can support both basic (i.e., based on keyword only) and advanced (i.e., including search-refining options) searches. For example, the user might choose to search on more than one word or to limit the search to pages in a particular language. The scope of the search should be limited to the current site (and not include the entire Internet), with an option to target the search to a section or subsection of the site if desired.

A search results page displays the search parameters, along with a list of "hits" which are ordered based on the extent to which they satisfy the parameters. Duplicate occurrences of the same page are removed from the search results, and pages that come from the same site are grouped together. If the number of "hits" is large, the results should be grouped and displayed in categories, with the most likely categories presented first. Each item in the "hits" list includes the page title (which serves as the link to that destination), the first 150-200 characters of the page abstract included in the page header, and the full URL for the page.

If users select one of the items in the "hits" list and the destination page is within the current Web site, the content of the page should be displayed to draw attention to those elements that match the search parameters. For example, all occurrences of the search terms can be highlighted on the page.

12.2.6 Content Pages

A content page presents text and/or graphic information related to a particular topic. While the length of the page depends on the nature and complexity of its content, users prefer pages containing short "chunks" of information that can be located and scanned quickly. However, extremely short pages can increase reading time by requiring users to download information about a topic on separate pages rather than reading it on a single page. If a page contains long passages of text, it should provide an option to save the content to disk and/or print it for reading offline.

A content page is designed according to a layout grid that defines a consistent format for text and graphic alignment, use of headings, and common navigation links. Important information is placed at the top of a content page so users will read it first and then read the other content if they need to see more detail. Because links can be distracting, they should not be embedded in the body text on a content page. Wherever possible, they should be placed at the beginning or end of a block of text, or they should be grouped at the bottom of the page or included in a panel to the right of the text.

The following are specific types of content pages that may be included in a Web site:

- An FAQ page presents a collection of frequently asked questions and answers organized by topic. The top of the page contains an alphabetized topic list that links

Downloaded from http://www.everyspec.com

to the section of the page on that topic. If desired, the FAQs for each topic can be identified by a keyword or phrase describing the subject addressed and ordered alphabetically by subject within each topic. If the FAQs are identified by subject, the page includes an index containing an alphabetized list of subjects with links to each FAQ on that subject.

- An error message page is displayed instead of a standard error screen (e.g., "404 error, file not found") when a page cannot be located. The page follows the same design template as other pages in the site and presents likely explanations for the error, suggests alternatives, and provides links to the home page, site index page, and/or search page for the site.

- A "What's New" page presents a list of changes to the content of a Web site and the date the change was made. If desired, the description of the change can include a link to the page containing the change.

Downloaded from http://www.everyspec.com

12.3 Site Design Considerations

12.3.1 Site Labeling

The URL for a site defines its Web address, with the path and file portions of the address uniquely identifying each page in the site. Users frequently rely on these addresses to understand the structure of the information they are viewing. As a result, a site defines its URLs so they have readable names that reflect the nature of the information they contain. In addition, URLs should be constructed to mirror the structure of the site and support "URL butchering" by users (i.e., deleting the end portion of a URL as a shortcut to navigate to a higher level of the site hierarchy).

A Web site uses only lower-case characters in its URLs so as to minimize the risk of errors when users enter an address. To ensure portability across platforms, the site includes the appropriate file extension for the content type (e.g., .gif for GIF files), uses only alphanumeric characters and underscore, hyphen, and period in its file names, and limits filename length to less than 32 characters. If a URL contains several short words, they should be run together (e.g., "www.testsite.mil"), rather than separated by a hyphen or period; a site should include abbreviations in a URL only if the name contains three or more words or is excessively long or difficult to spell.

A Web site includes all official notices concerning access to and use of DoD Web sites. This information should be presented on the home page or accessible from it. If a site uses copyrighted information from other sites, it obtains permission from the copyright holder before doing so. If a site is available to the general public, its contents have been approved for releasability and all of its links navigate only to other publicly available sites.

12.3.2 Site Updating and Maintenance

A logical pattern of links among pages in a Web site is defined so information content can be updated and maintained quickly and easily. The following guidelines apply to site updating and maintenance:

- When an update occurs, all local links within a site should be tested to ensure the destinations are valid, and all remote links should be checked to ensure they remain active and are being maintained.

- Pages that are time-sensitive or address a topic that changes frequently should be checked for currency, and out-of-date information removed.

- When the URL for a site changes, a page with the old address should be provided that identifies the new address and includes a link to it.

Downloaded from http://www.everyspec.com

13. Information Presentation

13.1 Text Information

13.1.1 Font Attributes

Guidelines concerning font attributes such as typeface, style, size, color are provided in section 10.2.2.1 for graphical user interfaces and in section 11.3.1.1 for Web user interfaces.

13.1.2 Grammar and Wording

Continuous text is presented following normal capitalization and punctuation rules. Guidelines for capitalization of graphical interface components and labels are provided in section 10.2.1.3. The following guidelines apply to grammar and wording:

- Arabic rather than Roman numerals should be used when information has to be numbered.
- Continuous text should be phrased in simple sentences, in the affirmative (rather than negative), and in active (rather than passive) voice. The referent for "it" or "they" in a sentence should be easily identified, and contractions and hyphenation should be avoided.
- A sequence of events or steps should be presented in the order they are performed.
- Paragraphs should be short (e.g., four to eight lines) and separated by at least one blank line.

Acronyms and abbreviations are used only when they are significantly shorter than the full word and are commonly understood by users (e.g., are related to normal language or are specific job-related terminology). Acronyms and abbreviations are the shortest possible length that will ensure uniqueness and are presented without punctuation (e.g., periods or other punctuation marks).

13.1.3 Date/Time and Latitude/Longitude Formats

The following formats are used to present date and time information:

- Date is displayed as YYYYMMDD, where YYYY is the year, MM is the month, and DD is the day, or as DD MMM YYYY, where DD is the day, MMM is the month, and YYYY is the year. In the former format, year, month, and day are all digits, and leading zeros are included as needed. In the latter format, month is indicated in capital letters, and leading zeros are included as needed.

Downloaded from http://www.everyspec.com

- Time is displayed as HHMM[SS]Z, where HH is the hour of a 24-hour day, MM is the minute, SS (optional) is the second, and Z is the time zone (Zulu is the default). An application provides users with the option to select other time zones as required. Leading zeros are included as needed. If desired, an application can include colons and/or spaces as part of the output format to improve readability.

- Date/Time Group is displayed as DDHHMMZ MMM YYYY, where DD is the day, HH is the hour, MM is the minute, Z is the time zone (Zulu is the default), MMM is the month, and YYYY is the year. Month is indicated in capital letters, and leading zeros are included as needed.

The following formats are used to present latitude and longitude information:

- Latitude is displayed as D{D}H, where D (one or two characters) is the degrees of latitude and H is the hemisphere (N for North, S for South), or as DD{MM{SS}}H, where DD is the degrees of latitude, MM is the minutes of latitude (optional), SS is the seconds of latitude (optional, but can only be given if minutes of latitude is provided), and H is the hemisphere (N for North, S for South).

- Longitude is displayed as D{D{D}}H, where D (one, two, or three characters) is the degrees of longitude and H is the hemisphere (E for East, W for West), or as DDD{MM{SS}}H, where DDD is the degrees of longitude, MM is the minutes of longitude (optional), SS is the seconds of longitude (optional, but can only be given if minutes of longitude is provided), and H is the hemisphere (E for East, W for West).

If latitude and longitude are presented together, latitude should precede (either above or to the left of) longitude, and the output format includes the appropriate symbols for degrees, minutes, and seconds to improve readability.

Downloaded from http://www.everyspec.com

13.2 Graphical Information

13.2.1 Line Graphs and Surface Charts

A line graph, shown in figure 13-1, is used to present trend information, spatially structured information, time-critical information, or relatively imprecise information. The axes of the graph are labeled and include the units of measurement as appropriate. The minimum and maximum values are indicated on each axis, with up to nine intermediate markings showing gradations on the axis. The starting point on each axis should be zero, with the gradations indicated in whole numbers, unless a zero starting point is inappropriate for the data being displayed. The gradations should be at standard intervals (e.g., 1, 2, 5, 10), with intervening gradations consistent with the scale interval.

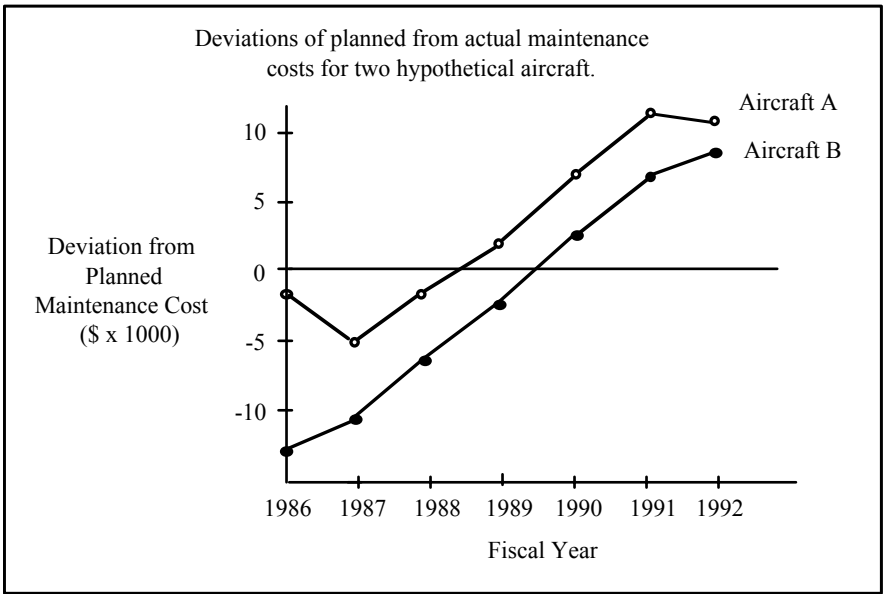


Figure 13-1. Example line graph.

A line graph contains no more than five lines, with each one identified by an adjacent label (rather than in a separate legend). The labels are oriented for normal reading and located next to the data being referenced. Critical or abnormal information is highlighted (e.g., by color, line thickness, annotation) to draw attention to it. If grid lines are included in the graph, they should be unobtrusive and not obscure the data plotted; users can display or suppress grid lines as desired.

If users are required to compare multiple lines, they should be presented on a single graph. If related data have to be presented in multiple graphs, the same coding scheme should be used on each graph, and users can redraw the graphs using the same scale to facilitate comparison. If users have to read precise values from a graph, they can display the actual data values on the graph and zoom the graph if necessary. The graph includes

Downloaded from http://www.everyspec.com

aids for scale interpretation (e.g., displays a grid upon request, provides vertical and horizontal rules that users can move to the intersection point).

A surface chart is a type of line graph in which the data being depicted represent all parts of a whole. The lines are stacked above one another to indicate aggregated amounts, and the area between each line is coded using different colors, shadings, or textures and labeled. The data categories are ordered to reflect the logical organization of the entity being displayed; if no ordering scheme exists, the data categories are ordered so the least variable data categories are presented at the bottom and the most variable ones at the top.

13.2.2 Bar Charts

A bar chart, shown in figure 13-2, is used to compare a single measure at several intervals. The bars are oriented either vertically or horizontally, and bars containing data to be compared are presented adjacent to one another. If the displayed data have to be compared with a critical value, a reference index should be provided. The axes of the chart are labeled and include the units of measurement as appropriate. The labels are oriented for normal reading.

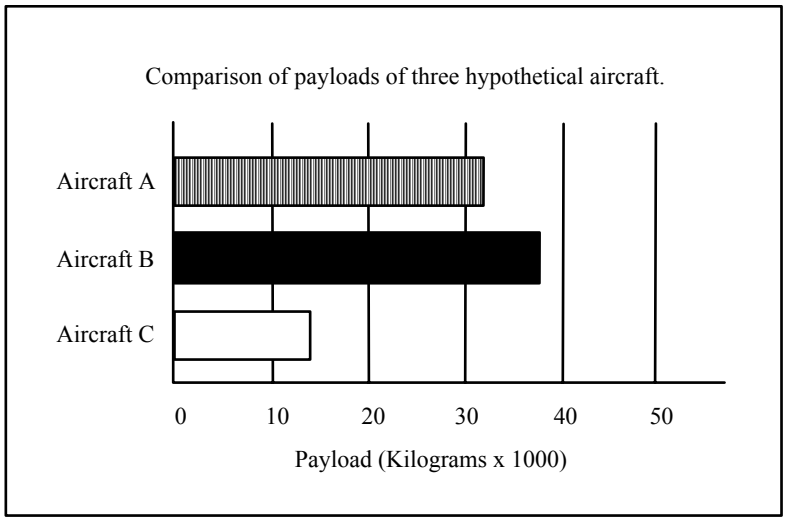


Figure 13-2. Example bar chart.

If the number of bars being displayed is small, they should be separated from each other, using one-half or less of the bar width as the spacing between them. If the number of bars is large, they should be placed adjacent to each other. Coding (e.g., color, shading, texture) is used to distinguish among different groups of bars or to highlight important data in one or more of the bars. Related groups of bars should be presented in a consistent order across multiple charts. If the bars are presented individually, each is labeled; if the bars are presented in pairs, they are labeled as a unit, with a legend provided that identifies each bar.

Downloaded from http://www.everyspec.com

Stacked bars are used to display the total measures and the portions represented by each part of the whole. The data categories are presented in the same sequence in each bar. Data categories are ordered so the least variable are at the bottom of the bar and the most variable are at the top. The areas within each bar are coded using different colors, shading, or texture and labeled.

13.2.3 Flow Charts

A flow chart is used to present a schematic representation of the decision points in a sequence or process. The path indicated in the flow chart is left to right, top to bottom, or clockwise. Each step in the sequence identifies a single decision, as shown in figure 13-3. The steps are coded (e.g., using different symbols and shapes) and include directional indicators (arrows) to identify the sequence to be followed. A legend describes each code used in the flow chart, and the text in the chart is oriented for normal reading.

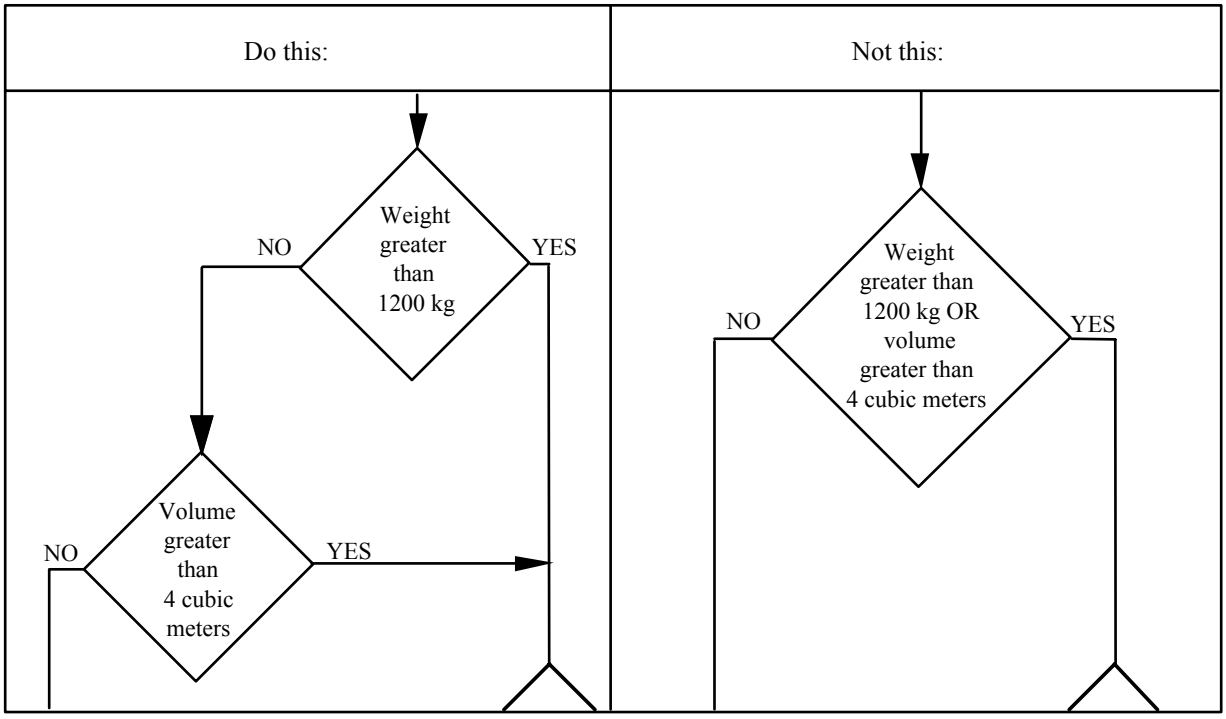


Figure 13-3. Example of labeling decision points in a flow chart.

The steps in a flow chart are ordered logically (i.e., follow the steps in the sequence or process from start to finish), or the most important decisions or the decisions that can be made with the greatest certainty are placed first. If no ordering scheme can be identified, the flow chart should be structured to minimize the length of the path through it. Important elements (e.g., paths through the chart) and critical information are highlighted.

Downloaded from http://www.everyspec.com

13.2.4 Pie Charts

A pie chart, shown in figure 13-4, is used to provide an approximation of how an entity is apportioned into component parts. The segments in the chart are coded using different colors, shadings, or textures and labeled. If the segment is too small to contain the label, it should be placed outside the segment, with a line from the label to the segment. The label describes the content of the segment, includes the number (i.e., percentage or actual value) being represented by the segment, and is oriented for normal reading. Important segments are highlighted by using special shading and by displacing them slightly from the remainder of the pie chart.

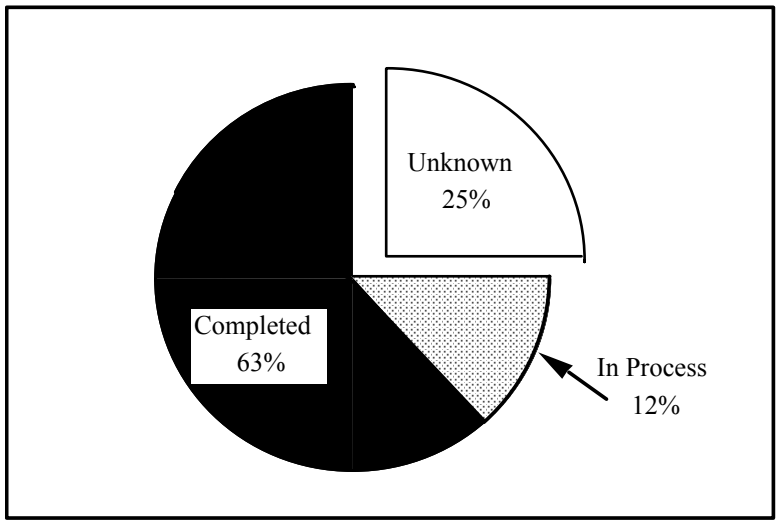


Figure 13-4. Example pie chart.

13.2.5 Graphic Schedules

A graphic schedule, shown in figure 13-5, is used to present the time sequence for a series of tasks. Time is presented on the horizontal axis, with tasks arrayed vertically. A task can be decomposed into events that are represented as icons. No more than nine unique icons are presented on a schedule at one time. The icon should be displayed to the right of its associated task, with the length of an icon proportional to the amount of time required to complete the task. If the schedule is cluttered or users require a high degree of precision, gridlines can be used to indicate the date and time; users can display or suppress the gridlines as desired.

Downloaded from http://www.everyspec.com

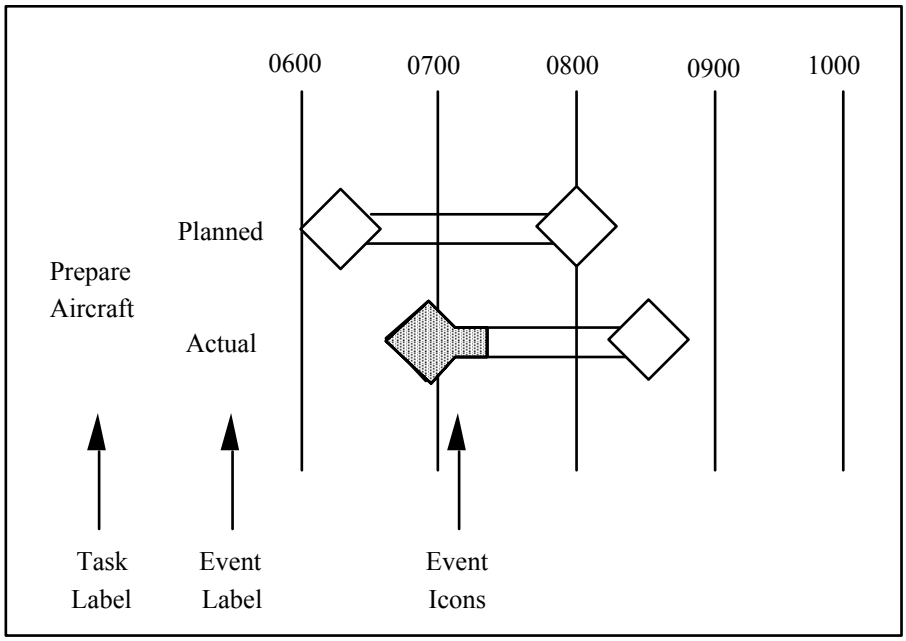


Figure 13-5. Example graphical schedule.

