INTERNATIONAL STANDARD

ISO 9241-143

First edition
2012-03-01

Ergonomics of human-system interaction —

Part 143: **Forms**

Ergonomie de l'interaction homme-système —

Partie 143: Formulaires

Reference number
ISO 9241-143:2012(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 9241-143 was prepared by Technical Committee ISO/TC 159, *Ergonomics*, Subcommittee SC 4, *Ergonomics of human-system interaction.*

This first edition of ISO 9241-143 cancels and replaces ISO 9241-17:1998, of which it constitutes a technical revision.

ISO 9241 consists of the following parts, under the general title *Ergonomic requirements for office work with visual display terminals (VDTs)*:

— *Part 1: General introduction*

— *Part 2: Guidance on task requirements*

— *Part 4: Keyboard requirements*

— *Part 5: Workstation layout and postural requirements*

— *Part 6: Guidance on the work environment*

— *Part 9: Requirements for non-keyboard input devices*

— *Part 11: Guidance on usability*

— *Part 12: Presentation of information*

— *Part 13: User guidance*

— *Part 14: Menu dialogues*

— *Part 15: Command dialogues*

— *Part 16: Direct manipulation dialogues*

— *Part 17: Form filling dialogues*

ISO 9241 also consists of the following parts, under the general title *Ergonomics of human-system interaction*:

— *Part 20: Accessibility guidelines for information/communication technology (ICT) equipment and services*

— *Part 100: Introduction to standards related to software ergonomics* [Technical Report]

— *Part 110: Dialogue principles*

— *Part 129: Guidance on software individualization*

— *Part 143: Forms*

— *Part 151: Guidance on World Wide Web user interfaces*

— *Part 154: Interactive voice response (IVR) applications*

— *Part 171: Guidance on software accessibility*

— *Part 210: Human-centred design for interactive systems*

— *Part 300: Introduction to electronic visual display requirements*

— *Part 302: Terminology for electronic visual displays*

— *Part 303: Requirements for electronic visual displays*

— *Part 304: User performance test methods for electronic visual displays*

— *Part 305: Optical laboratory test methods for electronic visual displays*

— *Part 306: Field assessment methods for electronic visual displays*

— *Part 307: Analysis and compliance test methods for electronic visual displays*

— *Part 308: Surface-conduction electron-emitter displays (SED)* [Technical Report]

— *Part 309: Organic light-emitting diode (OLED) displays* [Technical Report]

— *Part 310: Visibility, aesthetics and ergonomics of pixel defects* [Technical Report]

— *Part 331: Optical characteristics of autostereoscopic displays* [Technical Report]

— *Part 391: Requirements, analysis and compliance test methods for the reduction of photosensitive seizures*

— *Part 400: Principles and requirements for physical input devices*

— *Part 410: Design criteria for physical input devices*

— *Part 411: Evaluation methods for the design of physical input devices* [Technical Specification]

— *Part 420: Selection of physical input devices*

— *Part 910: Framework for tactile and haptic interaction*

— *Part 920: Guidance on tactile and haptic interactions*

User-interface elements, human-centred design and evaluation methods, ergonomic requirements for the reduction of visual fatigue from stereoscopic images, and the evaluation of tactile and haptic interactions are to form the subjects of future parts 161, 230, 392 and 940.

Introduction

This part of ISO 9241 is concerned with the ergonomic design of forms.

Forms, including dialogue boxes, are appropriate for data entry tasks requiring input or modification of multiple data items. Forms are used in various circumstances, including

— filling forms, such as income tax forms, registration (school, motor vehicle), and service order completion,

— entering information received over the telephone,

— interactively populating data in an application, such as database updates, consumer profiles and e-commerce transactions,

— specifying the application options and parameters (complex data retrieval requests, personalisation, system configurations settings), and

— responding to a mediate request for system information (e.g. using a dialogue box).

Forms can vary in content and complexity from a simple field to complex data entry that involves multiple data records. Forms are often based on a visual spatial metaphor but can be implemented in other modalities (e.g. voice user interfaces over the telephone).