If a task includes more than one event, each of the icons is labeled, and the labels are placed along the vertical axis or on or above the timeline. If appropriate, different scheduling attributes can be represented as symbols with the event icons. These symbols can be formed from various geometric shapes (e.g., circles, diamonds, squares) and coded (e.g., using different fill patterns) to indicate various schedule situations. If different types of events (e.g., ones undertaken at different locations) are presented, they are differentiated by color or shading or include a label (e.g., an alphanumeric designator) displayed on or above the event icon. If a coding scheme is used, a legend describing the coding technique should be included with the schedule.

Downloaded from http://www.everyspec.com

13.3 Information Coding

13.3.1 Color

The human performance literature indicates that the effectiveness of color coding varies with the task being performed. Color has been found to be a more effective code than other cues such as shape, size, or brightness when performing search and symbol identification tasks. The performance advantage of color increases with the density of the symbols in a display and when the number of nontarget symbols of a different color than the target increases. However, color used excessively or inappropriately in displays can degrade user performance when compared to monochromatic displays, and the addition of color can increase response time and the likelihood of error due to color confusions.

If color is used, it is available as a redundant code; color is the sole basis for coding only when other coding methods (e.g., shape, size) are inappropriate. The number of colors used for coding in an alphanumeric display does not exceed seven, with only four codes displayed at any one time. The number of colors used for coding in a graphic display does not exceed nine. When users encounter information that is color coded, they should be able to display the meaning of the code (e.g., view this information in the status bar). Slight shade changes in color are not used to show gradation or choice. However, if variations in shading are used for coding, the shading should be of sufficiently differing intensity as to be discriminable and not used to indicate object selection or for control of an application. If the levels of information being coded are ordered, the colors chosen to represent the levels should preserve the order.

The following conventions apply when color is used to indicate status:

Green/Blue = Operational/Normal/Noncritical
Yellow = Caution/Questionable
Red = Inoperative/Error

Section 14.2.4 provides examples of color use in coding tactical information. Mayhew in Principles and Guidelines in Software User Interface Design provides the following guidelines regarding color coding:

To denote:	Use:
Larger size	Saturated or bright colors
Smaller size	Desaturated or dark colors
Equal size	Colors equal in brightness
Heaviness	Saturated, dark colors
Lightness	Desaturated, light colors
Depth	Saturated, dark colors
Closeness	Saturated, bright colors
Height	Desaturated, light colors

Downloaded from http://www.everyspec.com

Low-end continuum	Short-wavelength dark colors
High-end continuum	Long-wavelength bright colors

The following guidelines apply to the use of color in computer display systems:

- Highly saturated colors, opposing colors (e.g., yellow and blue), and colors at spectral extremes (e.g., yellow and purple) should not be used together because they can cause afterimages, shadows, and depth effects.

- Pure white text should not be displayed on a pure black background because this combination produces halation that makes the text less readable. Saturated blue should be used only for background features in a display and not for critical data or for small lines or dots when the background is dark.

- Color perception can be affected by both brightness and type of lighting (e.g., incandescent vs. fluorescent). For example, bright ambient light desaturates display colors and can result in degraded color identification and discrimination.

- At normal viewing distance, maximal color sensitivity is not reached until the size of a colored area exceeds about a three-inch square. Smaller-size images become desaturated and change slightly in color. Also, small differences in actual color may not be discernible, and small adjacent colored images may be perceived to merge or mix.

- Color discrimination is better when color images are displayed on an achromatic background (black, gray, or white) and when achromatic images are displayed on a color background. If color images are displayed on color backgrounds, then background and symbol colors should contrast in both brightness and hue to ensure legibility.

13.3.2 Reverse Video

Reverse video is not used for coding since it already denotes highlighting in a graphical user interface (e.g., to indicate that an object has been selected). Reverse video can also reduce legibility and increase eye fatigue.

13.3.3 Size and Shape

If size is used, the number of size codes is limited to five or less.[1] Users should be required to interpret relative size rather than absolute size. Care should be taken when

[1] MIL-STD 1472F indicates that no more than three size levels shall be used.

Downloaded from http://www.everyspec.com

using size and color coding together since users' perception of object size can be manipulated by varying the color saturation and lightness of the object.

If shape is used, the number of shape codes is limited to five or less. The shapes used should relate to the object or operation being represented. Simple geometric shapes (e.g., circle, triangle, square) should be used, with color and detail added only as required for users to identify the meaning assigned to the shape.

13.3.4 Typography

In general, variations in typography are not used for coding, since they may conflict with font attributes selected by users in a system-level or browser-level setting and be illegible when rendered. However, if variations in typography are used, they are limited to no more than two font styles (e.g. regular and italics), two weights (e.g., regular and bold), or three sizes at one time. Capitalization can be used for emphasis in text, but it should not be the sole indication of critical information. While underlining can be effective in drawing user attention to specific text information, it should be used sparingly because it can reduce legibility and be confused with other interaction features (e.g., links on a Web page).

13.3.5 Flashing

Flashing is used only to display urgent information for user attention. No more than two levels of flashing with equal on/off intervals are used. If flashing is applied to a displayed item, a flashing symbol (such as asterisks) should be used rather than flashing the text itself. Users are able to acknowledge the event causing the flashing and suppress it if desired.

If compliance with accessibility standards is required, the flash or flicker frequency should not be greater than 2 Hz or lower than 55 Hz.[2]

13.3.6 Sound

If auditory signals are used to alert users about critical conditions or operations, they are used redundantly with a visual notification of the alert (e.g., changing the appearance of an icon, presenting a message window). If auditory signals are associated with noncritical operations (e.g., as a supplemental means for information presentation), they are used sparingly and users can acknowledge and turn off the signal at their discretion. Auditory signals should be intermittent in nature and distinctive in intensity and pitch, and the number of signals provided to users should not exceed four. The intensity, duration, and source location of the signal should be selected to be compatible with the acoustic environment of users and the requirements of other personnel in the surrounding area.

[2] Appendix B describes federal standards for information accessibility by individuals with disabilities.

Downloaded from http://www.everyspec.com

13.4 Dynamic Information

If visual information is being updated automatically, users are able to control the rate at which the updates occur. Users can freeze the display of the information and resume the updating either at the point of stoppage or at the current point in time. If users have to read dynamically changing information reliably and accurately, the update rate should be no more than once per second. If users have to identify the rate of change or read gross values, the update rate should be 2-5 times per second. Users should be prompted to return to automatic updating after freezing a display (e.g., while users execute a print command) and should be informed if significant changes occurred while the display was frozen.

Similar capabilities are available if users interact with auditory information. Users can control the playback of auditory information (e.g., start, stop, pause) as well as adjust the volume of the playback.

Downloaded from http://www.everyspec.com

This page intentionally left blank.

Downloaded from http://www.everyspec.com

14. Map Windows

14.1 Window Design

A map window, shown in figure 14-1, includes a menu bar and map area and may include a toolbar or palette and a status bar. The map area can present a two-dimensional (2D) or a three-dimensional (3D) perspective view of geo-spatial data, or it can be a blank area defined by some coordinate grid. If the window contains a toolbar and/or status bar, users can configure their contents as well as hide them in order to increase the size of the map area. A map window may allow users to view only one map at a time, or it may use a tab control to provide access to multiple maps simultaneously in a single window. While a tab control increases the number of maps available to users, this implementation can reduce the size of the map area and add unnecessary complexity to the management of the information by users.

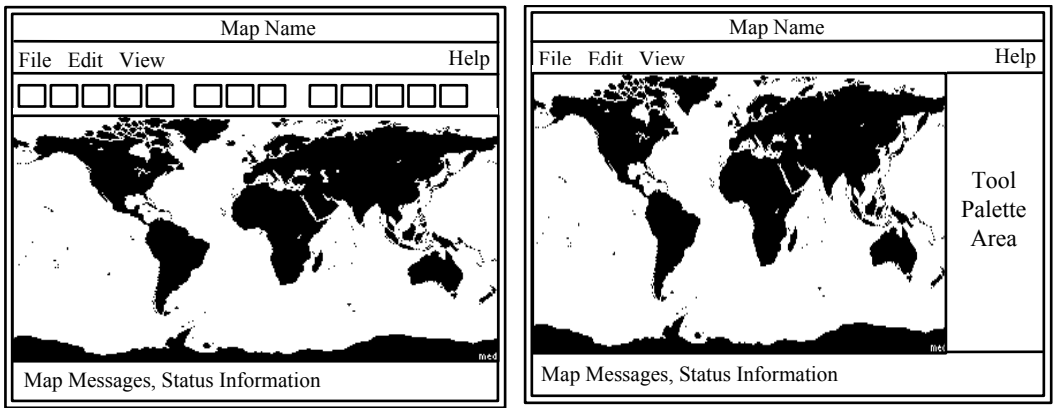


Figure 14-1. Example map windows.

Menu bar. The menu bar provides access to basic map display and manipulation functions and to application-specific map accessories (e.g., tactical decision aids, terrain analysis tools) that address mission requirements. If an application supports both 2D and 3D perspective displays, menu bar organization and content should be similar for both views in order to provide users with consistency in how they access map display and manipulation functions. The contents of any common menus (e.g., File, Edit) included in a map window comply with specifications in section 8.1.1.1. The leftmost menus in the menu bar contain options related to map display and manipulation, with any application-specific menus added to the right preceding the Help menu.

Toolbar. Redundant access to frequently executed map functions is provided in a toolbar or palette area in the window itself (see figure 10-1) or in a palette contained in a separate dialog window. A tool palette is recommended in map windows that contain a 3D perspective view in order to provide sufficient space from which to access the set of controls (e.g., sliders, thumbwheels) needed for display manipulation. The toolbar is

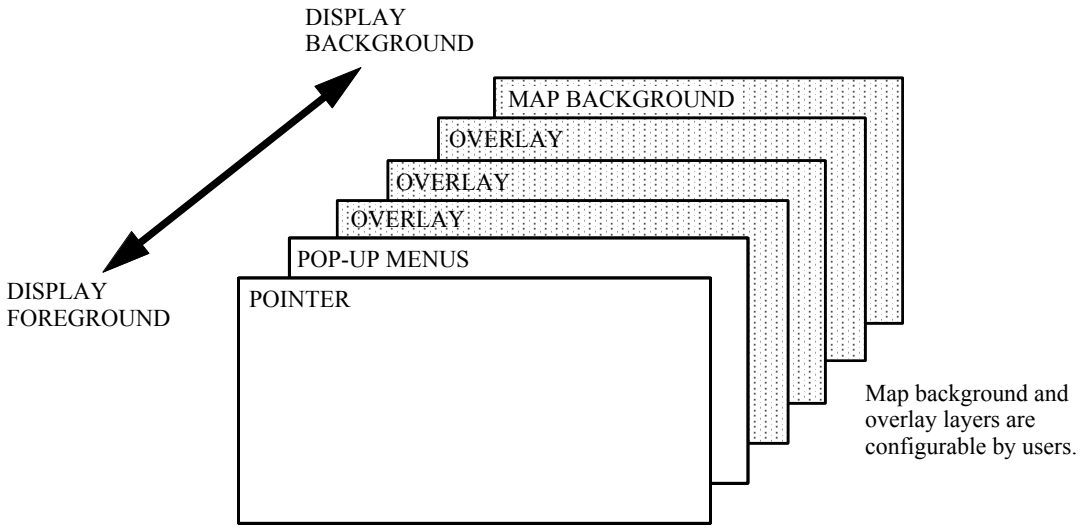
Downloaded from http://www.everyspec.com

designed in accordance with specifications in sections 6.4 and 8.1.1.2, with the content tailored to fit mission requirements and configurable by users.

If a toolbar is included in a map window, it is located below the menu bar (with a show/hide feature available in a View menu). If a tool palette is provided, it is placed next to or below the map area and may be a dedicated area in the window or controlled from a show/hide option in the View menu. If desired, an application can allow users to undock and re-dock the toolbar or tool palette. If a separate tool palette is used, it is accessed from the menu bar in the map window; the dialog window containing the palette can be positioned as desired (e.g., adjacent to the map area) and closed when no longer needed.

Map area. The map area consists of a series of layers containing a map background and one or more overlays. The map background contains the map itself and may also include features such as country boundaries, cities, rivers, and bridges that are attributes of the map. Each overlay contains a collection of map objects specified by the user. These objects include (1) tactical symbology representing entities such as ships, aircraft, or equipment, (2) user-defined map features, (3) tactical graphics such as points of interest, fortified areas, and minefields, and (4) drawn graphics such as circles, rectangles, and lines.

The map is the background layer in the map display space, with individual overlays arrayed above it, as shown in figure 14-2. The pointer is always the topmost layer and cannot be obscured by the objects at any layer or by a pop-up menu when one is displayed. Users can configure overlay layers and their contents and manipulate them in relation to the map background. The map space as a whole, the map background, and individual overlays and objects each possess characteristics that define what it is (i.e., properties), how it behaves (i.e., operations), and how it is influenced by other objects (i.e., relationships). Users can define and manipulate these characteristics from a pop-up menu available for each object.



Downloaded from http://www.everyspec.com

Figure 14-2. Representation of the map information space.

A 3D perspective display projects a volume of 3D space or information onto a 2D display screen. It is assumed that the screen is the x-y projection plane and the z-axis is orthogonal to the screen and pointing toward the user. A 3D perspective display creates the appearance of depth along the user's principal line of sight. Depth judgments require the user to judge which object is closer along the z-axis; altitude judgments require the user to judge which object is higher above the ground plane. The user relies on perceptual cues such as linear perspective, surface texture, gradients of illumination, occlusion, and size-distance relationships when forming these judgments.

Status bar. The status bar presents map messages (e.g., feedback during map redraw) to the user and displays map information such as map center, coordinates, and width. The status bar can also display the coordinates of the pointer position when it is on the map. The status bar complies with specifications in section 8.1.1.3, with specific contents and format tailored as needed by an application.

Downloaded from http://www.everyspec.com

14.2 Displaying and Interacting With Map Information

This section identifies the set of default map functions that are expected to be available in a map window; it is assumed that an application will select from this set and define extensions (e.g., in terms of map accessories) as needed to meet mission requirements. The functions apply to both 2D and 3D perspective displays, unless otherwise indicated.

14.2.1 Map Display and Manipulation

Users can select the default map or a saved map to view in the map area. An application can define the default map, or it can be one designated by users. Users should be able to identify a saved map by name, area covered, or attributes. An application includes identifying information (e.g., map name, area, projection, scale) with the map when it is displayed in the window (e.g., as part of the window title, in the status bar, or from a button in the tool area that presents the information in a separate window).

An application displays each map in the same orientation (north-oriented is the default). Users can change the orientation and display an annotation of orientation on the map if desired. Users can select the map features to be rendered and specify the draw rules (e.g., colors, labels) for these features. If appropriate, users can create special (i.e., user-defined) features and specify their placement and appearance.

If an application provides a 3D perspective display, users are able to define and then save the default orientation for a perspective view when it is first displayed (e.g., north/south/east/west perspective, degree of rotation from a 2D view). Users are also able to select and then change the point or object being viewed or attach the eyepoint to an object. Users can show and hide grid lines (e.g., latitude/longitude) and, if appropriate, modify the nature and placement of the light source illuminating the display space. When users modify the light source, the brightness and shading of objects change as needed to provide appropriate depth cueing and shadow effects. If appropriate, users can distort perspective (e.g., set the altitude scale to exaggerate terrain features along the vertical axis so they are easier to see).

An application provides a continuous coordinate indicator of the pointer location on the map (e.g., in the status bar of the window). The coordinates are updated as the user moves the pointer on the map and are expressed at a level of precision and in a coordinate system specified by the user. An application makes various map reading tools available so users can perform pointer-based queries (e.g., to identify coordinates and elevation, determine distance and bearing between points on the map). Users can enter latitude and longitude to the level of accuracy needed; when calculations such as range, bearing, and position are performed, the answer computed should reflect the degree of accuracy appropriate to the scale of the map displayed. If an application provides a 3D perspective display, users have access to enhanced map reading tools (e.g., line of sight time/distance/speed calculations, terrain masking) that are computed for the 3D display space.

Downloaded from http://www.everyspec.com

An application supports multiple methods for changing the map view. Users can set or reset the map center and width by entering the coordinates or clicking directly on the map. Users can select the unit of measurement and/or scale for entering these values. Users can zoom the map by entering the coordinates, clicking directly on the map, or drawing a box on the area to be zoomed. Users can enter the zoom factor or select a preset value. Users can pan the map continuously and/or in jumps based on screen distance, geographic distance, or a percentage value.

If an application provides a 3D perspective display, users are able to define the eyepoint by entering specific x, y, z coordinates or by identifying an object (e.g., own unit) to use as this reference. Users are able to modify the viewing angle (i.e., rotate the field of view or viewing orientation) in the display. Users are provided with controls (e.g., in a tool palette) for rotating the display about three axes: x rotating to the left and right, y rotating upward and downward, and z rotating into and out of the view plane. Users are able to adjust the rotation continuously (e.g., using sliders, thumbwheels, or dials) as well as enter specific x, y, and z parameters (e.g., in text boxes). Alternatively, an application can allow users to invoke a “rotate” mode and then click directly on the map to rotate it in preset amounts. Users can specify the center of rotation (e.g., defining the map center specifies the axis for rotation), or an application can assume the center to be the screen-coordinate origin, the origin of the object, or the center of the object.

Users can define a baseline (i.e., home) position on a map and return to this position quickly (e.g., by selecting a toolbar button). If desired, users can display a reference map (e.g., in a dialog window) that shows a miniature view of the full map and identifies the portion currently visible in the map area. The reference map is updated whenever users change the map view. If an application provides a 3D perspective display, users are able to relate the 2D and 3D views of the display space. For example, an application identifies (e.g., by line annotation) the portion of a 2D reference inset being shown in perspective view in the main map area.

An application provides an overlay manager that allows users to show/hide each overlay (i.e., identify the ones that are “visible”), configure the order in which overlays are displayed, and indicate the one to be active (i.e., on top). If appropriate, users can adjust the intensity of the map background to “fade out” selected map features so map symbols and graphics are easier to see. Users can send the contents of an overlay in a message to other locations and enable overlay sharing if they have authorization to do so. Access to an overlay by other users can be global (i.e., available to all users), shared (i.e., available to a subset of users), or personal (i.e., available to a single user).

14.2.2 Interaction With Map Objects

A map object is defined by its outside boundary. If a label is displayed with the object, the label is considered part of the object. A map object is plotted so its geometric center is at the actual location of the object. If an object has to be offset (e.g., to reduce display clutter), it is connected to its actual location using lines, arrows, or other graphics. The views of a map object are linked across all windows that display information about the

Downloaded from http://www.everyspec.com

object. For example, when users select the name of a map feature from a list in a dialog window, the graphic for that feature on the map also highlights. If an application provides a 3D perspective display, users are able to show or hide vertical reference lines that connect individual objects to the map surface. These lines allow users to identify object position in a perspective display as well as estimate the altitude of an object and judge relative altitude between objects.

Users interact with map objects following the rules and conventions described in sections 2 and 3.

- Users place the pointer on a map object and click BLeft to select it. The object highlights (e.g., changes color, displays "handles") to indicate its select state, and it moves to the foreground so that it is not obscured by other objects.

- Users place the pointer on a map object and double click BLeft to open or view its contents. For example, double clicking on a track object displays a window with information about the track, and double clicking on a user-defined graphic object displays a window with information about the graphic (e.g., the overlay that "owns" it, the date created).

- Users place the pointer on a map object and click BRight to display a pop-up menu listing properties and operations for the object. For example, the pop-up menu for a track object might contain a Readiness Data option that executes a query against the appropriate database and displays these data for the object; the pop-up menu for a graphic object might contain options for editing the object (e.g., Cut, Copy, Paste). Users can select multiple map objects, then click BRight to display a pop-up menu with options that apply to all of the objects in the selection.

Users can interact with individual map objects as well as perform operations on the collection of objects in an overlay. Each overlay is assumed to be "transparent" except where an object is present. As a result, users can select any "visible" object on which they can place the pointer (i.e., any object not obscured by another object). Users should also able to interact with an object that is hidden by another object. For example, users might perform a "drill-down" selection in which they would place the pointer over the obscured object and "query" that point within the map space. A pop-up menu listing all objects that intersect the pointer would be displayed from which users can select the object of interest. A 3D perspective display is assumed to be "flat" for the purpose of object selection; i.e., users place the pointer within the boundary of an object and click BLeft to select it. Users can rotate the display to reduce overlap, or they can use drill-down selection, if available, to select obscured objects.

An application can allow users to specify the frequency and rate at which information about map objects is updated and to temporarily stop and then resume auto-updating. An application can also allow users to share map objects. An object can be defined as global, shared, or personal. Users can create, edit, and delete personal objects and can perform these actions on global and shared objects if they have authorization to do so.

Downloaded from http://www.everyspec.com

If an application provides a 3D perspective display, it can support a "flythrough" capability that allows users to play back the movement of an object through the map display space. Users can select an object, specify its motion parameters (e.g., position, velocity, and acceleration), and define the point of reference for viewing the animation (e.g., attach the eyepoint to the moving object). The speed of the animation is sufficient for users to perceive motion (at least 15 frames per second), and the attributes of the map display space are updated dynamically to provide consistent perceptual cues about the visual environment. Users can select the time step of the playback and stop, pause, and resume the animation as desired. If appropriate, users can save the animation as a "movie" to replay as needed (e.g., as part of a briefing).

14.2.3 Tactical Symbology and Graphics

An application complies with applicable military standards when displaying tactical symbology and graphics. MIL-STD 2525B defines common warfighting symbology for use by C4I systems in DoD. An application with requirements in a specific operational domain not addressed by this MIL-STD should comply with relevant national and international symbology standards. If an application creates new symbology, it should do so in ways that are consistent with applicable standards.

An application displays the tactical symbols in MIL-STD 2525B and can display other domain-specific symbols as appropriate. If an application supports multiple symbol sets, users are able to select among them and identify the one currently in use (e.g., as part of the identifying information in the status bar). Because symbol sets can assign different meaning to the same symbol, an application does not mix symbol sets on the same display.

Users can configure the appearance of symbol features to be displayed based on user-specified criteria (e.g., by affiliation and/or battle dimension). These features include the symbol attributes (frame, fill, icon), color (color/monochrome, frame/text color), and size (small, medium, large) and whether graphic and/or text modifiers are displayed with the symbols. If appropriate, users can create and display system-unique symbols and add special (i.e., user-defined) extensions that override default symbol features. If an application displays multiple symbol sets, it provides the appropriate dialog support so that users can configure symbol features as desired when interacting with each symbol set, and it ensures that symbol features are legible and discriminable on a variety of map backgrounds.

An application displays the tactical graphics in MIL-STD 2525B and can display other domain-specific graphics (e.g., weather) as appropriate. Users can configure the appearance of the graphics (e.g., size, color), specify the position (e.g., lat/long coordinates) where the graphics are placed, and select the text modifiers to include with the graphics. Users also have access to drawing tools for creating and placing user-defined graphics (e.g., line, polyline, circle, rectangle, freeform, text) on the map. Users can define the attributes (e.g., line type, thickness, color; text font, style, size, color; fill

Downloaded from http://www.everyspec.com

pattern, color) of the object being drawn and specify whether placement is to be fixed (i.e., attached to specific coordinates) or dynamic (i.e., attached to another map object so that the graphic moves with the object). Users can draw these objects directly on the map or enter the parameters that define their size and placement (e.g., enter the lat/long coordinates for the center of a circle and the length of the radius).

An application provides users with the capability to declutter the map space and to select objects in a cluttered environment. Tools for reducing visual clutter include (1) offsetting the placement of objects from their actual location, (2) repositioning labels so they are not obscured by other objects, with a line connecting each label to its object, and (3) supporting variable coding of objects (i.e., high-interest objects are rendered as symbols and low-interest objects as dots). To select objects in a cluttered environment, users can perform drill-down selection of obscured objects or select from a pop-up list of the objects in the current selection.

14.2.4 Coding Tactical Information

If an application uses color to convey tactical meaning, each category of tactical data is represented by a different color. Color serves as a redundant code with another attribute of the object (e.g., size, shape, text) and is not the sole basis for coding. The colors selected to convey tactical meaning should be used consistently throughout an application. If one of these colors is assigned another meaning, a different shade should be selected so as to minimize the likelihood of confusion. If an application uses color to indicate threat status, it does so in accordance with applicable military standards (e.g., red represents a hostile threat). Section 13.3.1 provides additional direction on the use of color in information coding.

If an application uses color for the purpose of alerting, only the tactical information to which an application wants to direct user attention is assigned a unique color. A standard meaning in terms of alert criticality is assigned to each color, and that color is used consistently to convey this meaning. While alerting is usually indicated by assigning color to text information (e.g., in a list or table), colored icons can also be defined and appended to the information. Examples of how color might be applied as an alert indicator are provided below:

Vulnerability Time	Probability of Detection
Red = Vulnerable now	Red = High probability (greater than X percent)
Yellow = Vulnerable in X minutes	Yellow = Medium probability (less than X percent but greater than Y percent)
Green = Vulnerable in Y minutes	Green = Low probability (less than Y percent)
Blue = Not vulnerable	

Confidence Factor	Priority
Red = Unknown	Red = High
Yellow = Low	Yellow = Medium
Green = High	Green = Low

Downloaded from http://www.everyspec.com

Probability of Hostile Action	Action Items
Red = Imminent	Red = Now
Yellow = Probable	Yellow = In X minutes
Green = Possible	Green = No time limit
Blue = None	Blue = No action required

Downloaded from http://www.everyspec.com

This page intentionally left blank.

Downloaded from http://www.everyspec.com

15. Interacting With Drag-Enabled Objects

15.1 Drag Transfer Methods

An application is expected to present users with a collection of objects and views of objects, with each view providing a visual representation of the objects.[1] Drag transfer offers a mechanism for taking objects shown in one view and displaying them in another view. The requirements presented here build upon the basic mechanisms described in section 3.4.2 to address drag transfer when the drag site and drop site are decoupled, the objects are managed by multiple data sources, the transfer affects object visualization and may also cause a database change, and the objects being transferred may not be drag-enabled. The drag site and drop site are decoupled in that they only know a data transfer protocol; the drag site does not know what the drop site will do with the data received as a result of the transfer.

This section explains how users are expected to interact with drag-enabled objects, with examples describing how drag transfer is performed between the Integrated C4I System Framework (ICSF) and the Extensible Information System (XIS). ICSF generates a chart (map) view of the Common Operational Picture (COP) using data produced by the Track Management System (TMS), and XIS provides the capability to create other views of these data. While the examples focus on how drag transfer is performed between these two applications, the requirements are generic and applicable to any user interaction with drag-enabled objects in a COE-based system. The requirements can also be extended to other methods of object transfer (e.g., clipboard transfer via cut/copy/past commands, transfer via import/export commands) supported by the system.

15.1.1 Supported Transfer Actions

When users transfer an object from a drag site to a drop site, the default operation is a drag copy; a drag move is supported only when allowed by and appropriate to the drag site (e.g., the action does not interfere with filter settings in effect at the site). Users are able to perform the transfer using either direct manipulation (e.g., drag and drop) or indirect manipulation (e.g., selecting a Drag menu option, using a shortcut key). Users are able to specify the attributes of default transfer behavior (e.g., as a user preference setting) so they can customize the behavior to meet their needs. Possible default behaviors include executing the user action without prompting, displaying a dialog window with options, displaying a “cannot” pointer if any part of the drag is invalid, and displaying a “cannot” pointer only if all of the drag is invalid.

Section 3.4.2 explains how drag transfer is integrated with object selection, the visual feedback provided during a drag transfer, and the options available to cancel or undo a transfer.

[1] Section 16 describes object concepts and provides an overview of object-oriented design.

Downloaded from http://www.everyspec.com

15.1.2 Drag Site and Drop Site Responsibilities

Users are able to transfer an object in a view to another view if the object is drag-enabled. Unless authorized by the data producer at the drag site, the transfer of an object affects only its visualization and does not transfer its data from one database to another or transfer ownership of the data from the drag site to the drop site, even if the representation of the object changes as a result of the transfer. For example, an object in the TMS database might be plotted as a symbol in a chart view, as shown in figure 15-1. Dragging the object from the chart view to an XIS table view would cause the object to be visualized as a record in the table but would transfer ownership of the object from ICSF to XIS only if allowed by the TMS data producer.

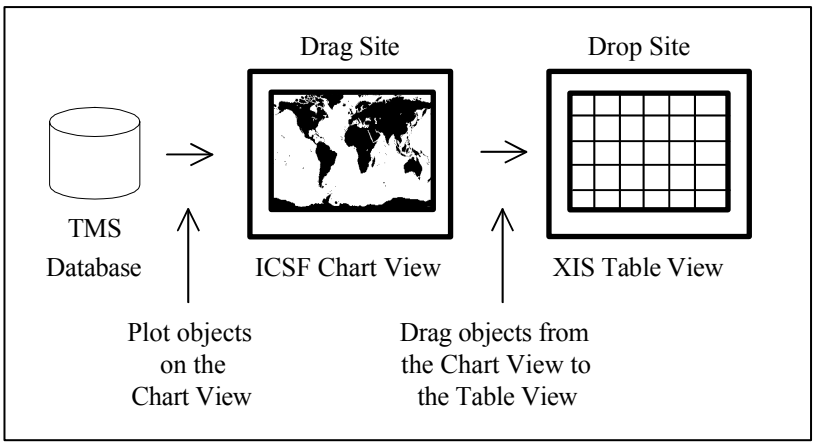


Figure 15-1. Dragging objects from a chart view to a table view in COP.

If a drag site contains objects that are not drag-enabled, it provides feedback to users if they attempt to transfer these objects. If some of the objects being transferred are drag-enabled, the drag site allows the drag to occur but notifies users that not all of the objects can be transferred to the drop site. The feedback includes reasons why the transfer failed if this information is available. If none of the objects is drag-enabled, the drag site notifies users when they start the drag operation and does not allow the transfer to occur.

The data producer at the drag site determines what data about an object can be transferred to a drop site and provides users with the capability to identify the transferable data for an object. For example, the data producer might flag these data in the property inspector for the object.

The drag site supports the transfer of any visual representation of an object that can be selected by users. For example, users can transfer an object by dragging its symbol in a chart view or its record in a table view. This action transfers all of the data that is available for the object. The drag site sends the data for an object made available by the data producer, while the data consumer at the drop site determines which of the data it can accept.

Downloaded from http://www.everyspec.com

The data consumer at the drop site determines what transferable data for an object should be available at the site. In addition, the data consumer determines whether or not to accept an update to the data for an object as changes occur at the data source. When a change to an object occurs, the default behavior is to update the object at the drop site. If a drop site maintains a static snapshot of the data for an object, users are informed and allowed to select between static and dynamic options as appropriate.

Downloaded from http://www.everyspec.com

15.2 Drag Transfer Results

15.2.1 Representation of Dropped Objects

A dropped object is rendered in the visual representation currently active at the drop site. For example, objects are visualized as records when dropped into a table view and as symbols based on the current plug-in when dropped into a chart view. The data consumer at the drop site is responsible for providing an appropriate visual representation for all objects that are transferred successfully.

The visibility and appearance of a dropped object are defined by the display settings in effect at the drop site. As a result, it is possible for an object to transfer successfully to a drop site but be hidden from view based on the filters that are currently active. The drop site is responsible for providing feedback (e.g., updating any visual cues about hidden objects) following the transfer so users can determine if the operation was performed correctly.

Users are able to change the appearance of a dropped object using the display controls provided by the drop site. For example, if users drag an object from a table view to a chart view, they are able to manipulate the appearance of the object using the display controls in the chart view and not those in the table view. Users do not need to know the "source" (native or dropped) of an object in order to identify the site where changes in object appearance are made.

It is possible that users might modify the appearance of an object prior to dragging it to the drop site and want to retain the user-defined attributes (e.g., object color) rather than reverting to the display settings in effect at the drop site. The drop site is able to support both options, with users able to select between them as appropriate.

A dropped object has the same appearance as an object that is native to the drop site. For example, the same MIL-STD 2525 symbol is used to display an object if it is plotted in a chart view by ICSF or if it is dragged into the chart. The drop site provides users with the capability to identify dropped objects and, if desired, distinguish among them based on the data source (e.g., originating database). Options include adding an indicator to a symbol for a dropped object and displaying a tree view of dropped objects organized by source. In addition, a dropped object and a native object respond in the same manner to changes in appearance as users modify the display settings at the drop site. The drop site provides users with the capability to act on (e.g., sort, select, filter) dropped objects based on data source.

When an object is dropped into a chart view, it may be stacked on top of a native object if the dropped object has the same coordinates as the native one. The drop site allows users to interact with an object that may be hidden by another object. For example, the drop site might support drill-down selection so users are able to select any or all of the objects at the pointer location and do not have to zoom the chart or apply declutter in order to interact with an obscured object.

Downloaded from http://www.everyspec.com

Transferring an object to a drop site where the object is already present does not create another instance of the object at the site. For example, users cannot drop an object into a chart view if the object is already present there (even though it may not be visible). If users drag a group of objects to a drop site and some of the objects are already present at the site, the drop site allows the transfer to occur but informs users that not all of the objects were transferred.

It is possible that users might want to have multiple instances of the same object at a drop site (e.g., to use as a template from which to clone copies or variations of the object for a planning or simulation scenario). The default behavior is to not allow duplicates at the drop site, but the option to drag multiple instances of an object to a drop site is available to users as appropriate.

The drop site determines whether or not to make the data for a dropped object available outside the site. Availability can be global (if the data are available on a wide-area network to all users), local (if the data are available locally to a subset of users), or personal (if the data are available only at the workstation on which it is generated). When an object is transferred to a drop site, its initial availability is personal (i.e., limited to the drop site where the transfer occurred). Following the transfer, a dropped object can be made available in the same manner as an object that is native to the drop site. A dropped object is automatically available outside the drop site at the time of the transfer only if agreed to by both the data producer and consumer.

15.2.2 Interacting With Shared Objects

15.2.2.1 Shared Object Behaviors

Once an object is transferred from the drag site to the drop site, different views of the same object are displayed simultaneously. The following object behaviors are shared between the views available to the user (i.e., at the workstation where the transfer occurred):

- A change in the select state of an object in a user view is shared by all user views containing the object. If desired, users can be provided with the option to share the select state among a subset of views as appropriate.

- A change in the visibility of an object in a user view does not affect its visibility in any other user view containing the object.

- A change in the appearance of an object in a user view does not affect its appearance in any other user view containing the object.[2]

[2] Changing how an object attribute is visualized differs from changing the visualization by changing the attribute. An example of the former is changing object color from blue to red, which affects only the user view where the change was made. An example of the latter is changing the threat assigned to an object

Downloaded from http://www.everyspec.com

The data producer retains ownership of an object that is present in multiple user views and determines whether the data consumer in a view can initiate any action to change the object.

15.2.2.2 Pop-Up Menus for Shared Objects

An object in a user view has a pop-up menu that provides access to frequently executed actions for the object. This menu includes an option to open a property inspector that displays data about the object. Any other menu options dealing with shared behaviors follow the rules described above. If the menu includes options to change the visibility and appearance of the object, the options apply only to the object in that view. The menu includes options to create, edit, and delete the object only if these actions are allowed by the producer of the object.

from friend to hostile, which impacts the visualization (i.e., changes object color from blue to red) by changing an attribute of the object and affects all user views containing the object.

Downloaded from http://www.everyspec.com

16. Object-Oriented Design

16.1 Object Concepts

16.1.1 Object Types

An object is a unit of information that is represented visually. An object possesses attributes that define its appearance or state, and it has behavior (i.e., the information it contains can be viewed and manipulated). Objects can be classified into the following types based on the function performed:

- A data object provides access to information for display or editing. A data object may be a composite object; i.e., it may be composed of or contain other objects. For example, a newsletter composed of text, graphics, and photographic images is a composite document. A composite object shares the characteristics and behaviors of both data objects and container objects.

- A container object stores and groups objects together. A container can contain other containers; for example, the desktop is a general-purpose container that serves as the user's main working environment and provides access to folders that store files as well as other folders.

- A device object allows users to interact with the computer-related objects in their work environment. A device object may have characteristics and behaviors of a data or container object. Device objects such as a trashcan or printer are normally defined at a system level and available to users on the desktop.

16.1.2 Object Views

A view presents information about an object and provides the means for users to access and modify the information. Object views are presented in windows, with different views presenting information in different formats. There are four types of object views:

- A composed view presents the information in an object in a form that shows its order and relationship to other components of the object. For example, the normal view of a document presents information in a format for editing, while the page view presents information in final print format. A composed view is the primary view for a data object. Changing the order of information within a composed view changes the meaning of the object; e.g., moving text from one part of a document to another changes the meaning of the document.

- A contents view displays information about the contents or components of an object. For example, the contents view of a folder presents information (e.g., name, size, date created) about the files it contains. A contents view is the primary view of a container object and may also be available for other types of objects (e.g., a composite object). A contents view can be presented in icon, list, or tree formats. Rearranging the order

Downloaded from http://www.everyspec.com

of information within a contents view does not change the meaning of the object; e.g., the set of files in a folder remain the same whether viewed in icon or list format or sorted in alphabetic or file size order.

- A properties view displays information about the characteristics or attributes of an object. For example, the properties view of a document provides information about text attributes such as font, type size, and color. The properties can be display-only, or users can be allowed to change them. A properties view is provided for all types of objects.

- A help view presents information to assist users in working with an object. This view provides information about the components of an object or about the object as a whole and is normally available from a menu bar or a pop-up menu. For example, when a text box has focus, SHIFT+F1 displays context-sensitive help about the information to be entered in the text box.

16.1.3 Object and Shortcut Icons

An object icon is the pictorial representation of an object. The appearance of the icon identifies the object and provides an indication of its state. The appearance of an object changes when an object is selected, when it is unavailable for selection, and when it is in use (e.g., when a view of an object is opened or a task that uses an object is invoked).

Multiple instances of an icon may be available for an object; i.e., the icon for an object may be supplemented by one or more shortcut icons. The shortcut provides a link to the object and has the same “look and feel” as the original. Users can interact with either the original object or one of the shortcuts.

Downloaded from http://www.everyspec.com

16.2 Application Design

Section 10 describes the design of an application-oriented user interface. While an object-oriented interface makes use of the same interface components as an application-oriented one, the two approaches differ in the model underlying the interface. Table 16-1, taken from Mandel in The Elements of User Interface Design, identifies these differences.

Table 16-1. Attributes of application-oriented and object-oriented user interfaces.

Application-Oriented User Interfaces	Object-Oriented User Interfaces
Application consists of an icon, primary window, and secondary windows.	Product consists of a collection of cooperating objects and views of objects.
Icons represent applications or opened windows.	Icons represent objects that may be directly manipulated.
Users must start application before working with objects.	Users open objects into views on the desktop.
Containment is mainly shown using text list boxes.	Folders and notebooks are visual containers.
Provide users with function needed to perform a task.	Provide users with supplies needed to perform a task.
Focus on the main task as determined by the application.	Focus on inputs and outputs for objects and tasks.
Related tasks supported by other applications.	Related tasks supported by use of other objects.
Rigid structure---by function.	Flexible structure---by object.
Users can get trapped in a task.	Users should not get trapped in a task.
Training is centered on the application and its function.	Training should focus on common paradigms, look and feel.
Users must follow structure of the application.	Users may perform task in their own way or innovate.
Many applications required---one per task.	Few objects---more reuse of the same objects in many tasks.

An application[1] may implement some elements of an object-oriented user interface (e.g., rely on objects for performing certain functions), or it may be designed entirely in accordance with object-oriented principles. If an application supports any elements of an object-oriented interface, it does so in accordance with the specifications provided here. [2]

16.2.1 Object Model

16.2.1.1 Objects

In an object-oriented interface, users interact with the collection of objects they need to perform a task, rather than launching and running individual applications, the latter being

[1] This document uses the term "application" when referring to the software needed to perform a set of related tasks, without regard to whether the software implements an application- or object-oriented approach in its user interface design.

[2] In describing attributes of an object-based design, this section sometimes reiterates specifications contained in or makes reference to other parts of this document.

Downloaded from http://www.everyspec.com

hidden unless users choose to see them. Different kinds of objects are available, each containing properties and supporting behaviors that are appropriate for the information represented by the object. Sets of objects are organized into classes, each identified by its own set of properties and behaviors, that help define relationships among the objects. A hierarchy of object classes is defined, with the subclass inheriting some of its properties and behaviors from its parent class but also containing new properties and behaviors that identify its unique features.