Form users fill-in, select entries for, modify fields and/or use, the form to retrieve information from the system.

This part of ISO 9241 is aimed at

a) user-interface designers, who will apply it during the development process,

b) the designers of printed forms which serve as source documents,

c) buyers, who will reference it during the product procurement process,

d) evaluators responsible for ensuring products meeting its requirements and recommendations,

e) the designers of development tools to be used by interface designers, and

f) end users, who will gain from the potential benefits it provides.

This part of ISO 9241 provides requirements and recommendations concerning forms. Some of these are conditional with respect to whether they are relevant in terms of context of use variables such as particular kinds of users, tasks, environments or technology.

Designers using this part of ISO 9241 ought to be able to determine whether they are developing an interface that will meet those of the standard's requirements and recommendations that are applicable. Likewise, buyers and evaluators ought to have a means of determining how a product matches the applicable requirements and recommendations. It is not intended that every requirement and recommendation given in this part of ISO 9241 be applied, only those that are relevant. Annex B provides an example of a procedure for evaluating the applicability of, and conformance with, the requirements and recommendations.

The application of this part of ISO 9241 is expected to improve the overall quality of the form, but this International Standard (like any other standard) will not guarantee the quality of the interface. Quality depends on specific usability criteria as set by the user, buyer or other form consumer, which may include specifications based on this part of ISO 9241.

ISO 9241-110 describes dialogue principles that are relevant for the design of forms. The principles provide the designer and evaluator with additional information concerning the ergonomic rationale for the various recommendations given in this part of ISO 9241 and, therefore, assist in making trade-offs. However, it is often necessary to base trade-offs on other considerations as well.

Ergonomics of human-system interaction —

Part 143:
Forms

1 Scope

This part of ISO 9241 provides requirements and recommendations for the design and evaluation of forms — in which the user fills-in, selects entries for, or modifies labelled fields on, a "form" or dialogue box presented by the system. Often the system then creates or updates the data associated with the form. Form-based entries typically are in the form of typed input (abbreviations, or full names) or selections from available option lists.

This part of ISO 9241 is applicable to forms regardless of the modality in which they are rendered (visual, spatial, vocal). However, much of the guidance is based on a model of visual and spatial relationship.

In addition, this part of ISO 9241 specifies the use of non-text methods for providing forms entries (e.g. list boxes) and pertains to dialogue boxes which utilize form techniques. Guidance is provided on the selection and design of those user-interface elements relevant to forms.

While lists used to enter forms data are covered in this part of ISO 9241, menus which are similar to lists are outside its scope but are covered in ISO 9241-14. Neither is this part of ISO 9241 applicable to the hardware aspects of forms.

NOTE Some of the requirements and recommendations in this part of ISO 9241 are based on Western Language conventions. For other languages, particular requirements or recommendations might need to be modified to fit the readability and/or text input considerations inherent in these languages.

The requirements and recommendations in this part of ISO 9241 are applicable throughout the development process — for example, as guidance for designers during design, as a basis for heuristic evaluation, as guidance for usability testing — and in the procurement process.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 9241-12:1998, *Ergonomic requirements for office work with visual display terminals (VDTs) — Part 12: Presentation of information*

ISO 9241-16, *Ergonomic requirements for office work with visual display terminals (VDTs) — Part 16: Direct manipulation dialogues*

ISO 9241-171:2008, *Ergonomics of human-system interaction — Part 171: Guidance on software accessibility*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1
analogue form element
user-interface element that allows the user to select a value within a continuous range

EXAMPLE Rotary dials, slider.

3.2
check box
check button
user-interface element used to select an option that can be either {“on” or “off”} or {“yes” or “no”} with a label (usually text) that indicates what the option is and an indicator (e.g. graphic checkmark within the individual check box) that indicates whether or not the option is selected

NOTE 1 Check boxes in a group are independent, i.e. selecting one option does not affect the selection of other options in that check box group. In a group of check boxes, one or many of them can be checked at the same time.

NOTE 2 Contrast with **radio button** (3.20).

3.3
combination box
combo box
user-interface element that combines a text field with a list box and allows the user to type input into a text field or to select an option from a list box, which will fill in the text field

NOTE 1 A combination box typically has a label (textual or graphical) that indicates the purpose of the combination box.

NOTE 2 Contrast with **list box** (3.15).

3.4
default action
operation predefined for an object or set of objects (including an entire dialogue box) that will take place when the user actuates the default action mechanism

3.5
dialogue box
subordinate form (typically presented in a separate window) that supplements and/or supports the interaction that is taking place in the main application dialogue

NOTE If a message requires user input it could be considered a dialogue box.

3.6
dialogue
interaction between a user and an interactive system as a sequence of user actions (inputs) and system responses (outputs) in order to achieve a goal

NOTE 1 User actions include not only entry of data but also navigational actions of the user.

NOTE 2 Dialogue refers to both the form (syntax) and the meaning (semantics) of interaction.

[ISO 9241-110:2006]

3.7
entry field
input field
field in which users can input data or edit displayed data

NOTE 1 Entry fields can be optional or mandatory (required) fields for the user.

NOTE 2 Content can be prepopulated with a default value relevant to the task.

NOTE 3 Data in entry fields could be numerical, textual, alphanumerical, etc.; see also **text field** (3.29) and ISO 9241-12:1998, 3.5.

[ISO 9241-12:1998, 3.5.1]

3.8
field
user-interface element in which data is entered or presented

NOTE Adapted from ISO 9241-12:1998.

3.9
form
structured display of fields and other user-interface elements that the user reads, fills in, selects entries for (e.g. through check boxes or radio buttons) or modifies

3.10
form element
user-interface element applicable to forms

3.11
form structure
organization and arrangement of information on individual and sets of forms, and the interconnections among forms for user navigation

3.12
hierarchical list
series of lists which are structured in a hierarchical or “tree-like” manner, where the selection of the initial item leads to another list containing additional items which can lead to another list, etc., until the desired item is located

3.13
implicit designator
portion of an option name or control label used for keyboard selection

[ISO 9241-171:2008, 3.16]

3.14
label
short descriptive title for an entry or read-only field, table, control or other user-interface element

NOTE 1 In some applications, labels are classified as protected fields.

NOTE 2 Adapted from ISO 9241-12:1998, 3.9.

3.15
list box
user-interface element that provides a presentation (usually vertical) of items from which the user can select only one (single selection) or more than one (multiple selection)

NOTE 1 The items in the list can be represented by a text and/or graphic label.

NOTE 2 One particular instance is the drop-down list box. When a drop-down list is inactive, it displays a single value. When activated, it displays (drops down) a list of values, from which the user can select one. When the user selects a new value, the user-interface element reverts to its inactive state, displaying the selected value.

3.16
list button
user-interface element used to access a list whereby when the button is activated, a pop-up list of items is displayed

NOTE The list button contains the name of the currently selected list item and has a graphic (usually a bar) indicating that additional selections are available.

3.17
name
word or phrase associated with a user-interface element and that is used to identify the element to the user

[ISO 9241-171:2008]

3.18
navigation
⟨forms⟩ ability to move from one user-interface element to another within a form, to move throughout a form and to move from form to form

3.19
push button
command button
user-interface element used for executing an immediate command or action

NOTE A push button typically contains a label, which can be textual and/or graphical.

3.20
radio button
user-interface element used to select an option from a group of mutually exclusive options, which has a (usually text) label that indicates what the option is and an indicator (e.g. graphic dot) that indicates whether or not the option is set

NOTE 1 Radio buttons exist in groups used for a "one of many" choice, where exactly one option in the group can be selected at any time, and selecting one results in the deselection of any different previously selected option.

NOTE 2 Contrast with **check box** (3.2).

3.21
read only field
protected field
field that contains data that cannot be modified by the user

NOTE Adapted from 9241-12:1998, 3.5.2.