Objects are presented to users either as icons or as components that are visible in a view of another object. Icons are provided for objects that need to be opened in their own windows. Users can choose the objects they need and then organize and group the objects for the tasks they want to perform. For example, users can create a hierarchy of folders and then place objects and/or shortcuts within the hierarchy as desired. In addition, users can decompose objects into their component objects as needed to perform a particular task.

16.2.1.2 Views

An object supports various views of its contents, each showing different aspects of the object. The icon for the object is itself a view that allows users to interact with the object as a whole. Other views are displayed in windows and allow users to examine information about the object and interact with it. Users choose an object to work with, then choose actions to apply to the object. These actions are available from pop-up menus associated with the icon for the object, from direct manipulation (e.g., drag transfer) of individual objects, and from commands available (e.g., in menus) within the view of the object. Attribute or state information is provided in the property view of an object and may be available for manipulation by users.

An application provides composed and property views for each data object, and contents and properties views for each container object. An application can support up to eight different composed views of a data object depending on the function or nature of the object.

Users are able to access multiple concurrent views of an object. Each view supports a specific set of tasks, with the information displayed and actions available tailored to the specific task being supported. If multiple views are available for an object, users should be able to select the one that opens as the default. If users execute the action to open the default view and this view is already open, the action can either raise the view to the top of the window stack or open a new view of the object in a separate window.

Users are able to open any view of an object from its icon on the desktop. Double clicking BLeft on the icon opens the default view of the object; clicking BRight on the icon activates a pop-up menu with options that open other views of the object. Once an object view is open, users can access the other views of the object (e.g., from a View menu). When a new view is opened, it may replace the previous view in the window, or users may be given the option to open the view in a separate window.

Downloaded from http://www.everyspec.com

The information in an object does not change when a new view of the object opens. Similarly, the information in an object does not change when a different view is selected, when the window in which the view is displayed is minimized and then reopened, or when a desktop session is ended and then restarted.

Multiple instances of the same view of an object can be displayed at the same time, and different views of the same object can be displayed simultaneously. In both cases, changes made to the content of one view are applied immediately to all other views of the same object. If multiple views of an object are displayed simultaneously (e.g., in different windows), the changes made to the content in one view are visible in the other views. For example, selecting the name of a unit in a contents view of tactical objects also highlights the symbol for the unit in a map view of the objects. However, changes made to the format of a view (e.g., changing from an icon to a list format) are local and do not affect other views of the object.

16.2.1.3 Object and Shortcut Icons

The icons in an application should be designed to show important characteristics of the objects they represent and provide visual cues that help identify object classes and the relationships among them. The characteristics included in the image may differ depending on icon size, with the graphic used in a small icon showing fewer characteristics than a full-size icon so they can be easily distinguished by users. Depending on the function performed by an object, users may be allowed to customize the graphic and/or edit the label for the icon.

Users are able to create shortcut icons for an object (e.g., by selecting a Create Shortcut option in the pop-up menu for the object icon). The graphic for a shortcut icon contains a unique visual feature (e.g., a link symbol) that distinguishes it from the icon for the original object. Shortcut icons can be placed in folders or directly on the desktop.

Changes made to the content or properties of an object are reflected in all shortcuts for the object, regardless of where the icons are located. Similarly, changes made to any shortcut for an object are reflected in the original object and all of its shortcuts. Double clicking BLeft on a shortcut icon opens a view of the object, in the same way that would occur if the action had been performed on the object itself. If an object is deleted, all shortcut icons for that object are also deleted; however, if a shortcut is deleted, the original object and other shortcuts to it are not deleted.

16.2.2 User-Computer Interaction

User-computer interaction in an object-oriented design is based on an object-action paradigm, as in an application-oriented interface. An application supports both pointing device and keyboard methods for navigation, selection, and transfer, as described in section 3. The methods available are consistent across object classes and appropriate to the type of view being presented.

Downloaded from http://www.everyspec.com

16.2.3 Window Design

Window types in an object-oriented design are based on the type of information displayed in the window. Primary windows are used to display the view(s) of an object, dialog windows to specify actions related to an object, and message windows to provide information that may require user intervention. Primary and secondary windows are organized into families, with the same parent-child relationships and the same window management functions as in an application-oriented design. Secondary windows are modeless, unless it is impossible for users to continue until they have responded to the window.

16.2.3.1 Window Titles

The title of a primary window contains the name of the object and the view presented in the window. The following guidelines apply to window titles:

- In a composed view, the name of the view should describe the view itself (e.g., Normal, Outline) and not contain the term "Composed View."
- In a contents view, the name of the view should describe the view format (e.g., Icon View, Tree View).
- In a properties view, the name of the view should include the term "Properties."
- If the window displays a composite document, the title of the window should be the name of the document; the title should not change when different objects within the document receive focus.

16.2.3.2 Menu Bars

If the menu bar in a primary window contains any of the following common menus, they are ordered: Object, Selected, Edit, View, and Help.

Object menu. The Object menu contains options that affect the object whose view is currently displayed in a window. The title of this menu is the class name or actual name of the object.

Selected menu. The Selected menu contains options that affect the object(s) currently selected in a window. When an option is activated, it is applied to all of these objects. The contents of this menu can change depending on the object(s) selected in the window. These changes should be made by adding/removing options, rather than by managing their state (i.e., showing them as available/unavailable as appropriate). If a Selected menu is not included in the menu bar, a Properties option can be included in an Edit menu.

Downloaded from http://www.everyspec.com

Edit menu. The Edit menu contains options that modify the data in a window, as in an application-oriented design. Section 8.1.1.1 describes the options included in this menu.

View menu. The View menu contains options that change the presentation of the current view and switch to other views of the object if more than one view is available. The names of the views can be listed in the menu itself (and delimited by separators from other options in the menu), or the menu can include an option (e.g., Change View) that parents a submenu listing the names of the views. In both cases, visual cues are provided that identify the current view shown in the window; e.g., the option can be displayed as unavailable or as a state toggle (i.e., preceded by a check mark).

As in an application-oriented design, the options in the View menu affect the view of the object but not the underlying content of the object. When users activate a menu option that opens a different view, it may be displayed in the same window or in a new window. If multiple composed views of a data object are available, users can access them in the View menu. If multiple contents views (e.g., in list or tree format) of a container object are provided, this menu includes options for users to select the detail information to be presented and the order in which it appears in the window.

Help menu. The Help menu contains options that provide access to online help for the object(s) in the window. Section 8.1.1.1 describes the options included in this menu.

16.2.3.3 Toolbars

If the window includes a toolbar, its contents apply to the object whose view is presented in the window. The availability of individual toolbar buttons is linked to the corresponding option or control in the window. Existing buttons may be replaced or new buttons added to the toolbar when objects within the window are selected (i.e., receive focus).

16.2.3.4 Controls

Unless otherwise indicated, controls have the same appearance and behavior and follow the same window design guidelines when available in an object-oriented design as in an application-based one. An application complies with specifications contained in sections 6, 7, and 8 with regard to the implementation of these interface components.

16.2.3.5 Pop-up Menus

An application provides a pop-up menu for each object with which users can interact. The menu contains frequently executed actions for the object; options pertaining to the views available for the object should be presented first, followed by data transfer and utility options, and then by application-specific options.

Downloaded from http://www.everyspec.com

This page intentionally left blank.

Downloaded from http://www.everyspec.com

17. User Interface Internationalization

17.1 Operating With Non-US Languages

Internationalization is the process of generalizing software so it can handle multiple languages (i.e., locales) and cultural conventions without the need for re-design or re-compilation. If an application designed for a US audience will be used in combined or coalition warfare operations, it needs to provide a user interface that matches users' expectations, interacts with users in their native language, and displays data in a manner that is consistent with users' cultural conventions. The guidelines in this section are provided to assist developers with a requirement to provide internationalized software and are not considered in determining COE style compliance.

17.1.1 Character Rendering

Languages can be categorized in terms of the characters or symbols in which they are written. To facilitate computer processing, a character set is defined for each language to contain its written letters, numbers, and punctuation marks, with each character in the set represented by a binary value. Most European languages, including English, are based on the Roman (or Latin) alphabet. Because these languages contain fewer than 200 basic characters (i.e., the 26 letters in the alphabet, with upper case, lower case, and accented variations), their character sets can be encoded in a single-byte. ASCII (American Standard Code for Information Interchange) is the most commonly used single-byte code set for representing English-language text for an American user base. However, because ASCII encodes a limited number of characters, it is insufficient for representing text in languages other than American English.

While most European languages are based on the Roman alphabet, many of them contain extended characters (i.e., ones that do not exist in English and are not available in ASCII) in their character sets. These characters include accented vowels such as é and ê; characters such as the French ç, the Spanish ñ, and the German ß and ü; and combined characters such as æ. In addition, some European languages may not use the entire Roman alphabet; Italian, for example, lacks the letter k. Despite these variations, all text in Roman-based languages is written from left to right, with each new character appended to the right of the previous character. Furthermore, the appearance of a character and its order within a character sequence do not change as new characters are entered.

While languages with fewer than 200 characters can be encoded in a single byte, complex languages such as Chinese, Japanese, and Korean contain several thousand unique ideographic symbols from which words are composed. Encoding such a large character set requires two (or more) bytes per character rather than one. Unicode provides the capability to represent multi-byte character sets and is the preferred approach for encoding the alphabets, ideograph sets, and symbols for all languages in the world. In multi-byte languages, there may be no "natural order" to characters vis-à-vis sorting, and no concept of or distinction between upper-case and lower-case forms. Because

Downloaded from http://www.everyspec.com

characters are composed of many strokes, text in these languages may require more space to display, and text entry may be a more complex process than in single-byte languages

In most Roman-based languages, each character is rendered as a separate symbol of fixed shape, and characters are written in the approximate order in which they are pronounced. In other languages, however, the way in which characters are rendered graphically depends on their linguistic context. In Arabic, for example, a character may be displayed in several different forms, depending on whether it is displayed alone or as the first, middle, or last character in a word. In contextual languages, the characters that make up a symbol may be entered in several different orders, and entering a new symbol may change or even eliminate a previously entered symbol. As a result, complex algorithms may be needed to manage text line length (e.g., line breaks, justification), support editing individual characters within a symbol, and provide search and sort features that can recognize multiple encodings of the same symbol.

Most languages are unidirectional; i.e., lines of text are presented uniformly from left to right or from top to bottom. Although Asian languages such as Chinese, Japanese, and Korean may present horizontal and vertical text on the same page, they are considered to be unidirectional because they do not mix directions in a single line of text. By contrast, Arabic and Hebrew are bi-directional; text in these languages is written from right to left, but numbers and foreign words in the same text are written from left to right. Because the direction of text entry may change from one character to the next, appropriate text handling procedures must be available in both right-to-left and left-to-right text.

17.1.2 Structural Rules for Character Handling

An application is able to accept and process all of the characters in the character set used by the target language (i.e., the one to which the application is being converted). Because languages differ in their structural rules for character handling, assumptions made when processing a US character set may be inappropriate or inaccurate when applied to a language with extended characters. In particular, a US application is likely to require modification in order to correctly handle case conversion, ligatures, special characters, and word and character boundaries in the target language.

Case conversion. In US software, case conversion is usually performed by adding or subtracting a constant (i.e., 32) to or from the ASCII code for the character. In extended character sets, case conversion is more complicated because there is no constant difference between the numerical equivalents for upper- and lower-case representations of characters. In addition, the distinction made in English between upper-case and lower-case letters may be ambiguous or not exist at all in other languages. For example, Chinese, Japanese, and Korean have no case distinction. In other languages, an accented vowel in lower case may retain its accent in upper case, or the accent may disappear; alternatively, a vowel in upper case may or may not contain an accent in lower case, depending on the word and the language.

Downloaded from http://www.everyspec.com

Ligatures. Ligatures are sequences of characters that are treated as a unit; e.g., æ is a combination of a and e. In some languages, ligatures can be entered as a single character or as two separate characters. In the latter case, both letters would be capitalized in words that are proper nouns; for example, Iceland is written as IJsland in Dutch. Ligatures occur frequently in contextual languages such as Arabic. A US application may require revision in order to handle any ligatures that occur in the target language.

Special characters. Because languages differ in the meaning assigned to special characters, a US application that uses characters such as apostrophes as delimiters in a text string and restricts their use to this function may require modification when converted to certain European languages. For example, French and Italian replace the terminal vowel in an article by an apostrophe when the following noun has an initial vowel. In addition, some languages include special characters that may not be present in English or use these characters in ways that differ from US usage. For example, Spanish starts exclamations and questions with inverted exclamation mark and question mark characters, while French includes a space between the last word of a sentence and a concluding exclamation mark or question mark. Finally, diacritical marks (i.e., the signs modifying the value or sound of characters) may have different meanings in different languages. For example, certain diacritical marks specify the doubling of consonants in Arabic but may indicate pitch in Vietnamese.

Word and character boundaries. A word consists of a string of characters preceded and followed by delimiters. In a US application, these delimiters are usually blanks or spaces but may also include the unused portion of the Roman character set. This latter approach can be problematic when converting to a Roman-based language where these characters need to be interpreted as part of the word and not considered as delimiters. In addition, in some languages, a blank is acceptable as a numeric or phrase separator and so would not be appropriate to use as a standard word delimiter.

Downloaded from http://www.everyspec.com

17.2 Text Translation

17.2.1 Creating Internationalized English Text

The process of translating the text displayed by a US application begins with the creation of an "internationalized English" version of the text. All of the text is reviewed and, if necessary, modified to ensure that it is easy to understand and use. Message text (e.g., in message windows, online help) is presented in short, simple, declarative sentences whenever possible. Excessive use of subordinate and coordinating phrases is avoided, and ideas are expressed as concretely as possible. Ambiguous language, humor, jargon, and cryptic messages are likely to cause difficulty for non-US audiences and so need to be eliminated. Likewise, compound adjectives, strings of nouns, long sentences with many ideas, and negative questions can be difficult to understand and so are not used. If an application needs to explain a series of concepts, they are presented in the form of a list, rather than in a text string separated by commas.

The content of each window in an application is checked for US-specific language prior to translation to minimize the likelihood of misinterpretation by the target audience. The goal is to use only those terms that are employed in the same way throughout the English-speaking world. The use of acronyms and abbreviations is limited since many are not recognized internationally and may have different meaning, depending on where they are used. Similarly, when large numbers are presented, they are written as numerals; for example, the term "billion" means one thousand million in the US but one million million in some European countries. To minimize confusion, the names of months are written out when they appear as part of a date. For example, 06/10/94 can be interpreted as June 10 or October 6, depending on the user's cultural background.

An application avoids presenting examples that may be uniquely American. A generic term is used, rather than what something is called in the US. For example, "stock exchange index" is an international term while "the Dow" is specifically American and "the FTSE 100" is specifically British. The messages in an application are reviewed to determine if they may be interpreted by the target audience in ways other than intended. For example, "as soon as possible" means "immediately" in the US but "when convenient" in other countries. Terms such as "left hand" can be offensive in some cultures and need to be replaced with "on the left" or "left side."

17.2.2 Translating Text and Messages

Once an internationalized English version of the text displayed by an application is created, it is translated into the target language. If appropriate, the translation is tailored to the target language in the specific country or region that will be using the application. Translated text contains proper technical terminology, especially when the terminology may differ from conversational expression in the target language. Care is taken to maintain distinctions in terminology that may be translated into the same text string in the target language. For example, Cancel and Undo are normally translated as "annulez" in French, even though these commands have different meanings in English. Similarly,

Downloaded from http://www.everyspec.com

spelling and grammar can differ among varieties of a single language and so need to be adapted accordingly.

The same terminology is used in both the user interface and documentation for an application. While the goal is to provide an accurate translation of all text into the target language, it is acceptable to use US words in the text if the target language does not have an adequate vocabulary of technical words or if the target audience is accustomed to dealing with US terminology.

The accuracy of translated messages is verified since it is possible for the text of one message to be the same as another, especially if the original messages were very similar or were worded ambiguously. In addition, the meaningfulness of the translation is checked against the situation that invoked it to ensure that the information being conveyed in the original message is also conveyed in the translated version.

Translated text is reviewed to ensure that it makes grammatical sense. Some applications construct messages from two or more substrings; for example, the name of the file being deleted is inserted into the text string requesting the user to confirm deletion of the file. While this approach may work in English, the linguistic characteristics (e.g., gender, word order, special characters) of the resulting text can be awkward or inappropriate when translated into the target language. When messages are constructed by nesting or concatenating strings, the translated text that results is frequently meaningless or syntactically impossible because words or phrases were not modified to fit the grammatical rules of the target language.

Translated text uses the same character set and font as the rest of the application. Line breaks and other format changes that may have been introduced with the translation are checked for accuracy since US rules for hyphenation, punctuation, or capitalization are likely to be different from those in the target language. If typographic variations such as italics or boldface have been added as part of the translation, they are checked to ensure that they are suitable in the target language. Language environments have evolved unique rules defining how elements such as title lines, bulleted lists, and footnotes are used to distinguish among levels of expression and to indicate how expressions are related. The appearance of the translated text is adapted as needed to satisfy these rules.

17.2.3 Translating Documentation

The documentation for an internationalized application describes a representative sample of the internationalized capabilities provided by the software. For example, an explanation of how a sorting function works describes the kinds of sorts that are performed, explains that the current locale affects the output, and provides several examples that are representative of the locales supported.

Documentation text contains simplified English. Whenever possible, a single term is selected to express a concept, and the use of synonyms is minimized. However, these changes are made in such a way as to not reduce the precision of the text, create awkward

Downloaded from http://www.everyspec.com

phrasing (with an increase in overall text length), or produce unacceptably dull or boring text. References (e.g., to sample users) and examples that are specific to one culture are modified to be more international in focus. Any graphic symbols used in the documentation are reviewed to minimize the extent to which they are culture-specific. If necessary, a table is provided that lists the symbols and their interpretation. Finally, documentation sections such as glossaries and indexes are expanded as needed to help non-US readers find information. For example, glossaries define words that may have a different technical meaning or not exist at all in the locales supported by an application.

Because the order of items in a sorted list usually changes following translation, references to the position of items in the list are removed from documentation. Similarly, when collation sequences are described, the results are not described as sorted "alphabetically" since ideographic languages cannot be sorted alphabetically. Instead, output is described as appearing "in sorted order as determined by the current locale." Other changes needed when internationalizing documentation include ensuring that terms such as ASCII, text, byte, and character are used appropriately, describing any assumptions made about date and time formats, replacing references to Yes and No (e.g., when describing actions in response to a message) with words that are appropriate to the locale, and presenting the names of any individuals (e.g., sample users) in an order that is correct for the specific culture. Finally, lengthy text explanations (e.g., in online help, training materials, or other documentation) may need to be restructured or reorganized so they follow the rules used by the target audience in organizing technical discussions or sequences of explanations.

Downloaded from http://www.everyspec.com

17.3 Text Input Methods

17.3.1 Keyboards and Keyboard Input

A workstation usually has a keyboard layout tailored to the target language, and this layout may be different from the one available on US workstations. Conventions concerning the location of characters vary from language to language and sometimes from country to country within the same language. For example, the German keyboard reverses Z and Y from their positions on the US keyboard, and the Spanish keyboard has a different layout in Spain than in Latin America. In addition, languages may add, omit, or change the characters on a keyboard. For example, the British keyboard contains the currency symbol for pound instead of #, and the Spanish keyboard has ñ where the US keyboard has L. Non-US keyboards may mark each key with up to four different characters. Users press modifier keys (e.g., SHIFT and/or ALT) in combination with the key to enter the various characters marked on the key.

Because computers respond to specific physical keypresses regardless of what markings appear on the keys, a different keyboard may not be required when converting a US application into another language. A keyboard can be adapted by replacing the symbols on each key, either with adhesive labels or new key covers. An application then maps the individual keystrokes to the character set for the other language and displays the appropriate characters. If this approach is used, the function keys on the keyboard need to be mapped to the same actions as in the original software, and any messages generated when these keys are pressed are displayed as they were prior to the conversion.

Languages where diacritical marks are used extensively (e.g., French) usually provide keyboards that allow users to generate characters with these marks with a single keystroke. However, because English has very few accents, users with a US keyboard have to execute a combination of keystrokes in order to enter an extended character. An application can use “dead” keys or a compose-based method to produce this type of input.

With “dead” keys, the keystrokes consist of a “dead” (i.e., nonspacing) key, followed by the character (e.g., a vowel) to be displayed with an accent. A different dead key is assigned to each accent. When a dead key is pressed, a text input mode is invoked; the symbol on the key is not displayed, and the text cursor does not move. The mode is automatically disabled following the next keystroke; the appropriate dead key is pressed each time an accented character is entered. If an invalid character (e.g., a consonant) is entered, the character is displayed without an accent, and feedback (e.g., a beep) is provided to indicate that the keystroke was invalid.