3.22
scroll bar
user-interface element that allows a user to view objects that extend beyond the size of a displayed related window or list by moving them into or out of the available display area

NOTE 1 A scroll bar for an area indicates whether or not additional information is available and can also indicate the relative position of the displayed part of the information.

NOTE 2 Adapted from 9241-16:1999, 3.19.

3.23
scroll box
elevator
thumb
user-interface element in the shape of a rectangle within a scroll bar that allows the user to move to a specific region of a displayed file by dragging the rectangle to the appropriate location within the scrollbar (shaft)

NOTE The length of the scroll box typically denotes the relative amount of material in the file (e.g. form).

3.24
selection cursor
indicator of an item whose selection state can be changed by a defined confirming action

3.25
selection indication
visual or other cue that indicates the selected element on the display, to which the user can apply a subsequent action

[ISO 9241-16:1999, 3.21]

3.26
selection list
choice list
user-interface elements presented as a list containing a number of items from which a user can select

NOTE 1 Single selection or multiple selection can be possible. The number of items can be fixed or can change during the dialogue.

NOTE 2 Selection lists are often presented in a box, i.e. list box 3.15.

3.27
stepper
user-interface element that allows the user to move through the available alternatives, options or values, one at a time

NOTE 1 A stepper typically has an area that displays the current value and one or two graphical elements that allow the user to move through the options.

NOTE 2 Steppers can be combined with data entry fields.

NOTE 3 There are various types of stepper elements including spin buttons, cycle buttons, little arrows and sliders.

3.28
tab set
user-interface element that uses a metaphor of a set of index cards with tabs to identify them

NOTE Each tab in a tab set has an associated set of displayed information and/or controls. Selection of a different tab within the tab set allows users to move among the various sets of information and options.

3.29
text field
user-interface element that allows the user to enter character based data

NOTE A label can be used to indicate the type of information to be provided in the field.

3.30
toggle button
user-interface element that provides a choice between two states

NOTE A toggle button's two states, set and unset, are typically shown by its appearing “pushed in” or “popped out” respectively.

3.31
user-interface element
user-interface object
entity of the user interface that is presented to the user by the software

EXAMPLE Text, graphic, control.

NOTE 1 User-interface elements can be interactive or not.

NOTE 2 Both entities relevant to the task and entities of the user interface are regarded as user-interface elements. A user-interface element can be a visual representation or an interaction mechanism for a task object (such as a letter, sales order, electronic part or wiring diagram) or a system object (such as a printer, hard disk or network connection). It can be possible for the user to directly manipulate some of these user-interface elements.

NOTE 3 User-interface elements in a graphical user interface include such things as basic objects (such as window title bars, menu items, push buttons, image maps, and editable text fields) or containers (such as windows, grouping boxes, menu bars, menus, groups of mutually-exclusive option buttons, and compound images that are made up of several smaller images). User-interface elements in an audio user interface include such things as menus, menu items, messages, and action prompts.

[ISO 9241-171:2008, 3.38]

4 Forms

4.1 Selection

Forms, including dialogue boxes, should be used for structured data entry tasks requiring input or modification of multiple data items.

A major use of forms is in entering information into forms displayed on a computer. Also, forms are often used for data entry into a computer from a paper source document. Examples include income tax forms, registration (school, motor vehicle) and service order completion. Another use of forms is for entering information received over the telephone.

Dialogue boxes are commonly used for specifying application options and parameters. Also, dialogue boxes are often appropriate for certain complex data retrieval requests where the user might find it easier to fill in parameter information than to input the parameters via a command language.

Interface design depends upon the task, the user, the environment and the available technology. Consequently, this part of ISO 9241 cannot be applied without knowledge of the design and use context of the interface and it is not intended to be used as a prescriptive set of rules to be applied in their entirety (see ISO 9241-11 and ISO 9241-210). Rather, it assumes that the designer has proper information available concerning task and user requirements and understands the use of available technology (this may require consultation with a qualified ergonomics professional as well as empirical testing with real users).