In a compose-based input method, when a predefined control key is pressed, a text input mode is invoked that forms the next two keystrokes into a single character. When the first character (e.g., a vowel) is typed, nothing is displayed by an application. When the second character (e.g., the diacritic) is entered, the completed character is displayed, and the input mode is automatically exited.

Downloaded from http://www.everyspec.com

17.3.2 Approaches to Text Entry

Text entry using pre-edit methods. In most languages, users perform text entry by typing directly into a text box. However, if a keyboard cannot produce all of the symbols in a target language, a pre-edit step may be needed. Users type characters from the keyboard, usually into a pre-edit area, and then execute an action to convert the characters into other symbols appropriate to the language. These symbols are then displayed in the text box.

When a pre-edit step is required, text entry can be performed on-the-spot, over-the-spot, or off-the-spot. On-the-spot means that as users type, the characters appear directly in the text box which can contain both text in unconverted form and converted symbols. Although more difficult to implement, this approach is preferred because it is more similar to text entry as normally performed by users. In over-the-spot, a separate pre-edit area is provided for each text box; when users convert their input into final form, the symbols are displayed in the appropriate text box. Off-the-spot also provides a separate pre-edit area but uses the same area for multiple text boxes; in this case, when users convert their input into final form, the symbols are displayed in the text box that has input focus.

When text entry includes a pre-edit step, an application provides feedback concerning the status of the input after users enter text in the pre-edit area and then execute an action to convert the input into final form. If insufficient information is available to perform the conversion, an application can prompt users to enter more pre-edit text, present them with a list of choices from which to select, or indicate that the conversion has failed. If an on-the-spot approach is implemented, the text box provides a visual distinction (e.g., a different text font or color) between original input and converted text so users can easily distinguish between the two. If the pre-edit area is provided in a separate dialog window, the window is modeless so users are not restricted to only performing text entry.

Text entry in languages with large character sets. Several options are available to support keyboard input in languages that have large character sets. Whatever method is selected must be able to accommodate context-specific variations within the language as users perform text entry. With each keystroke, converted text changes as needed in order to create a new compound character or add a mark to a previous character.

With the first option, the component elements of each character are marked on the keyboard. As users press individual keys, the elements are displayed. When a character is complete, it is displayed in place of its components. This method has been used to perform text entry in Chinese and Korean.

With the second option, users enter each character phonetically, and the phonetic form is automatically translated into the correct character. When more than one character has the same pronunciation, users are presented with a list of phonetically similar characters from which to choose. For example, users enter a root or radical character from the

Downloaded from http://www.everyspec.com

keyboard, then select additional strokes to complete the character from a set displayed by the application. This method has been used to convert Roman characters to Chinese ideographs, and Hiragana and Katakana characters to Japanese Kanji.

With the third option, users type the decimal or hexadecimal encoded value for a character or select the value from a list. If the value matches an entry in the code set, the corresponding character is displayed by the application.

Text entry in mixed character sets. Users may need to perform text entry in more than one character set (e.g., English and Korean) or in multiple locale-specific character sets (e.g., Kanji and Katakana). This flexibility can be provided by defining text input modes in each character set, along with a special keyboard character that allows users to toggle between the character sets as desired. Users with a keyboard where two character sets are marked on the keys select one of the modes to begin text entry. All of the typed text is interpreted in this character set. When the special character is encountered, the text mode toggles to the other character set and all subsequent input is interpreted in this set.

Text entry in bi-directional languages. Because bi-directional languages write text in both right-to-left and left-to-right directions, text entry may be performed in either direction, depending on the contents of a text box, and may require input in both directions within a single box. An application can provide automatic handling of directionality based on the characters being typed, or it can support an input mode that users invoke to switch the language and direction of text entry.

17.3.3 Other Text Entry Actions

The text cursor remains visible during text entry to indicate the locus of typed input. In addition, the text cursor does not disappear from view as it moves from one character to the next in a string of single-byte and multi-byte characters. If the target language is bi-directional, an application can support multiple text cursors within a single text area, one indicating when text can be added in the current input direction and the other marking the last place where the direction of input changed. In contextual languages (i.e., where the appearance of existing text can change as new characters are entered), the text insertion point can move backward or forward as users perform text entry; in this case, the text cursor is displayed in a manner that is consistent with the movement of the text insertion point.

The arrow keys move the text cursor in the direction of the arrow regardless of the direction in which text is currently being entered. DELETE deletes text in the direction opposite to the direction in which text is being entered.

If an application is being converted to a contextual language, it needs to define how certain keystrokes affect a compound symbol that is composed of several separate characters. For example, an application needs to determine when DELETE cancels the previous keystroke (i.e., removes a character) or deletes the entire symbol. In addition, an application may need to limit the ability to insert or delete individual characters in a word

Downloaded from http://www.everyspec.com

if these actions would change neighboring characters or alter the appearance of the word in unintended or confusing ways.

Downloaded from http://www.everyspec.com

17.4 Internationalizing User Interface Features

17.4.1 Text Expansion

When text is translated into another language, the result is often longer than the original English. For example, the phrase "message pop-up" translates to "Nachrichtenüberlagerrungsfenster" in German and "janela de sobreposiçao de mensagem" in Portuguese. The increase in text length may be as much as 200 percent, depending on the length of the original text. Some of this increase may result from the addition of spaces that were not present in the original text. Table 17-1 lists allowances for expansion recommended by MS Windows based on text length in English. This table refers to the number of characters in a message, with characters in multi-byte languages (e.g., Japanese) taking two bytes per character.

Table 17-1. Allowances for text expansion.

Length of English Text	Additional Space Required
Up to 10 characters	200 percent
11 - 20 characters	100 percent
21 - 30 characters	80 percent
31 - 50 characters	60 percent
51 - 70 characters	40 percent
Over 70 characters	30 percent

Note: This table was taken from the Microsoft Windows Software Development Kit -- Additional Windows Development Notes, as published in Software Internationalization and Localization: An Introduction.

Translated text may require adjustments in the horizontal spacing between specific pairs of characters. For example, in English, the characters f and i look better when displayed closer together than other pairs of characters. The horizontal spacing algorithms used by an application need to accommodate adjustments in the spacing of non-US characters, including pairs of characters (e.g., æ) that may be part of an extended character set.

The height of a line of translated text may be twice the height of the text in English. Roman-based languages may supplement the character set with diacritical marks that extend above or below the basic symbol. In non-Roman languages, marks may be stacked two or three high, and small versions of characters may be placed above, below, or beside the primary symbols, causing wide variations in the height of each text line. Because of their complexity, ideographs require more space to display the strokes within them. For example, some complex Chinese characters may need to be displayed at least 50 percent larger than alphabetic characters in order to be readable. The minimum size for ideographs is usually 16 x 16 pixels. Translated text may also require adjustments to the vertical spacing between lines to ensure legibility and readability when displayed by an

Downloaded from http://www.everyspec.com

application or printed. Extended characters, and in particular those with diacritical marks, may require additional spacing, especially when printed in upper case.

It is likely that the size and placement of controls in application windows will require adjustment in order to accommodate text expansion following translation. Menus and dialog windows will also need to increase in size in order to accommodate the longer text. Similarly, more vertical space may be needed in window components such as the title bar to accommodate larger character size, especially in languages such as Chinese, Japanese, and Korean. Finally, the size of the text included in the label of a window icon may need to increase to accommodate an extended character set, and the icon graphic may also contain embedded text that needs to be translated.

Each application window needs to be checked to ensure that all of the translated text fits properly within the window and that individual controls are positioned correctly within each window area. For example, when column headings are translated, they may be longer than the data they contain and need to be broken into more than one line of text. Similarly, when text labels are translated, the placement of the associated text boxes may be altered and require repositioning in order to be properly aligned within the window.

17.4.2 Nonlinguistic Text Features

Capitalization, punctuation, and word order. Text displayed by an application follows the rules for capitalization, punctuation, and word order used in the target language. For example, in German, all nouns are capitalized, regardless of their position within a phrase or sentence. Depending on the language, quotation marks may be displayed as “quotation”, «quotation», »quotation«, or „quotation.“ Interrogatory sentences in Spanish begin with an inverted question mark and end with a question mark in normal orientation. Adjectives precede nouns in English word order but may follow nouns in other languages.

Hyphenation. An application performs hyphenation in a manner that is consistent with the rules of the target language. These rules may call for changing the characters in a word when it is hyphenated at the end of a line, or placing hyphens between individual words when they extend beyond the end of a line. For example, in German, “drucken” and “heißen” become “druk-ken” and “heis-sen” when hyphenated; in French, a hyphen is added between a personal pronoun and “même” (e.g., “eux-même”) when these words extend beyond the end of a line.

Justification. The justification routines used by an application conform to the rules of the target language and may require some character-processing logic in order to do so. For example, in Asian languages where spaces are not used to delimit words, line breaks can occur anywhere within a word. However, because symbols are represented by a multi-byte character, line breaks cannot occur within a symbol nor can punctuation be the first character on a new line. Alternatively, languages such as Arabic and Hindi do not allow breaks within words. In this case, the justification algorithm used by an application must

Downloaded from http://www.everyspec.com

accommodate this restriction and be able to produce justified text without excessive space between words.

Abbreviations. The abbreviations used in US software may have different meanings in other languages or not be used at all. For example, while # is commonly used as an abbreviation for number, this character is not meaningful outside the US. The symbol @ means "at" in the US but "each" in the United Kingdom. The abbreviations for ordinals are 1st, 2nd, 3rd, etc. in the US, but 1°, 2°, 3° or 1^a, 2^a, 3^a in other languages, depending on the gender of the subject.

Typography. If an application displays text in a Roman-based character set, it supports the fonts (e.g., Times and Helvetica), sizes (e.g., 10-point, 12-point), and styles (e.g., regular, italic, bold) that are normally available in the target language. An application also accommodates any unique typographic conventions when displaying translated text. For example, stress in writing is indicated through the use of italics in English but by letter spacing or boldface in European languages. In Japanese, stress is indicated by underlining characters, putting a light gray background behind them, or writing the text in Katakana.

Reordering sorted information. If an application presents sets of related items (e.g., in lists, option menus) in alphabetical order, it reorders the items after translation based on a sort sequence that is meaningful to the target audience. The most appropriate order may vary by application and depend on the information displayed in the items. Section 17.4.6 provides additional information on sorting and collation.

17.4.3 Data Formats

An application is able to recognize and correctly handle the range of formats that are used to express data in the target language. The labels for all data entry areas are modified to include the appropriate unit of measurement. An application either converts the data format to one that is familiar to users or provides the capability to display data in alternate formats so users can select the one that is most meaningful to them. For example, US users prefer to measure length in feet and yards while European users are more familiar with the metric system. If the content of data entry areas is not converted to a format that is familiar to the target audience, then the data format is included as part of the data label. If an application supports converting to and from both millimeters and inches, the number of digits stored is sufficient to prevent truncation errors during conversion.

The presentation of date and time is modifiable by users so they can display this information in the appropriate time zone (e.g., India rather than Zulu) and modify it for other zones as needed. Numeric data is properly aligned according to the particular numerical separators and indicators used in the target language. In addition, if an application allows users to manipulate text, the different forms of tabulation available are modified as needed (e.g., allow the decimal tab to work with commas rather than periods) to accommodate the data formats used in the target language.

Downloaded from http://www.everyspec.com

Number systems and formats. While Arabic numerals (e.g., 0, 1, 2, etc.) are widely accepted, some languages have their own number systems. In some cases (e.g., Chinese), the symbols are substitutes for Arabic numerals, while in others (e.g., Ethiopia), there are special characters for numbers such as 10 and 100.

When presenting numbers, a comma, period, space, and apostrophe can be used as separators for units of thousands. In some cases, an explicit separator is not required for numbers less than 10,000. Numbers can be grouped by thousands or ten thousands. The period, comma, and center dot can be used as separators for decimal numbers. Positive and negative numbers can be indicated by + and - symbols appearing either before or after the number, and negative numbers can be enclosed in parentheses.

Measurement systems and arithmetic operations. US users are familiar with the Imperial system of measurement that uses inches and fractions of an inch (e.g., halves, quarters, eighths) while users outside the US rely on the metric system which measures in meters, liters, and grams. In addition, the US relies on the Fahrenheit scale for temperature while the rest of the world uses Celsius. Similarly, cultures vary in the manner in which certain arithmetic operations are performed. For example, some countries have rules for rounding numbers that differ from those used in the US. In addition, accounting rules (e.g., to calculate compound interest) vary from locale to locale.

Currency. The comma, period, and colon can be used as separators for currency. Currency indicators include a number of symbols (e.g., $, British pound, and the Japanese yen), alphabetic characters (e.g., FF, SFRs, kr), and combinations (e.g., CZ$), and can be placed at the beginning, middle, or end of the currency expression. There can be one or no space between the currency symbol and the amount, and currency symbols can be up to four characters in length. Most currencies (except Japan) include two digits to indicate fractional money amounts.

Date and time. The hyphen, comma, period, space, and slash can be used as separators for the day, month, and year, or separators can be left out altogether. In numeric date formats, the month and day fields can be reversed, and in some cases, the year field can come first. Month and day names can be capitalized or in lower case and can be abbreviated using the first two or three letters or some other combination of letters.

The manner in which a date is expressed can be affected by the calendar system being used. While dates are usually based on the Gregorian calendar, some cultures use lunar calendars or the Jewish or Arabic calendar or can express the date based on the year of accession of the Emperor, as in Japan. These calendars can include day names for more than seven days, and month names for more than twelve months. Moslem countries such as Saudi Arabia and Egypt use a calendar with 12 months but only 354 or 355 days. The first day of the week is Sunday in the US but Monday in European countries, a difference that affects the manner in which calendars are displayed.

Downloaded from http://www.everyspec.com

The colon, period, and space can be used as separators for hours, minutes, and seconds. The letter h can separate hours and minutes. Both 12-hour and 24-hour notation can be used. For 12-hour notation, a.m. or p.m. can appear after the time.

Although the world is divided into 24 standard time zones, countries have the freedom to set their own times. For example, in South America, Surinam's time is 30 minutes different from that of the next zone, and Guyana's is 45 minutes different. The same time zone can have multiple names, and different time zones can share the same abbreviation. Finally, countries differ in their rules concerning daylight savings time or may not use it at all, and the hemispheres differ in when it starts and ends because the seasons are reversed.

Addresses and telephone numbers. Addresses vary from two to six lines long and can include any character used in the character set for a language. The house number precedes the street name in the US and United Kingdom but follows the street name in most other European countries. Postal codes appear in various positions and can include alphabetic characters (e.g., an abbreviation for the country), separators (usually spaces), and numbers (up to seven characters and numbers in length). In many countries, each part of an address is written on a separate line; however, in South Korea, the entire address is placed on a single line, with the specific format used varying for central cities and local areas.

Telephone numbers can contain blanks, commas, hyphens, periods, and square brackets as separators. Telephone numbers can be displayed in local, national, and international formats. Local formats vary widely. National formats can have an area code in parentheses, while international formats can drop the parentheses but add a plus sign at the beginning of the number to indicate the country code.

17.4.4 Graphics

Icons and symbols. The icons and symbols used by an application may be unfamiliar to users outside the US. For example, a mail application that changes a mailbox graphic to indicate receipt of new mail may be unrecognizable in another culture where mailboxes have a different appearance or may not be used. Certain images, colors, and numbers of objects in a group may evoke a negative reaction in another culture so they obscure or contradict the message they are intended to convey. As a result, the icons used in an application may need to be modified in order to match the image or symbol to the culture in which the application will be used.

Whenever possible, an application uses international symbols in its icons. If a new symbol is created, it represents a basic, concrete concept because concrete icons require less explanation than abstract ones. In addition, each new symbol needs to be compared with existing symbols to ensure there are no conflicts. The use of stars and crosses as part of the symbol is avoided. Text is not included in an icon graphic because it will need to be translated and may not fit into the icon when presented in the target language.

Downloaded from http://www.everyspec.com

Drawings. An application incorporates translated text and adjusts data formats as needed when presenting graphic information (e.g., line graphs, bar charts, flow charts). The size of the graphics objects may need to be enlarged to accommodate the increased length of translated text. Alternatively, application graphics can be modified to place text adjacent to, rather than within, the object so changes in text length do not affect size of individual objects or the overall illustration.

Graphic design conventions vary from culture to culture. For example, Japanese artists tend to draw tables of data differently than Western artists do. As a result, an application may require modification to accommodate these conventions.

Tactical symbology. When an application presents tactical data (e.g., in a map window), users are able to access a variety of map features in order to customize the display to match their preferred mode for viewing and interpreting this information. For example, where US users are likely to display road features for navigation in urban areas, Korean operators may prefer to see neighborhood names as key map landmarks.

Visual cues for alerting. The specific visual signals used by an application, especially for alerting, are reviewed to ensure that they convey the desired meaning in the target culture and that their representations within the software are not objectionable to users. Alert and warning messages can be supplemented with icons so an application communicates critical information in both text and graphic form.

17.4.5 Keyboard Interaction

Mnemonics and shortcut keys. When menu options are translated, any mnemonics or shortcut keys included with the options need to be modified to reflect the translated text. In general, the guidelines for mnemonics in languages with single-byte character sets also apply to languages with multi-byte character sets, except for how mnemonics are displayed. An application translated from the former to the latter can retain the mnemonics used in the single-byte version, with the mnemonic displayed in parentheses following the text of the menu option. If all of the characters in a menu option have been assigned as mnemonics or if the choice consists of multi-byte characters, an application can use another letter or keyboard character. The same mnemonic is assigned to an option whenever it appears in a menu.

The layout of the user's keyboard needs to be considered when selecting the key combinations for the mnemonics and shortcut keys to assign to translated menu options. First, keys are selected to minimize the disruption or relearning required to execute a mnemonic or shortcut key, especially a frequently used one. Second, there are no conflicts between the key combinations for entering accented characters (e.g., if a US keyboard is being used) and those being used for mnemonics and shortcut keys. Finally, some non-US keyboards contain only one ALT, located either on the left or right side of the keyboard. The ease with which users can execute the key combination for a mnemonic is considered if one of these keyboards is being used.

Downloaded from http://www.everyspec.com

Speed search and text search. If an application is translated into a language that contains accents on the first letter of words, users are able to perform a speed search (see section 6.5.2) by typing an unaccented upper-case or lower-case letter, and the search finds instances of both unaccented and accented first letters.

17.4.6 Text Manipulation

Sorting and collation. An application makes use of linguistic sort sequences to order the contents of alphanumeric lists or to add new information to an already sorted list. In US software, characters are usually compared according to their binary value in the code set, with characters ordered on the basis of these values. However, variations are frequently required to reflect linguistic conventions since the binary sequence of characters may not match the linguistic sequence for the language. For example, variations may be needed to handle characters with functional equivalence (e.g., Mac and Mc usually appear together) and to address situations where a character should be ignored (e.g., re-locate and relocate should be placed together). In addition, where US software typically provides a single sorting algorithm to accommodate such variations, other languages usually support multiple sort orders. As a result, an application needs to provide users with the ability to choose a sort order that meets their needs.

Sorting rules for European languages must be able to handle extended character sets and language-specific conventions, independent of the binary values assigned to characters. These languages may contain letters after “z” or sort letters out of the standard alphabetic sequence used in the US. For example, some of these languages contain double characters that sort as one combined character, or a single character that is treated as a double character. In Spanish, double characters such as “ch” and “ll” sort as a single character, and in German, ß is a single character that is treated as “ss” when found in a word. Madell, Parsons, and Abegg in Developing and Localizing International Software provide the following examples of differences in sorting order based on ASCII and German rules, and ASCII and Spanish rules:

Sorted by ASCII rules	Sorted by German rules	Sorted by ASCII rules	Sorted by Spanish rules
Airplane	Airplane	chaleco	cuna
Zebra	ähnlich	cuna	chaleco
bird	bird	día	día
car	car	llave	loro
ähnlich	Zebra	loro	llave
		maíz	maíz

In the case of complex (i.e., multi-byte) languages, expressions can be written in a mixture of character sets. For example, the Japanese word for “water” may be written as a single Kanji character, as two Hiragana characters, as two Katakana characters, or as the four-letter Romaji expression “mizu.” As a result, sorting algorithms in these languages must be able to accept multiple character patterns as representing the same

Downloaded from http://www.everyspec.com

expression. These algorithms can combine a sorting order among the character sets with a sorting order for expressions within each set. In addition, an application may need to provide a sort order based on a symbol feature that is not captured within the character code. For example, Chinese expressions may need to be sorted by the numeric value of the character as represented in the coded character set as well as by the number of strokes required to represent the character, the radical (i.e., root) of the character, or the number of strokes added to the radical. Finally, an application may need to implement a sort order based on the way symbols are pronounced. In this case, each symbol may have to be stored in both graphic and phonetic form, with the resulting sort order listing symbols that are phonetically similar but visually different near each other.

Editing functions. Editing functions (e.g., search and replace, cut and paste, and spell checking) can accommodate the unique features of the target language, including instances where the appearance of a word changes when it is hyphenated, where it appears in lower rather than upper case, or where it contains a combination character such as æ. In contextual languages such as Thai, the characters that make up a compound symbol may be entered in several different orders, with the appearance of the symbol dependent on the order in which the characters are entered. In other languages (e.g., Greek), the appearance of a character can vary depending on its position in a word. If an application performs string searches in these languages, it is able to recognize any of several possible character sequences and judge them to be the same or different as appropriate.

17.4.7 Adjustments for Bi-directional Languages

If an application is localized to a bi-directional language such as Hebrew or Arabic, window orientation and information orientation within each window are adjusted as appropriate for right-to-left presentation. Window appearance is the mirror image of that in English-based windows, except that the location of the window buttons in the title bar does not change. In addition, window placement is oriented right-to-left; i.e., a primary window is positioned to the right and its child window(s) to the left.

With respect to information orientation in bi-directional languages, an application complies with the style specifications in this document, except that “right” and “left” are interchanged. However, physical right and left remain the same. As in unidirectional languages, LEFT and RIGHT move the cursor in the arrow direction; the right and left buttons on the pointing device behave as defined here, with left and right movement of the pointing device moving the pointer in these directions.

With respect to information content, an application provides translations for titles, headings, prompts, and other window controls, except for English acronyms not normally translated and the names of keys on the keyboard. If an application chooses to mix right-to-left and left-to-right elements within the same window, it follows the relevant specifications defining information display for unidirectional and bi-directional languages.

Downloaded from http://www.everyspec.com

17.4.8 Adjustments for Vertical Languages

Asian languages such as Chinese and Japanese contain a combination of horizontally and vertically written characters, with the latter written from top to bottom and each new line starting to the left of the previous one. If an application is converted to a vertical language, it provides translations for titles, headings, prompts, and other window controls, except for English acronyms not normally translated and the names of keys on the keyboard. An application displays this information in the orientation expected by users, complying with relevant specifications defining information display for horizontal and vertical languages. In addition, an application supports data entry for vertically written text using one of the text-input methods described in section 17.3 and presenting a vertically-oriented text entry area.

17.4.9 Printing

Peripheral devices such as printers are capable of handling the character set for the target language; i.e., the full character set can be loaded on the printer, and the printer can produce all of the extended characters required by the language.

While the standard paper size in the US is 8.5 x 11 inches, most countries use ISO A4 size which is slightly longer and narrower than the US standard. As a result, printer capabilities (e.g., different paper trays) may need to be adjusted in order to handle the standard paper and envelope sizes used by the target audience, and an application may need to be modified to handle the varying page layouts dictated by the different paper sizes. For example, hardcoded rules regarding paper margins are removed, and users are allowed to specify how they want text to appear and to do so using measurement units with which they are familiar.

Adjustments made in window format to accommodate text expansion also need to consider text presentation when the content of the window is printed. In particular, the amount of vertical space between lines of text is sufficient to print all extended characters, including those with accents, in both upper and lower case.

17.4.10 Internationalized Web Sites

While it is desirable to present the content of a Web site in a user's preferred language, the cost to translate the entire site into each of the languages used by its audience can be prohibitive. An alternative is to provide translations for some pages and leave others in the original language. This hybrid approach requires that a site decide on a default language, with users able to switch between available languages on the home page and lower-level pages as needed. If a site cannot decide on a default language for its home page, it can provide a staging page where users select their preferred language before navigating to the home page. Whenever users are presented with a choice of languages, the list should contain the name of the language as a word (rather than as a graphic such as a national flag), using the language's own name for itself.

Downloaded from http://www.everyspec.com

Because it is unlikely that all of the content of a Web site will be translated into every language it supports, the site needs to support a multilingual search capability in order to cover the entire information space of the site. This capability should be designed so that users enter the desired search terms in their preferred language and the site translates the terms into the requested languages before performing the search. This approach is preferred to having users identify the appropriate translated synonyms in each of the languages, which may result in inaccurate or incomplete searches.

Downloaded from http://www.everyspec.com

Appendix A: Action Vocabulary

This appendix presents a common vocabulary of action terminology, along with recommended mnemonics and/or shortcut keys. A mnemonic other than the one listed here may be used in order to assign a unique character in a menu, menu bar, or window. Mnemonics and shortcut keys are indicated as they would appear in the application, rather than in upper-case letters (which is the typographic convention in this document).

NOTE: In GNOME, the same shortcut keys may be assigned to multiple actions.

Term	GNOME		Java		Motif		MS Windows		Action
About	A	-----							Displays information about an application (e.g., name, version).
About <application name>			A	-----	A	-----	A	-----	Displays information about an application (e.g., name, version).
Add									Adds an object to a collection.
Add Bookmark	A	Ctrl+D							Adds a bookmark for an object to a bookmark list.
Align Center			C	Ctrl+E					Centers text between the margins. Aligns a graphic object with the middle of its container.
Align Left			L	Ctrl+L			L	Ctrl+L	Aligns text with the left margin. Aligns a graphic object with the left edge of its container.
Align Right			R	Ctrl+R			R	Ctrl+R	Aligns text with the right margin. Aligns a graphic object with the right edge of its container.
Append									Adds an object to the end of a collection.
Apply									Executes the changes made in a window but does not close it.
<application name> Help			--	F1			H	F1	Displays help information for an application.
Arrange All							A	-----	Arranges windows in a top to bottom format.
Back	B	Alt+Left					B	Alt+Left	Navigates back one page in a history list.
Best Fit	F	-----							Sets the magnification level for viewing an object so it fills the window.

Downloaded from http://www.everyspec.com

Term		GNOME		Java		Motif		MS Windows	Action
Bold	B	Ctrl+B	B	Ctrl+B			B	Ctrl+B	Sets font style to bold.
Bookmarks (menu title)	B	-----							Provides access to options that allow users to browse objects.
Break							B	-----	Inserts a page break.
Browse							B	-----	Opens a view of the file system so the user can select a file.
Cancel[1]									Closes a window without executing the changes made in it.
Cascade							C	-----	Arranges windows in a cascade format.
Cells	C	-----					e	-----	Sets the properties of cells in a table.
Center							C	Ctrl+E	Centers text between the margins. Aligns a graphic object with the middle of its container.
Change View					C	-----			Changes to a different view of an object.
Clear					e	Delete	a	Delete	Removes an object from its location without copying it to the clipboard or compressing the remaining space.
Close (in a Control menu)	C	Alt+F4			C	Alt+F4	C	Alt+F4	Closes a window; requests confirmation if unsaved changes have been made.
Close (in a Control menu)							C	Ctrl+W	Closes a document window; requests confirmation if unsaved changes have been made.
Close (in a File menu)	C	Ctrl+W	C	Ctrl+W	C	-----	C	-----	Closes an object; requests confirmation if unsaved changes have been made.
Close All	C	Shift+ Ctrl+W							Closes all objects; requests confirmation if unsaved changes have been made.
Columns	C	-----					C	-----	Inserts columns in a table.
Compare									Displays information on multiple objects in a collection.
Contents	C	F1	C	-----					Displays a list of help topics for an application.
Copy	C	Ctrl+C	C	Ctrl+C	C	Ctrl+C	C	Ctrl+C	Copies an object to the clipboard without removing it from its location.

[1] Cancel is used when the settings selected by users are reversible; i.e., Cancel returns the settings in a window to their most recent past state. Close is used when the settings are saved, when a new state has been defined, or when new data have been generated as a result of selections made in the window.

Downloaded from http://www.everyspec.com

Term	GNOME		Java		Motif		MS Windows		Action
Copy Here							C	-----	Copies an object to a destination specified by the user.
Create Shortcut							S	-----	Creates a shortcut icon for an object.
Customize							C	-----	Customizes the contents of menus and toolbars.
Cut	t	Ctrl+X	t	Ctrl+X	t	Ctrl+X	t	Ctrl+X	Copies an object to the clipboard and removes it from its location.
Date and Time	D	-----					T	-----	Inserts a date and/or time.
Defaults									Displays the default settings defined by the application.
Delete	D	Delete	D	Delete	D	Delete	D	-----	Removes an object from its location without copying it to the clipboard, and compresses the remaining space.
Deselect All	l	Shift+ Ctrl+A			l	Ctrl+\			Deselects all selected objects in a collection.
Details	D	-----	D	-----			D	-----	Displays objects as a list of small icons with labels and other information to the right.
Diagram							g	-----	Inserts a diagram.
Drag					--	-----			Transfers an object from a drag site to a drop site.
Duplicate	p	Ctrl+U			--	-----	i	Ctrl+D	Creates a copy of an object.
Edit (menu title)	E	-----	E	-----	E	-----	E	-----	Provides access to options that change the content of an object and locate specific information in the object.
Edit									Edits the contents of an object.
Edit Bookmarks	E	Ctrl+B							Edits the contents of a bookmark list.
Exit			x	-----	x	-----	x	-----	Ends processing by an application; closes all application windows and requests confirmation if unsaved changes have been made.
External Link	x	-----							Inserts a link to an object in an external file.
File							l	-----	Inserts the contents of a file.
File (menu title)	F	-----	F	-----	F	-----	F	-----	Provides access to options that manipulates the object(s) represented in a window.
Filter	l	-----	F	-----					Hides an object based on criteria specified by the user.

Downloaded from http://www.everyspec.com

Term		GNOME		Java		Motif		MS Windows	Action
Find	F	Ctrl+F	F	Ctrl+F	--	Ctrl+F	F	Ctrl+F	Searches for an object that matches criteria specified by the user.
Find Again			n	Ctrl+G					Searches for the next instance of an object that matches criteria specified by the user.
Find Next	x	Ctrl+G							Searches for the next instance of an object that matches criteria specified by the user.
Find Previous	v	Shift+ Ctrl+G							Searches for the previous instance of an object that matches criteria specified by the user.
First Page	F	Ctrl+Home							Navigates to the first page in a document.
Font	F	-----	F	-----			F	-----	Sets the font of text.
Format (menu title)	m	-----	r	-----			o	-----	Provides access to options that change the visual appearance of an object.
Forward	F	Alt+Right					F	Alt+Right	Navigates forward one page in a history list.
From File	F	-----							Inserts an object from an external file.
Go (menu title)	G	-----							Provides access to options that navigate within an object or collection of objects.
Go To							G	Ctrl+G	Navigates to a location specified by the user.
Go To Page	G	-----							Navigates to a document page specified by the user.
Graph	G	-----							Inserts a graph.
Group									Combines individual objects into a grouped object.
Help (menu title)	H	-----	H	-----	H	-----	H	-----	Provides access to options that provide additional information about a window or application.
Help									Displays help information for a window.
Hide							H	-----	Removes elements (e.g., toolbar, status bar) from view in a window.
Home	H	Alt+Home							Navigates to a home page.
Hyperlink							i	Ctrl+K	Inserts a hyperlink.
Icons	I	-----							Displays objects as rows and columns of large icons with labels below.

Downloaded from http://www.everyspec.com

Term	GNOME		Java		Motif		MS Windows		Action
Image	I	-----							Inserts an image.
Index			I	-----					Displays an index of help topics for an application.
Insert	I	-----							Adds an object at a location within a collection.
Insert (menu title)	I	-----			I	-----	I	-----	Provides access to options that list the types of objects that can be inserted in a window.
Invert Selection		Ctrl+I							Toggles the select state of objects in a window.
Italic	I	Ctrl+I	I	Ctrl+I			I	Ctrl+I	Sets font style to italic.
Large Icons			g	-----			g	-----	Displays objects as rows and columns of large icons with labels below.
Last Page	L	Ctrl+End							Navigates to the last page in a document.
Layer	r	-----							Sets the properties of the layers in a multi-layered object.
Links							k	-----	Creates a link between two objects.
Link Here							L	-----	Creates a link between an object and a destination indicated by the user.
List (in a Format menu)	L	-----							Sets the properties of a list.
List (in a View menu)	L	-----	L	-----			L	-----	Displays objects as a list of small icons with labels on the right.
Location	L	-----							Navigates to a URL specified by the user.
Logout									Ends processing by a system; closes all windows and requests confirmation if unsaved changes have been made.
Lower					L	-----			Moves a window to the bottom of the window stack.
Mark									Adds an annotation (e.g., an asterisk) to an object.
Maximize	x	Alt+F10			x	Alt+F10	x	-----	Changes a window to its maximum size.
Merge									Combines the content of two objects into a single object.
Minimize	n	Alt+F9			n	Alt+F9	n	-----	Changes a window to its minimum size.
More					--	-----			Expands a window to display additional information.
Move	M	Alt+F7			M	Alt+F7	M	-----	Changes the location of a window.
Move Here							M	-----	Moves an object to a destination specified by the user.

Downloaded from http://www.everyspec.com

Term	GNOME		Java		Motif		MS Windows		Action
Move to <workspace name>	--	-----							Places a window in a workspace.
New	N	Ctrl+N	N	Ctrl+N	N	Ctrl+N	N	Ctrl+N	Creates a new object.
New Window					N	-----	N	-----	Opens a new window containing the current view of an object.
Next									Navigates to the next object in a collection.
Next Page	N	PageDown							Navigates to the next page in a document.
No									Indicates a negative response to a question.
Normal			N	-----					Sets font style to normal.
Normal Size	N	Ctrl+=							Resets the magnification level for viewing an object to its default value.
Previous Page	P	PageUp							Navigates to the previous page in a document.
Primary Copy					--	-----			Copies an object to a destination without using the clipboard for intermediate storage.
Primary Link					--	-----			Places a link from an object to a destination without using the clipboard for intermediate storage.
Primary Move					--	-----			Moves an object to a destination without using the clipboard for intermediate storage.
Print	P	Ctrl+P	P	Ctrl+P	P	Ctrl+P	P	Ctrl+P	Prints an object.
Print One									Prints an object using the default print parameters.
Print Preview	w	-----					v	-----	Displays the contents of an object as it would appear when printed.
Print Setup					R	-----			Sets the parameters for printing.
Properties	t	-----			s	Ctrl+I	r	-----	Sets the properties of an object or collection of objects.
Put on All Workspaces	A	-----							Places a window in all workspaces.
Quit	Q	Ctrl+Q							Ends processing by an application; closes all application windows and requests confirmation if unsaved changes have been made.

Downloaded from http://www.everyspec.com

Term		GNOME		Java		Motif		MS Windows	Action
Redo	R	Shift+ Ctrl+Z	R	Ctrl+Y	R	Ctrl+Y	R	Ctrl+Y	Performs the most recently undone action. Identifies the specific action if the Redo can perform a sequence of undone actions.
Reference					R	-----			Displays help reference pages for an application.
Refresh	R	Ctrl+R	R	F5			R	F5	Updates the contents of an object to reflect its current state.
Rename	--	F2					m	-----	Renames an object; does not affect the contents of the object.
Replace	e	Ctrl+H	R	Ctrl+H	--	-----	e	Ctrl+H	Searches for an object and replaces it with a different object specified by the user.
Reset					--	-----			Restores the settings in a window to their state at the last time a change was saved; if no changes were saved, restores the settings to their state when the window was first opened.
Reset to Default					--	-----			Restores the settings in a window to the default defined by an application.
Resize	R	Alt+F8							Changes the size of a window.
Restore					R	Alt+F5	R	-----	Restores a window to its previous size and location.
Resume					--	-----	R	-----	Resumes a process.
Retry					--	-----	R	-----	Causes a process to be attempted again.
Revert	R	-----			--	-----			Reverts an object to its state when it was last saved.
Roll Up	U	-----							Changes a window to show only its title bar.
Rows	R	-----					R	-----	Inserts rows in a table.
Save	S	Ctrl+S	S	Ctrl+S	S	Ctrl+S	S	Ctrl+S	Saves an object; prompts the user for a name if the object does not have one.
Save A Copy	v	-----							Saves a copy of an object; prompts the user for the new name.
Save All	S	-----							Saves all open objects; prompts the user for names if the objects do not have one.
Save As	A	Shift+ Ctrl+S	a	-----	A	-----	A	-----	Saves an object under a new name; prompts the user for the new name.
Save As Default					--	-----			Saves the current settings in a window as the default.
Search			S	-----					Searches for an object that matches criteria specified by the user.
Select All	A	Ctrl+A	A	Ctrl+A	S	Ctrl+A	A	Ctrl+A	Selects all objects in a collection.

Downloaded from http://www.everyspec.com

Term	GNOME		Java		Motif		MS Windows	Action
Selected (menu title)				S	-----			Provides access to options for manipulating the object(s) currently selected in a window.
Send To	e	Ctrl+E				d	-----	Sends an object to another location or user.
Sheet	S	-----						Inserts a new sheet in a workbook.
Show						S	-----	Displays elements (e.g., toolbar, status bar) in a window.
Size (in a Control menu)				S	Alt+F8	S	-----	Changes the size of a window.
Size (in a Format menu)		z	-----					Sets the font size of text.
Small Icons		m	-----			m	-----	Displays objects in rows and columns of small icons with labels to the right.
Sort				--	-----			Arranges a collection of objects in an order specified by the user.
Sort Ascending								Arranges a collection of objects in ascending order.
Sort By	S	-----	S	-----				Arranges a collection of objects in an order based on an attribute specified by the user.
Sort Descending								Arranges a collection of objects in descending order.
Spelling						S	F7	Checks the spelling of text.
Split						p	-----	Splits a window into panes.
Start								Begins or turns on a process.
Status Bar	S	-----				B	-----	Shows or hides the status bar in a window.
Stop				--	-----	S	-----	Ends or turns off a process.
Style	S	-----	S	-----		S	-----	Sets the formatting of text.
Symbol	m	-----				S	-----	Inserts a special symbol.
Table of Contents				C	-----			Displays a list of help topics for an application.
Tasks				T	-----			Displays information on how to perform tasks using an application.
Theme						h	-----	Sets the theme for a document.
Tile						T	-----	Arranges windows in a tiled format.
Toolbar	T	-----				T	-----	Shows or hides the toolbar in a window.

Downloaded from http://www.everyspec.com

Term	GNOME		Java		Motif		MS Windows		Action
Tools (menu title)							T	-----	Provides access to options for launching the tools provided in an application.
Tutorial			T	-----					Displays a tutorial on how to use an application.
Underline	U	Ctrl+U	U	Ctrl+U			U	Ctrl+U	Sets font style to underline.
Undo	U	Ctrl+Z	U	Ctrl+Z	U	Ctrl+Z	U	Ctrl+Z	Reverses the most recently executed action. Identifies the specific action if Undo can reverse a sequence of previous actions.
Ungroup									Separates a grouped object into individual objects.
Unmark									Removes the annotation (e.g., an asterisk) to an object.
Unmaximize	x	-----							Restores a maximized window to its previous size and location.
Unminimize	n	-----							Restores a minimized window to its previous size and location.
Unoccupy Workspace					U	-----			Removes a window from the current workspace.
Unroll	U	-----							Restores a shaded window to its previous size.
Up	U	Alt+Up							Navigates to the parent of a page.
Update									Displays updated information on the status of a process.
Using Help					U	-----			Displays information about how the help system works.
View (menu title)	V	-----	V	-----	V	-----	V	-----	Provides access to options for changing the user's view of an object.
What's This?							W	Shift+F1	Invokes Help mode so the user can indicate where help is needed in a window.
Window (menu title)	W	-----			W	-----	W	-----	Provides access to options that apply to the open windows in an application.
Worksheet							W	-----	Inserts a new sheet in a workbook.

Downloaded from http://www.everyspec.com

Term		GNOME		Java	Motif		MS Windows	Action
Yes								Indicates an affirmative response to a question.
Zoom						Z	-----	Changes the magnification level for viewing an object.
Zoom In	Z	Ctrl+Plus	I	-----				Increases the magnification level for viewing an object.
Zoom Out	O	Ctrl+Minus	O	-----				Decreases the magnification level for viewing an object.

Downloaded from http://www.everyspec.com

Appendix B: Federal Accessibility Standards

Section 508 of the Rehabilitation Act of 1973 requires that electronic and information technology developed, procured, maintained, or used by the federal government be accessible to people with disabilities. Amendments made to the Rehabilitation Act by Congress in 1998 expanded and strengthened these requirements by creating mandatory standards concerning technology accessibility that are being incorporated into federal procurement regulations. The accessibility standards, which were published in December 2000 and took effect in June 2001, cover all types of information technology in the federal sector. The standards apply to software applications and operating systems, web-based intranet and internet information and applications, telecommunications products, video and multi-media products, self-contained, closed products such as information kiosks, and desktop and portable computers.

Federal agencies are required to comply with the accessibility standards unless the standards would impose an undue burden (i.e., significant difficulty or expense) on the agency. Even if the agency demonstrates that compliance poses an undue burden, it must still provide information through an alternate means of access that can be used by individuals with disabilities. Federal agencies must procure accessible information technology regardless of whether they have employees with disabilities. Agencies are to apply the standards prospectively and are not required to retrofit existing electronic and information technology.