4.2 General requirements and recommendations

4.2.1 Form titles

Forms shall be titled unless a title would be redundant (e.g. if it is provided by the surrounding user interface). Any title shall clearly indicate the purpose of the form and differentiate it from other forms.

NOTE 1 In visually displayed forms, titles are usually placed at the top of the form (or page in the case of a web-based application).

NOTE 2 In window-based interfaces, the title of a form can be the window title, if the form is the only content of the window.

NOTE 3 Forms are typically embedded in larger user interfaces, such as applications.

4.2.2 Relationship of form title to application structure

The title should provide the user with a sense of the location of the form within the application structure.

4.2.3 Visual coding

If the task requires, or is enhanced by, discrimination between user entries, defaults and previously entered data, and the form is presented visually, distinctively different visual coding should be used.

Distinctively different visual coding shall not be limited to colour (see ISO 9241-171:2008, 10.4.1).

4.2.4 Appearance of form elements

If elements have different states, the current state of the element shall be clearly indicated by a perceptible cue.

EXAMPLE 1 Inactive elements are dimmed.

EXAMPLE 2 In voice-based interaction, unavailable elements might not be presented but still be responded to by the system to tell the user that the option is inactive.

EXAMPLE 3 A check box has a check or tick in it to show it has been selected.

NOTE See ISO 9241-171:2008, 8.5.4, on providing information on a state to assistive technology.

4.2.5 Form display density

Unless required by the task or context of use, forms should limit the density of the textual information displayed.

NOTE For character density within forms, the limit of 40 % overall density, based on a percentage of the total available form space filled, is appropriate (see ISO 9241-12:1998, 5.4.2).

4.2.6 Complexity

Complexity in dialogue boxes and forms should be appropriate for the task. If a dialogue box or form supports basic functionality, as well as advanced, auxiliary or less frequently used functionality, it should use one of the following to reduce the complexity:

a) an expandable form to present the advanced, auxiliary or less frequently used functionality;

b) additional forms or dialogue boxes with advanced, auxiliary or less frequently used functions, accessible via controls within the higher level dialogue box or form;

c) a multiple-page dialogue box, with advanced, auxiliary or less frequently used functions on later pages (e.g. tabbed dialogues, scrollable dialogues).

4.2.7 Restricting use of expanded dialogues

Dialogue box expansions and additional dialogues should be restricted to functions that are needed only by a subset of users or that are not needed for the typical execution of a task.

4.2.8 Instructions

Access to instructions should be provided.

NOTE 1 Instructions are particularly important for helping the user to navigate through, complete, save and transmit the form.

NOTE 2 Instructions are helpful for infrequent users and users unfamiliar with the form.

NOTE 3 A list at the beginning of the form as to the information required to complete the form is particularly useful for blind or elderly users.

NOTE 4 Instructions are helpful for any user-interface element that the user might be unfamiliar with.

NOTE 5 It is preferable to place instructions where they can most easily be used (see ISO 9241-13), without their getting in the way of completion of the task when not needed.

4.2.9 Help

If the user of the form needs further information on how to complete an entry in the form, access to completion assistance or Help should be provided.

EXAMPLE 1 When the user puts the cursor on the entry field, a pop-up message containing context-sensitive help is provided.

EXAMPLE 2 The user clicks on a Help button to obtain specific guidance for completing the entry in the form.

NOTE Completion assistance is particularly helpful for blind or elderly users and non-native language speakers.

4.2.10 Overview of structure

If beneficial to task performance and if the form structure is complex, an overview of the form structure or a visual presentation of the structure should be provided to the user.

EXAMPLE A flow diagram showing the form's structural components is provided upon request by the user.

4.2.11 Use of modal and modeless dialogue boxes

Unless it is necessary for command completion, or if important to prevent further interaction until a condition is satisfied, modeless rather than modal dialogue boxes should be used.

NOTE 1 In modal dialogue boxes, the user completes the dialogue and closes it before continuing with the application. These dialog boxes are best used for critical or infrequent, one-off tasks that require completion before continuing.