The scope of Section 508 is limited to the federal sector, with a limited exemption for systems used for military command, weaponry, intelligence, and cryptologic activities. The exemption does not apply to routine business and administrative systems used for other defense-related purposes or by defense agencies or personnel. The following excerpt from the Federal Register provides additional information on the applicability of the standards to defense systems:

> "... Software used for payroll, finance, word processing software used for production of routine documents, ordinary telephones, copies, fax machines, and web applications must still comply with the standards even if they are developed, procured, maintained, or used by an agency engaged in intelligence or military activities. The [Access] Board understands that the Department of Defense interprets this to mean that a computer designed to provide early missile launch detection would not be subject to these standards, nor would administrative or business systems that must be architecturally tightly coupled with a mission critical, national security system, to ensure interoperability and mission accomplishment."

Section 508 requires that individuals with disabilities have access to and use of information that is comparable to that provided to federal employees and members of the public who are not disabled. According to the standards, information technology is accessible to people with disabilities if it can be used in a variety of ways that do not depend on a single sense or ability. Section 508 does not require that accessibility-related

Downloaded from http://www.everyspec.com

software or peripheral devices be purchased, only that the information delivered to or provided by users be available to assistive technology. Subparts B, C, and D of the accessibility standards created under Section 508 follow this overview; additional information about Section 508 requirements can be found at http://www.access-board.gov/sec508/508 standards.htm.

Part 1194 - ELECTRONIC AND INFORMATION TECHNOLOGY ACCESSIBILITY STANDARDS

Subpart B - Technical Standards

§ 1194.21 Software applications and operating systems.

(a) When software is designed to run on a system that has a keyboard, product functions shall be executable from a keyboard where the function itself or the result of performing a function can be discerned textually.

(b) Applications shall not disrupt or disable activated features of other products that are identified as accessibility features, where those features are developed and documented according to industry standards. Applications also shall not disrupt or disable activated features of any operating system that are identified as accessibility features where the application programming interface for those accessibility features has been documented by the manufacturer of the operating system and is available to the product developer.

(c) A well-defined on-screen indication of the current focus shall be provided that moves among interactive interface elements as the input focus changes. The focus shall be programmatically exposed so that assistive technology can track focus and focus changes.

(d) Sufficient information about a user interface element including the identity, operation and state of the element shall be available to assistive technology. When an image represents a program element, the information conveyed by the image must also be available in text.

(e) When bitmap images are used to identify controls, status indicators, or other programmatic elements, the meaning assigned to those images shall be consistent throughout an application's performance.

(f) Textual information shall be provided through operating system functions for displaying text. The minimum information that shall be made available is text content, text input caret location, and text attributes.

(g) Applications shall not override user selected contrast and color selections and other individual display attributes.

Downloaded from http://www.everyspec.com

(h) When animation is displayed, the information shall be displayable in at least one non-animated presentation mode at the option of the user.

(i) Color coding shall not be used as the only means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.

(j) When a product permits a user to adjust color and contrast settings, a variety of color selections capable of producing a range of contrast levels shall be provided.

(k) Software shall not use flashing or blinking text, objects, or other elements having a flash or blink frequency greater than 2 Hz and lower than 55 Hz.

(l) When electronic forms are used, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues.

§ 1194.22 Web-based intranet and internet information and applications.

(a) A text equivalent for every non-text element shall be provided (e.g., via "alt", "longdesc", or in element content).

(b) Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation.

(c) Web pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup.

(d) Documents shall be organized so they are readable without requiring an associated style sheet.

(e) Redundant text links shall be provided for each active region of a server-side image map.

(f) Client-side image maps shall be provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape.

(g) Row and column headers shall be identified for data tables.

(h) Markup shall be used to associate data cells and header cells for data tables that have two or more logical levels of row or column headers.

(i) Frames shall be titled with text that facilitates frame identification and navigation.

(j) Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz.

Downloaded from http://www.everyspec.com

(k) A text-only page, with equivalent information or functionality, shall be provided to make a web site comply with the provisions of this part, when compliance cannot be accomplished in any other way. The content of the text-only page shall be updated whenever the primary page changes.

(l) When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology.

(m) When a web page requires that an applet, plug-in or other application be present on the client system to interpret page content, the page must provide a link to a plug-in or applet that complies with §1194.21(a) through (l).

(n) When electronic forms are designed to be completed on-line, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues.

(o) A method shall be provided that permits users to skip repetitive navigation links.

(p) When a timed response is required, the user shall be alerted and given sufficient time to indicate more time is required.

§ 1194.23 Telecommunications products.

(a) Telecommunications products or systems which provide a function allowing voice communication and which do not themselves provide a TTY functionality shall provide a standard non-acoustic connection point for TTYs. Microphones shall be capable of being turned on and off to allow the user to intermix speech with TTY use.

(b) Telecommunications products which include voice communication functionality shall support all commonly used cross-manufacturer non-proprietary standard TTY signal protocols.

(c) Voice mail, auto-attendant, and interactive voice response telecommunications systems shall be usable by TTY users with their TTYs.

(d) Voice mail, messaging, auto-attendant, and interactive voice response telecommunications systems that require a response from a user within a time interval, shall give an alert when the time interval is about to run out, and shall provide sufficient time for the user to indicate more time is required.

(e) Where provided, caller identification and similar telecommunications functions shall also be available for users of TTYs, and for users who cannot see displays.

Downloaded from http://www.everyspec.com

(f) For transmitted voice signals, telecommunications products shall provide a gain adjustable up to a minimum of 20 dB. For incremental volume control, at least one intermediate step of 12 dB of gain shall be provided.

(g) If the telecommunications product allows a user to adjust the receive volume, a function shall be provided to automatically reset the volume to the default level after every use.

(h) Where a telecommunications product delivers output by an audio transducer which is normally help up to the ear, a means of effective magnetic wireless coupling to hearing technologies shall be provided.

(i) Interference to hearing technologies (including hearing aids, cochlear implants, and assistive listening devices) shall be reduced to the lowest possible level that allows a user of hearing technologies to utilize the telecommunications product.

(j) Products that transmit or conduct information or communication, shall pass through cross-manufacturer, non-proprietary, industry-standard codes, translation protocols, formats, or other information necessary to provide the information or communication in a usable format. Technologies which use encoding, signal compression, format transformation, or similar techniques shall not remove information needed or access or shall restore it upon delivery.

(k) Products which have mechanically operated controls or keys, shall comply with the following:

(1) Controls and keys shall be tactilely discernible without activating the controls or keys.

(2) Controls and keys shall be operable with one hand and shall not require tight grasping, pinching, or twisting of the wrist. The force required to activate controls and keys shall be 5 lbs. (22.2N) maximum.

(3) If key repeat is supported, the delay before repeat shall be adjustable to at least 2 seconds. Key repeat rate shall be adjustable to 2 seconds per character.

(4) The status of all locking or toggle controls or keys shall be visually discernible, and discernible either through touch or sound.

§ 1194.24 Video and multimedia products.

(a) All analog television displays 13 inches and larger, and computer equipment that includes analog television receiver or display circuitry, shall be equipped with caption decoder circuitry which appropriately receives, decodes, and displays closed captions from broadcast, cable, videotape, and DVD signals. As soon as practicable, but not later than July 1, 2002, widescreen digital television (DTV) displays measuring at least 7.8

Downloaded from http://www.everyspec.com

inches vertically, DTV sets with conventional displays measuring at least 13 inches vertically, and stand-alone DTV tuners, whether or not they are marketed with display screens, and computer equipment that includes DTV receiver or display circuitry, shall be equipped with caption decoder circuitry which appropriately receives, decodes, and displays closed captions from broadcast, cable, videotape, and DVD signals.

(b) Television tuners, including tuner cards for use in computers, shall be equipped with secondary audio program playback circuitry.

(c) All training and informational video and multimedia productions which support the agency's mission, regardless of format, that contain speech or other audio information necessary for the comprehension of the content, shall be open or closed captioned.

(d) All training and informational video and multimedia productions which support the agency's mission, regardless of format, that contain visual information necessary for the comprehension of the content, shall be audio described.

(e) Display or presentation of alternate text presentation or audio descriptions shall be user-selectable unless permanent.

§ 1194.25 Self contained, closed products.

(a) Self contained products shall be usable by people with disabilities without requiring an end-user to attach assistive technology to the product. Personal headsets for private listening are not assistive technology.

(b) When a timed response is required, the user shall be alerted and given sufficient time to indicate more time is required.

(c) Where a product utilizes touchscreens or contact-sensitive controls, an input method shall be provided that complies with § 1194.23 (k) (1) through (4).

(d) When biometric forms of user identification or control are used, an alternative form of identification or activation, which does not require the user to possess particular biological characteristics, shall also be provided.

(e) When products provide auditory output, the audio signal shall be provided at a standard signal level through an industry standard connector that will allow for private listening. The product must provide the ability to interrupt, pause, and restart the audio at any time.

(f) When products deliver voice output in a public area, incremental volume control shall be provided with output amplification up to a level of at least 65 dB. Where the ambient noise level of the environment is above 45 dB, a volume gain of at least 20 dB above the ambient level shall be user selectable. A function shall be provided to automatically reset the volume to the default level after every use.

Downloaded from http://www.everyspec.com

(g) Color coding shall not be used as the only means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.

(h) When a product permits a user to adjust color and contrast settings, a range of color selections capable of producing a variety of contrast levels shall be provided.

(i) Products shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz.

(j) Products which are freestanding, non-portable, and intended to be used in one location and which have operable controls shall comply with the following:

(1) The position of any operable control shall be determined with respect to a vertical plane, which is 48 inches in length, centered on the operable control, and at the maximum protrusion of the product within the 48 inch length.

(2) Where any operable control is 10 inches or less behind the reference plane, the height shall be 54 inches maximum and 15 inches minimum above the floor.

(3) Where any operable control is more than 10 inches and not more than24 inches behind the reference plane, the height shall be 46 inches maximum and 15 inches minimum above the floor.

(4) Operable controls shall not be more than 24 inches behind the reference plane.

§ 1194.26 Desktop and portable computers.

(a) All mechanically operated controls and keys shall comply with § 1194.23 (k) (1) through (4).

(b) If a product utilizes touchscreens or touch-operated controls, an input method shall be provided that complies with § 1194.23 (k) (1) through (4).

(c) Where biometric forms of user identification or control are used, an alternative form of identification or activation, which does not require the user to possess particular biological characteristics, shall also be provided.

(d) Where provided, at least one of each type of expansion slots, ports and connectors shall comply with publicly available industry standards.

Subpart C - Functional Performance Criteria

§ 1194.31 Functional performance criteria.

Downloaded from http://www.everyspec.com

(a) At least one mode of operation and information retrieval that does not require user vision shall be provided, or support for assistive technology used by people who are blind or visually impaired shall be provided.

(b) At least one mode of operation and information retrieval that does not require visual acuity grater than 20/70 shall be provided in audio and enlarged print output working together or independently, or support for assistive technology used by people who are visually impaired shall be provided.

(c) At least one mode of operation and information retrieval that does not require user hearing shall be provided, or support for assistive technology used by people who are deaf or hard of hearing shall be provided.

(d) Where audio information is important for the use of a product, at least one mode of operation and information retrieval shall be provided in an enhanced auditory fashion, or support for assistive hearing devices shall be provided.

(e) At least one mode of operation and information retrieval that does not require user speech shall be provided, or support for assistive technology used by people with disabilities shall be provided.

(f) At least one mode of operation and information retrieval that does not require fine motor control or simultaneous actions and that is operable with limited reach and strength shall be provided.

Subpart D - Information, Documentation, and Support

§ 1194.41 Information, documentation, and support.

(a) Product support documentation provided to end-users shall be made available in alternate formats upon request, at no additional charge.

(b) End-users shall have access to a description of the accessibility and compatibility features of products in alternate formats or alternate methods upon request, at no additional charge.

(c) Support services for products shall accommodate the communication needs of end-users with disabilities.

Downloaded from http://www.everyspec.com

Appendix C: Acronyms and Abbreviations

2D	Two-dimensional
3D	Three-dimensional
ASCII	American Standard Code for Information Interchange
C4I	Command, Control, Communications, Computers, and Intelligence
CDE	Common Desktop Environment
COE	Common Operating Environment
COP	Common Operational Picture
COTS	Commercial Off-the-Shelf
CSS	Cascading Style Sheets
DII	Defense Information Infrastructure
DISA	Defense Information Systems Agency
DoD	Department of Defense
FAQ	Frequently Asked Questions
GCCS	Global Command and Control System
GCSS	Global Combat Support System
GIF	Graphics Interchange Format
GNOME	GNU (GNU's Not Unix) Object Model Environment
GUI	Graphical User Interface
HTML	Hypertext Markup Language
ICSF	Integrated C4I System Framework
I&RTS	Integration and Runtime Specification
JPEG	Joint Photographic Experts Group
JTA	Joint Technical Architecture
MDI	Multiple Document Interface
MIL-STD	Military Standard
MS	Microsoft
SCTP	Style Compliance Test Protocol
SDI	Single Document Interface
TMS	Track Management System
URL	Uniform Resource Locator
XIS	Extensible Information Framework

Downloaded from http://www.everyspec.com

This page intentionally left blank.

Downloaded from http://www.everyspec.com

Appendix D: UIS, GNOME, Java, Motif, and MS Windows Terminology

UIS Term	GNOME Term	Java Term	Motif Term	MS Windows Term
Activation	Activation	Activation	Activation	Activation
Application	Application	Application	Application	Application
Application help	Online help	----	Online documentation	Reference help
Application icon	Application icon	-----	Application icon	Application icon
Assistant window	Assistant	Wizard	-----	Wizard
Badge	-----	Badge	-----	-----
Check box	Check box	Check box	Check box	Check box
Client area	-----	Control area	Client area	-----
Clipboard transfer	-----	-----	Clipboard transfer	Command-based transfer
Close button	Close window button	Close button	-----	Close button
Combo box	Combo box	-----	-----	Combo box
Container	Container	Container	Parent container	Container
Context-sensitive help	Context-sensitive help	Contextual help	Context-sensitive help	Context-sensitive help
Control	Control	Control	Control	Control
Control menu	Window menu	-----	Window menu	Window menu
Control menu button	Window menu button	-----	Window menu button	-----
Default action	Default action	Default action	Default action	Default action
Desktop	Desktop	Desktop	Desktop	Desktop
Dialog window	Dialog box	Dialog box	Dialog	Dialog box
Display control button	-----	Turner	Display control button	Display control button
Document window	-----	-----	-----	Document window
Drag transfer	Drag and drop	Drag and drop	Drag and drop transfer	Drag and drop
Drop-down combo box	Drop-down combo box	Editable combo box	Drop-down combination box	Drop-down combo box
Drop-down list box	Drop-down list box	Non-editable combo box	Drop-down list box	Drop-down list box
Element cursor	-----	Rectangular border	Element cursor	Outline cursor
Ellipsis	Ellipsis	Ellipsis	Ellipsis	Ellipsis

Downloaded from http://www.everyspec.com

UIS Term	GNOME Term	Java Term	Motif Term	MS Windows Term
Expand-action button	-----	-----	Expand button	Unfold button
Expandable window	-----	-----	Expandable window	Expandable window
Explicit apply window	Explicit apply window	Explicit apply window	-----	-----
Explicit destruction	-----	-----	Explicit destruction	-----
Folder	Folder	Folder	Folder	Folder
Function key	Function key	Function key	Function key	-----
Grip handle	-----	Drag area	-----	Grip handle
Group box	Frame	Titled border	Group box	Group box
Handle	Handle	-----	Grab handle	Handle
Help mode	Context-sensitive help	Contextual help	Context-sensitive help	Help mode
Hot spot	-----	-----	Hot spot	Hot spot
Hot zone	-----	-----	-----	Hot zone
Icon	Icon	Icon	Icon	Icon
Icon box	-----	-----	Window icon box	-----
Input focus	Focus	Keyboard focus	Input focus	Input focus
Instant apply window	Instant apply window	-----	-----	-----
Instant help	Help	Help	Instant help	-----
Keyboard	Keyboard	Keyboard	Keyboard	Keyboard
Label	Label	Label	Label	Label
List box	List	List box	List box	List box
Location cursor	-----	-----	Cursor	Input focus indicator
Maximize button	Maximize button	Maximize button	Maximize button	Maximize button
Menu	Menu	Menu	Menu	Menu
Menu bar	Menu bar	Menu bar	Menu bar	Menu bar
Menu button	-----	-----	-----	Menu button
Menu option	Menu item	Menu item	Menu option	Menu item

Downloaded from http://www.everyspec.com

UIS Term	GNOME Term	Java Term	Motif Term	MS Windows Term
Menu title	Menu title	Menu title	Menu bar item	Menu title
Menu window	-----	-----	Menu dialog	-----
Message window	Alert	Alert box	Message dialog	Message box
Minimize button	Minimize button	Minimize button	Minimize button	Minimize button
Mnemonic	Access key	Mnemonic	Mnemonic	Access key
Modifier key	Modifier key	Modifier key	Modifier key	Modifier key
Mouse-over border	-----	Mouse-over border	-----	Hot-tracked appearance
Mouse-over feedback	-----	Mouse-over feedback	-----	-----
Multiple-column list box	-----	Table	Multiple-column list box	Multiple-column list box
Multiple document interface	Multiple document interface	Multiple document interface	-----	Multiple document interface
Navigation	Navigation	Navigation	Navigation	Navigation
Object	Object	Object	Object	Object
Object-action paradigm	-----	-----	Object-action selection model	Object-action paradigm
Option menu	Option menu	-----	Option menu	-----
Paned area	-----	Split pane	Paned box	Pane
Pointer	Pointer	Pointer	Pointer	Pointer
Pointing device	Pointing device	Pointing device	Pointing device	Pointing device
Pop-up menu	Pop-up menu	Contextual menu	Pop-up menu	Shortcut menu
Pop-up window	-----	Plain window	-----	Pop-up window
Primary transfer	-----	-----	Primary transfer	-----
Primary window	Primary window	Primary window	Primary window	Primary window
Processing mode	Mode	Mode	Mode	Mode
Progress bar	Progress bar	Progress bar	Gauge	Progress indicator
Pull-down menu	Drop-down menu	Drop-down menu	Pull-down menu	Drop-down menu
Push button	Button	Command button	Push button	Command button
Quick transfer	-----	-----	Quick transfer	-----
Radio button	Radio button	Radio button	Radio button	Option button
Resize border	Border	Border	Size border	Sizing border

Downloaded from http://www.everyspec.com

UIS Term	GNOME Term	Java Term	Motif Term	MS Windows Term
Sash	-----	-----	Sash	-----
Scale	Slider	Slider	Slider	Slider
Scroll arrow	-----	Scroll arrow	Stepper arrow	Scroll arrow button
Scroll bar	Scroll bar	Scroll bar	Scroll bar	Scroll bar
What's This? button	-----	-----	-----	What's This? button
Window	Window	Window	Window	Window
Window button	Window button	-----	-----	Window button
Window family	-----	-----	Window family	-----
Window frame	Border	Border	Window frame	Window frame
Window icon	-----	-----	Window icon	-----
Window mode	Window mode	Window mode	Window mode	Window mode
Window title	Window title	Window title	Window title	Window title
Zoom button	-----	Zoom button	-----	-----

Downloaded from http://www.everyspec.com

Appendix E: Glossary

A

Activation – The manner in which users invoke an action on the currently selected object(s).

Applet - An application program that is written in a platform-independent language (e.g., Java) and integrated into a Web page.

Application – The software to perform a set of related tasks. An application can be composed of one or more segments, each taken from a different source.

Application help - Information about the features of an application and how to use them.

Application icon - A pictorial representation of an application that is displayed on the desktop and, when activated, launches the application.

Assistant window - A dialog window containing multiple pages that guide users through the steps in a task.

B

Background image - A graphic that is displayed behind the text on a Web page.

Badge – An indicator added to a toolbar button label.

Bar chart - A graphic display that compares a single measure at several intervals.

Browser – The software that renders information specified in a markup language (e.g., HTML) in a standard format on different platforms. A browser provides the window in which users view the information on a Web page.

C

Check box - A control for making a many-of-many selection. A check box consists of a square-shaped indicator and a label describing the state being set.

Client area - An area of a window where an application displays information and interacts with the user.

Clipboard transfer - An interaction technique in which users transfer an object from its current location to a temporary clipboard and then from the clipboard to a new location.

Close button - A window component that, when activated, dismisses the window.

Downloaded from http://www.everyspec.com

Combo box - A control consisting of a text box and a list box. Users type in the text box, or they select an item in the list box to display in the text box.