NOTE 2 In modeless dialogue boxes, the user is able to switch between the dialog box and the application as desired. These dialog boxes are best used for frequent, repetitive, on-going tasks.

EXAMPLE 1 A modeless dialogue is used to allow the user to set preferences. This dialogue allows the user to navigate within the application to determine the impact of the settings as they are made.

EXAMPLE 2 A modal dialogue is presented to prompt the user to save data that would otherwise be lost when the user is closing a window.

4.2.12 Accessibility

Forms, dialogue boxes and their elements shall be in accordance with ISO 9241-171.

NOTE Conformity with ISO 9241-171 implies that every applicable requirement and every applicable recommendation is met. See ISO 9241-171:2008, 7.2 for details.

4.2.13 Consistency

Behaviour (conceptual, semantic and lexical) of forms should be consistent through platforms and systems (e.g. for mobility… PCs, PDAs, Kiosks), wherever possible.

5 Information presentation

5.1 General

ISO 9241-12 shall be consulted for general guidance on information presentation concerning forms.

NOTE Many of the recommendations in Clause 5 are intended for forms presented in the visual modality and are based on left-to-right writing.

5.2 Layout

5.2.1 Paper document source

If a paper document is used as the source for computer input and the form is to be presented visually, the forms screen should be designed to be consistent with the structure of the paper source document in terms of item ordering, grouping, units for input values (e.g. millimetres or metres).

NOTE Compatibility with paper source documents is an important layout consideration. However, if the layout of the paper source document is not compatible with efficient task performance, redesign of the source document is worthy of consideration. If there are conflicting requirements between those filling in the paper forms (e.g. customers) and those filling in the computer forms, it can be more important to ensure those filling in the paper forms achieve the highest efficiency at the expense of the computer input.

5.2.2 No source document

If forms do not depend upon a source document, the following is recommended.

a) The arrangement of user-interface elements in the dialogue should be based on the sequence that is appropriate for the written language of the user who will use the dialogue.

   EXAMPLE 1 For an English-based dialogue, the elements are arranged from left to right and top to bottom.

b) Entry fields should be grouped by, e.g. function or importance, or optimized based on input sequences from the user's point of view.

   EXAMPLE 2 In a telephone sales transaction, the form is dynamically modified on the basis of customer choices to guide the sales person in completing the transaction.

   NOTE 1 If data is supplied by a customer, the sequence can depend on customer needs.

   NOTE 2 See ISO 9241-12:1998, 5.6 for guidance on grouping.

5.2.3 Order of required fields and optional fields

If the form contains both required fields and optional fields within a functional or logical grouping of fields, required fields should be positioned first, unless such positioning is inappropriate to the user's task (e.g. it would not be consistent with a paper source document).

NOTE If users include the elderly, separating required and optional fields is particularly important, since elderly users often do not recognize coding typically used (e.g. the asterisk) for required fields.

5.2.4 Alphanumeric field alignment

If the form is presented visually and if appropriate to the language context, alphanumeric entry fields should be aligned vertically in one or more columns and left-justified within each column.

NOTE This will improve visual scanning and often minimizes the keystrokes required to move between fields.

5.2.5 Numerical field alignment

If the form is presented visually and if groups of entry fields are all numeric and field lengths are different, the content of the fields should be displayed right-justified. If numerical fields contain decimal markers, they should be aligned to the decimal marker.

5.2.6 Allowable field values

Information should be provided indicating allowable field values (i.e. either display the information on the form or on demand).

5.3 Names and labels

5.3.1 Consistency

Naming of labels should be consistent through different platforms and systems (e.g. for mobility… PCs, PDAs, Kiosks).

5.3.2 Naming elements and groups of elements

All elements and groups of elements shall have a name, whether or not the label is visually displayed (see ISO 9241-171:2008, 8.1).

NOTE 1 Names can be utilized by a screen reader even if the corresponding label is not displayed.

NOTE 2 In many cases, both are handled as attribute (or property) of user-interface element.

5.3.3 Label visual design

ISO 9241-12:1998, 5.9, shall be consulted for general guidance on visual design of labels.

5.3.4 Differing label lengths

If text or alphanumeric fields are aligned vertically in columns, label lengths could differ significantly and the task involves sequential data entry, labels should be right-justified and fields should be left-justified, or the fields should be left-justified and the labels placed above the fields and left-justified with the fields.

NOTE 1 Significant differences can be defined as a condition in which the horizontal dimension of the longest label is twice that of the shortest label

EXAMPLE 1 In Figure 1, the field labels differ significantly and the labels are right justified.

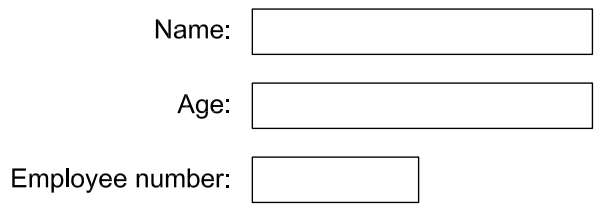

Figure 1 — Differing label lengths with labels to the left of the entry

EXAMPLE 2 In Figure 2, the field labels differ significantly and the labels are placed above the fields.

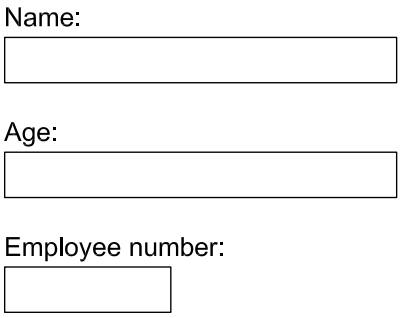

Figure 2 — Differing label lengths with labels above entries

NOTE 2 Generally, top-positioned labels work well for short forms that are familiar to the user.

5.3.5 Similar label lengths

If text or alphanumeric fields are aligned vertically in columns and field label lengths do not differ significantly, field labels and fields both may be left-justified.

EXAMPLE Since “Name” and “Age” do not differ significantly in length, the labels are left-justified, see Figure 3.

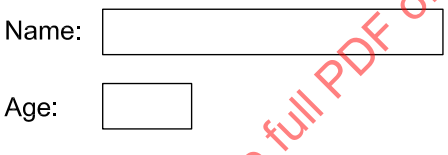

Figure 3 — Labels and fields left justified

NOTE It is important to consider that the space at magnification levels can be significantly larger than at the normal viewing level.

5.3.6 Label placement for search tasks

If the user's task involves searching a group of labels for a particular label, then field labels should be left-justified.

5.3.7 Consistent label positions

Labels should be consistently positioned for the type of user-interface element or group of elements within a given context.

5.3.8 Label position for check boxes or radio buttons

Labels for check boxes or radio buttons should be consistently located to the right, unless the button is large enough for the text to be located inside the button.

5.3.9 Labels for screen reader use

If the label name is to be read by a screen reader, the individual check box or radio button label name should include the label name of the group.

NOTE See ISO 9241-171 for accessibility issues concerning names and labels.

5.3.10 Groups of fields

If fields are grouped, the label of the group should be placed at the top of the group.

5.3.11 Multiple instances of fields in a matrix

If a label is used for multiple instances of fields in a matrix (e.g. table), the label should be located above the column (see Example 1) or to the left of the row (see Example 2).

EXAMPLE 1 In Figure 4, the labels, “Last Name”, “First Name” and “Age in Years” are displayed above the columns.

Last Name	First Name	Age in Years
Williams	Peter	35
Collins	Robert	41
Robertson	William	26
Hofmann	David	32

Figure 4 — Labels located above columns

EXAMPLE 2 In Figure 5, the labels, “Last Name”, “First Name” and “Age in Years” are displayed to the left of the rows.

Last Name	Williams	Collins	Robertson	Hofmann
First Name	Peter	Robert	William	David
Age in Years	35	41	26	32

Figure 5 — Labels located to the left of rows

5.3.12 Descriptive field labels

All fields should be clearly and unambiguously labelled to describe the purpose or function of content of each.