Container - An object that holds other objects.

Content page - A Web page that presents text and graphic information about a specific topic or concept.

Context-sensitive help – Contextual information about individual objects in the window with focus.

Control - An interface component with which users interact (e.g., select an item, enter text, initiate an action).

Control menu - A menu that contains options for performing window management functions in a window.

Control menu button - A window component that, when activated, displays the Control menu for a window.

D

Data entry page – A Web page that allows users to interact with the information on the page.

Default action - The action that users would most likely want to execute in the window with focus.

Desktop - The workspace on which windows and objects are placed. The desktop normally fills the entire screen.

Dialog window - A secondary window used for short-term interactions related to sub-task or function control.

Display control button - A button located to the left of a container icon in a tree control that toggles between showing and hiding the objects in the container.

Document window - A window in an MDI application that displays the content of a document or file.

Drag transfer - An interaction technique in which users transfer an object by dragging it from one location to another.

Drop-down combo box - A control consisting of a text box, a list box, and a down-pointing arrow button. Users type in the text box, or they activate the arrow button to display the list and select an item which is displayed in the text box.

Downloaded from http://www.everyspec.com

Drop-down list box - A control consisting of a text-display box, a list box, and a down-pointing arrow button. Users activate the arrow button to display the list and select an item which is displayed in the text-display box.

E

Element cursor - A location cursor that indicates focus in controls such as radio buttons, check boxes, and push buttons.

Ellipsis - A three-dot suffix included in a menu option or control to indicate that additional information is required.

Expand-action button - A push button that, when activated, increases the size of an expandable window to display additional information and/or actions.

Expandable window - A dialog window that can be expanded in size to display additional information and/or actions.

Explicit apply window – A dialog window where user choices take effect only after users have selected OK or an equivalent push button in the window.

Explicit destruction - The requirement that users confirm an action with irreversible negative consequences before the action is executed.

F

Flow chart - A graphic display that presents a schematic representation of the decision points in a sequence or process.

Folder - A container for storing files and applications on the desktop.

Form – A collection of controls (e.g., push buttons, text boxes) on a Web page for interacting with the information on the page.

Frame - A paned area on a Web page.

Function key - A key on the keyboard that performs a single predefined action.

G

Graphic schedule - A graphic display that presents the time sequence for a series of tasks.

Grip handle – The part of a movable menu bar or toolbar that, when dragged, detaches the control and displays it in a separate window.

Downloaded from http://www.everyspec.com

Group box - A frame drawn around a set of controls to indicate they are related.

H

Handle – A location cursor that indicates focus in a graphic object and can be used to manipulate (e.g., move, resize) the object.

Help mode – Context-sensitive information about the object selected by the user.

Home page - A Web page that provides links to all major content areas and support tools in an application.

Hot spot - An area of the pointer that indicates where pointing device operations occur.

Hot zone - An area that the pointer must be within to be considered on an object.

Hyperlink - A technique for navigating between the pages in a Web site.

Hypertext Markup Language - A markup language that describes the structure of a document.

I

Icon - A graphical image that represents an object, window, or application.

Icon box - A window on the desktop that contains the icons for all open windows.

Image map - A graphic image on a Web page that is divided into multiple regions, each of which navigates to a different destination.

Input focus - The window or object within a window to which keyboard input is directed. In explicit mode, focus is assigned by explicit user action, either with the pointing device or from the keyboard; in implicit mode, focus moves with the pointer, and no explicit user action is required.

Instant apply window – A dialog window where user choices take effect immediately, without users selecting OK or an equivalent push button in the window.

Instant help - Context-sensitive information about the object that currently has focus in a window.

K

Keyboard - An input device that enables users to type information and navigate to and select objects in a window.

Downloaded from http://www.everyspec.com

L

Label - Text or a graphic that identifies an object or control or provides information about it.

Line graph - A graphic display that presents trend information, spatially structured information, time-critical information, or relatively imprecise information.

List box - A control for selecting from a list of items.

Location cursor - A graphic indicator that identifies the object to which keyboard input is directed.

M

Maximize button - A window component that, when activated, expands a window to its largest size.

Menu - A list of options from which users choose. The options can execute commands, display a window (to request more information from users) or submenu, and specify settings.

Menu bar - An area at the top of a window where the titles of pull-down menus are displayed.

Menu button - A push button or toolbar button that, when activated, displays a pull-down menu. A menu button includes a triangular down-pointing arrow as part of the button label.

Menu option - The commands, routings, and settings that are included in a menu.

Menu page - A Web page with text and/or graphic links to information pages or other navigation pages in an application.

Menu title - A label in the menu bar of a window where users access a pull-down menu.

Menu window - A window that is displayed when users activate the tear-off button in a menu.

Message window - A secondary window that presents messages to the user.

Minimize button - A window component that, when activated, reduces a window and all of its children to a window icon or taskbar button.

Downloaded from http://www.everyspec.com

Mnemonic - An alphanumeric character in the label of a menu or control that, when used in combination with ALT, navigates to the component and selects or activates it.

Modifier key - A key that, when pressed in combination with another key, changes the meaning of the other key. Modifier keys are ALT, CTRL, and SHIFT.

Mouse-over border – The border of a toolbar button that is visible only when the pointer is over the button.

Mouse-over feedback – A change in the appearance of an object when the pointer is over it.

Multiple-column list box - A list box that displays multiple columns of information for each item in the list.

Multiple document interface - An application that can open more than one file in a window at a time.

N

Navigation - The manner in which users move the pointer or location cursor to the object(s) with which they want to interact.

Navigation bar - An area of a Web page with links that support navigation within an application.

O

Object - A unit of information that is represented visually. An object possesses attributes that define its appearance or state, and it has behavior (i.e., the information it contains can be viewed and manipulated).

Object-action paradigm - An interaction model in which users first select an object (so that it has input focus) and then select an action to perform on the object.

Option menu - A control for selecting from a set of mutually exclusive options. An option menu consists of an option button which, when activated, displays the menu. When users select one of the menu options, it is displayed in the option button.

P

Page footer - An area of a Web page that contains a signature (e.g. the page author) and may also include a navigation bar, security markings, and other identifying information about the page.

Downloaded from http://www.everyspec.com

Page header - An area of a Web page that contains the page name and a navigation bar and may also include a banner graphic, page path, and security markings.

Page path - A navigation aid on a Web page that identifies the sequence of pages that lead to the current page from the home page.

Page title - A label in the title bar of the browser that indicates the name of the Web page displayed in the window.

Paned area – A viewing area in a window that has been divided by a sash or a split box.

Pie chart - A graphic display that provides an approximation of how an entity is apportioned into component parts.

Pointer - A graphic indicator that represents the current location of the pointing device and identifies where pointing device operations occur.

Pointing device - An input device (e.g., a mouse) that enables users to move a pointer on the screen, select objects on which the pointer is placed, and manipulate the objects directly.

Pop-up menu - A menu that is activated at the location of a selected object and contains options relevant to the object.

Pop-up window - A window that does not have a title bar or support standard window management functions. A pop-up window can be used to display a splash screen or a tooltip.

Primary transfer - An interaction technique in which users transfer a selectable object directly to a destination without using the clipboard for temporary storage.

Primary window - A window where most user interactions occur when performing a task in an application. A primary window is the parent for all other windows in its family.

Processing mode - An application state where user actions have different results depending on the mode currently in effect.

Progress bar – A control that fills dynamically to indicate an operation is in progress.

Pull-down menu - A menu that is activated from the menu bar in a window.

Push button - A control that initiates an action.

Q

Downloaded from http://www.everyspec.com

Quick transfer - An interaction technique in which users temporarily select and immediately transfer an object to a new location.

R

Radio button - A control for making a one-of-many selection. A radio button consists of a circular indicator and a label describing the state being set.

Resize border - A window component that, when dragged, changes the size of the window.

S

Sash - A control for adjusting the size of the viewing areas in a paned window.

Scale - A control that sets a value within a predefined ordered range. A scale consists of a scale bar and an indicator for setting a scale value.

Scroll arrow - A part of a scroll bar that, when activated, scrolls incrementally through the information in a window.

Scroll bar - A control that scrolls the information in a window when the information exceeds the space available to display it. A scroll bar contains a scroll bar shaft, a scroll box, and scroll arrows.

Scroll bar shaft - A part of a scroll bar that represents the length of the information that users can scroll.

Scroll box - A part of a scroll bar that indicates the relative position of the information currently displayed. Users drag the scroll box to cause a scrolling action.

Scrolled area – A viewing area in a window that includes horizontal and/or vertical scroll bars.

Search query page - A Web page that allows users to type a keyword and then activate a search engine that locates each instance of the word.

Search results page - A Web page that displays the set of “hits” that match the search parameters entered by the user.

Secondary window - A window that presents supplementary or supporting data or operations. A secondary window can be the child of another window, either primary or secondary. There are two types of secondary windows: dialog windows and message windows.

Downloaded from http://www.everyspec.com

Selection - The manner in which users identify the object(s) with which they want to interact.

Separator - A line that delimits a group of related options or controls in a menu or window, or the different areas of a Web page.

Shortcut icon - An icon that represents a link to another object.

Shortcut key - One or more keys that execute the action of a menu option in the window with input focus.

Single document interface - An application that can open only one file in a window at any given time.

Site index page – A Web page that provides an index of the information in a Web site.

Size grip - A window component that, when dragged, changes the size of a window.

Speed search - A keyboard navigation technique for locating specific text patterns in a list box.

Spin box - A control for selecting from a range of values that have an ordered or natural sequence. A spin box consists of a text area and a pair of arrow buttons that users activate to increase or decrease the value displayed in the text area.

Split bar - A separator that serves as the boundary between the paned areas in a window.

Split box - A control for dividing a paned window into viewing areas and adjusting the size of the areas. A split box is located at the top of a vertical scroll bar or at the left end of a horizontal scroll bar.

Stacking order - An attribute of input focus that determines whether assigning focus also surfaces the window (i.e., raises it to the top of the window stack).

Status bar - An area at the bottom of a window that presents noncritical information to users, provides simple help, and indicates the status or current status of an application.

Submenu - A child menu that is parented by an option in a pull-down or pop-up menu. The parent option includes a right pointing arrow to indicate the presence of the submenu.

Surface chart - A type of line graph that depicts data representing all parts of a whole.

Downloaded from http://www.everyspec.com

T

Tab control - A control that displays multiple views of information within a single window.

Tab group - A control or group of controls to which users navigate by using TAB. Users navigate within a tab group by using the arrow keys.

Tear-off menu - A pull-down menu, pop-up menu, or submenu that can be displayed in a menu window. A menu that can be torn off contains a tear-off button that is the first option in the menu.

Text cursor - A location cursor that indicates where text input is displayed in a window.

Text box - A control for entering and editing text.

Theme – A collection of color and font attributes that define the visual appearance of user interface components.

Title bar - A window component that displays the window title and, when dragged, moves the window to a new location.

Title bar icon - A window component that represents the application or document displayed in a window.

Toolbar – An area of a window containing a collection of toolbar buttons.

Toolbar button - A control that invokes modes, applies settings, and executes actions in a window.

Tooltip – A pop-up window containing information about an object that is displayed when the pointer is placed on the object.

Tooltip help – Context-sensitive information about an object that is displayed using a tooltip.

Tree control - A list box that displays the hierarchical relationship among the objects in the list.

U

Utility window – A dialog window that remains open when the corresponding primary window closes. A utility window is used to display a collection of tools, colors, or patterns.

Downloaded from http://www.everyspec.com

V

Variable function key - A key on the keyboard that performs different actions depending on the current state of an application.

View - The presentation of information about an object.

W

Web page - The text and graphic information that is displayed by a browser.

What's This? button - A window component that, when activated, displays context-sensitive help.

Window - The portion of the display screen delimited by a border in which information is displayed.

Window button – A button in the title bar of a window that performs a window management function (e.g., minimize, maximize, close).

Window family - A parent window and all of its subordinate child windows.

Window frame - A window component that surrounds the client area and provides access to window management functions in a window.

Window icon - A representation of a minimized window.

Window mode - A mode assigned to a secondary window that determines the extent to which users can interact with other windows while the window is open. Users can interact with other windows when a modeless window is open, but are restricted from interacting with other application and/or system windows when a modal window is open.

Window title - A label in the title bar that indicates the name of a window.

Z

Zoom button - A control for adjusting the size of the viewing areas in a paned window.

Downloaded from http://www.everyspec.com

This page intentionally left blank.

Downloaded from http://www.everyspec.com

Appendix F: Style Requirements for COE Compliance

This appendix identifies the COE style specifications[1] that must be satisfied in order to be considered compliant with the style-related compliance checklist items in version 4.3 of the COE I&RTS. The mapping of specifications to checklist items is intended to define a migration path for existing software to become fully COE-compliant in the style area. New software is expected to be developed in accordance with COE requirements for style implementation (see section 1.4.2) and to be fully compliant with COE style specifications.

Standards Compliance (Level 1)

Item 1-11. The application complies with the style of the native GUI. (See GUI compliance requirements in the *COE UIS*.)

- The application complies with the minimum COE requirements for style implementation in section 1.5.3.
- The application complies with (or has an approved migration plan to comply with) COE specifications for:
  - Input devices in section 2.
  - User-computer interaction in section 3.
  - Window management in section 4.
  - Menus in section 5.
  - Controls in sections 6 and 7.
  - Action vocabulary, mnemonics, and shortcut keys in appendix A.

Minimal COE Compliance (Level 5)

5-30. The segment is fully compliant with the style of the native GUI as defined in the *COE UIS* document.

- The segment is fully compliant with level 1 style specifications.
- The segment complies with (or has an approved migration plan to comply with) COE specifications for:
  - Window design in section 8.
  - Map windows in section 14.

5-62. The segment supports HTML 3.2 and complies with style specifications (see the *COE UIS*) for Web applications.

5-63. The segment provides a notification to "disadvantaged" users if they are using a browser that does not support the features provided by the segment.

- The segment complies with (or has an approved migration plan to comply with) COE specifications for:
  - Web page design in section 11.
  - Web site design in section 12

[1] The content of section 17 on user interface internationalization is not mapped to a COE compliance level.

Downloaded from http://www.everyspec.com

5-66. The desktop is configured in accordance with the *COE UIS*.

- The segment complies with (or has an approved migration plan to comply with) COE specifications for:
  - Integration with the desktop in section 10.2.

Intermediate COE Compliance (Level 6)

6-7. The segment is compliant with the *COE UIS* or has minimal deviations that have been approved by the Chief Engineer. "Minimal" is defined in the *COE UIS* and in the absence of *COE UIS* direction, at the discretion of the Chief Engineer.

- The segment is fully compliant with level 5 style specifications.
- The segment complies with (or has an approved migration plan to comply with) COE specifications for:
  - User support resources in section 9.
  - Information presentation in section 13.
  - Interacting with drag-enabled objects in section 15.

Full COE Compliance (Level 8)

8-8. The segment is fully compliant with the *COE UIS*.

- The segment is fully compliant with level 6 style specifications.
- The segment complies with COE specifications for:
  - Application design in section 10.1 if the segment provides an application-oriented user interface.
  - Object-oriented design in section 16 if the segment provides an object-oriented user interface.

Downloaded from http://www.everyspec.com

Index

Acronyms and Abbreviations C-1
Action Vocabulary A-1
Activation 3-5
Animation on Web Pages 11-10
Applets on Web Pages 11-10
Application Configuration 10-5
Application Design 10-1
Application Design in Object-Oriented Interfaces 16-3

Bar Charts 13-4

Capitalization 10-6
Cascading Style Sheets 11-6
Check Boxes 6-6
Clipboard Transfer 3-6
Color Attributes, Management of 10-7
Combo Boxes (GNOME, MS Windows) 7-5
Compliance with COE Requirements 1-5
Compliance with COE Requirements, Minimum 1-6
Compliance with Federal Accessibility Standards 1-8
Compliance with Microsoft Certification Requirements 1-8
Conditional Text Boxes 7-3
Content Pages 12-6
Control Menu 4-3
Controls 6-1, 7-1

Data Entry Pages 12-5
Date/Time Formats 13-1
Dialog Window Content 8-9
Dialog Window Design 8-9
Dialog Window Titles 8-9
Dialog Windows, Types of 8-14
Drag Transfer 3-6
Drag-Enabled Objects, Interacting with 15-1
Drop-Down Combo Boxes 7-7
Drop-Down List Boxes 6-12

Error Detection 3-10
Explicit Destruction 3-10

Federal Accessibility Standards B-1
Feedback, Immediate 3-9
Flow Charts 13-5
Font Attributes, Management of 10-7
Frames 11-5

Downloaded from http://www.everyspec.com

Function Keys, Fixed 2-5
Function Keys, Variable 2-7

Glossary E-1
Graphic Schedules 13-6

Help, Application 9-3
Help, Context-Sensitive 9-1
Help, Tooltip 9-1
Help Dialog Windows 9-2
Help Pop-Up Windows 9-2
Home Pages 12-4

Icons 10-5
Image Maps on Web Pages 11-9
Images on Web Pages 11-9
Information, Coding of 13-8
Information, Dynamic 13-12
Information, Graphical 13-3
Information, Text 13-1
Information, Presentation of 13-1
Input Devices 2-1
Input Devices, Alternate 2-10
Input Devices, Interchangeability Between 2-9
Input Focus 4-5
Input Focus, Management of 10-8
Interactive Control 3-9
Internationalization 17-1

Keyboard Input 2-5

Latitude/Longitude Formats 13-1
Layout and Visual Alignment 10-8
Line Graphs 13-3
Link Titles 11-3
Links 11-1
Links, Embedded 11-2
Links, Structural 11-2
List Boxes 6-9
List Boxes, Multiple-Column 6-10
List Boxes, Multiple-Selection 6-11

Map Windows 14-1
Menu Bars 8-2
Menu Design 5-8
Menu Pages 12-4
Menus 5-1
Message Window Content 8-17

Downloaded from http://www.everyspec.com

Message Window Design 8-16
Message Window Titles 8-16
Message Windows, Types of 8-19
Mnemonics 2-5
Multimedia on Web Pages 11-10
Multiple-Document Interface Applications (MS Windows) 10-2

Navigation 3-1
Nested Applications 10-3

Object Concepts 16-1
Object Icons 16-1
Object Model 16-3
Object Types 16-1
Object Views 16-1
Object-Action Paradigm 3-9
Object-Oriented Design 16-1
Option Menus (GNOME, Motif) 6-14

Page Footers 11-1
Page Headers 11-1
Page Paths 11-3
Page Titles 11-1
Paned Areas 7-16
Parent-Child Relationships in Windows 4-11
Pie Charts 13-6
Pointers 2-1
Pointer Shapes 2-1
Pointing Device Buttons 2-3
Pointing Device Features, Management of 10-8
Pointing Device Input 2-1
Pop-Up Menus 5-4
Primary Transfer (Motif) 3-7
Primary Window Content 8-2
Primary Window Design 8-1
Primary Window Titles 8-1
Processing Modes 3-11
Progress Bars 8-18
Pull-Down Menus 5-1
Push Buttons 6-1

Quick Transfer (Motif) 3-7

Radio Buttons 6-4

Scales 7-11
Scrolled Areas 7-14
Search Query Pages 12-5

Downloaded from http://www.everyspec.com

Search Results Pages 12-5
Selection 3-3
Shortcut Icons 16-2
Shortcut Keys 2-6
Single-Document Interface Applications 10-1
Site Index Pages 12-5
Source Documents 1-11
Spin Boxes (GNOME, Motif, MS Windows) 7-10
Stacking Order 4-6
Status Bars 8-8
Style Management 10-7
Submenus 5-6
Surface Charts 13-3

Tab Controls 7-12
Tactical Graphics 10-7
Tactical Information, Coding of 10-8
Tactical Symbology 10-7
Terminology D-1
Text Boxes 7-1
Text Boxes, Conditional 7-2
Text Entry 2-5
Text-Display Boxes 7-2
Text on Web Pages 11-7
Toolbar Buttons 6-7
Toolbars 8-7
Transfer 3-6
Tree Controls 6-16

Undo Capability 3-11
User Error, Minimizing 3-9
User Support Resources 9-1
User-Computer Interaction 3-1

Web Page Content 11-7
Web Page Design 11-1
Web Page Design Considerations 11-11
Web Page Design Templates 11-4
Web Page Layout Tables 11-4
Web Site Design 12-1
Web Site Design Considerations 12-8
Window Buttons (GNOME, MS Windows) 4-13
Window Components 4-1
Window Design 8-1
Window Families 4-11
Window Icons (Motif) 4-14
Window Location 4-15
Window Management 4-1

Downloaded from http://www.everyspec.com

Window Management Considerations 4-15
Window Mode 4-11
Window Size 4-15

Downloaded from http://www.everyspec.com

This page intentionally left blank.