5.3.13 Distinctive field labels

Distinctive words and/or codes (e.g. position, border, font, colour) should be used for entry field labels, and the approach used should be applied consistently throughout the form so that entry fields will not be confused with data, instructions, etc.

NOTE See ISO 9241-12 and ISO 9241-171 for guidance on limiting the exclusive use of colour.

5.3.14 Consistent labels

The mapping between the labels and functions for form elements should be consistent across a software application product or service.

EXAMPLE Table 1 depicts the element label/functionality of the “OK”, “Apply” and “Cancel buttons".

Table 1 — Element label/functionality

Element label	Functionality
OK	Applies any changes made to the elements in the dialogue box and dismisses the dialogue box.
Apply	Applies any changes made to the elements in the dialogue box and leaves the dialogue box up (i.e. does not dismiss the dialogue box).
Cancel	Dismisses the dialogue box without performing any changes that have not yet been applied.

5.3.15 Symbols or units

Symbols or units ($, f, %, mph, cm, l, etc.) should be displayed as additional labels when required by the user for interpretation of the data in an entry field, see ISO 9241-12:1998, 5.9.7.

NOTE The symbol or unit can be added to the column label in the case of a column of fields.

5.3.16 Initial upper-case letter for field labels

To facilitate readability, text field labels in English should begin with an upper-case letter. The rest of the label should contain lower-case (small) letters, except for instances where the label is a logo, an acronym, a numeric value, or language convention requires each word in the label to begin with a capital letter.

NOTE For languages other than English other rules of writing can apply. For example, in German all nouns begin with an upper-case letter.

5.3.17 Implicit designator for elements

To facilitate user navigation within the form or dialogue box, an implicit designator should be assigned to the label of each element.

NOTE Since each check box in a group represents a separate element, an implicit designator would need to be assigned for each check box in a group of check boxes.

5.3.18 Multiple pages

5.3.18.1 In a sequence of multiple forms, the location within the sequence shall be provided and placed at the same location on each form.

EXAMPLE 1 "Page 1 of 3" is positioned close to the title at the top of the form.

EXAMPLE 2 The list of forms is presented and the current form is highlighted in this list.

5.3.18.2 If the form is columnar, the labels of the columns should be redisplayed.

5.4 Visual cues in fields and forms elements

5.4.1 Design considerations

Cueing should be provided appropriate to the modality of presentation.

NOTE The recommendations concerning visual cues in fields and form elements are intended for forms presented in the visual modality.

EXAMPLE Grouping fields by distance/spacing provide the cue that they belong to each other in a visual or tactile modality. Auditory modality use pauses for different groups.

5.4.2 Fields with fixed length entry

If the text entry fields are of fixed length, lengths should be explicit (see ISO 9241-12:1998, 5.10.4).

EXAMPLE Character spaces are shown, separated by boxes to depict the exact length of the field as shown in Figure 6.

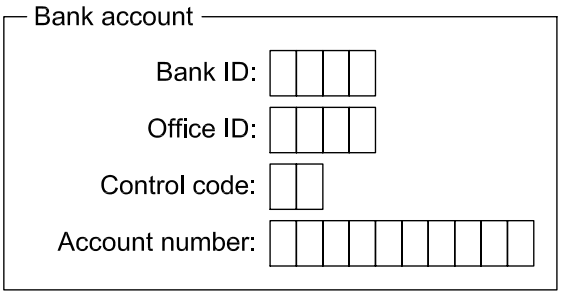

Figure 6 — Fixed length entry fields explicitly shown

NOTE Lengthy and/or complex groups of characters could be broken up into significant elements, possibly labelled accordingly.

5.4.3 Fields with maximum length

If the text entry fields have a maximum length, this should be explicit.

EXAMPLE 1 The maximum length is shown by the amount of space in the field.

EXAMPLE 2 A long field is accompanied by an indicator of maximum number of characters available.

5.4.4 Representation of optional and required entries

Required and optional entry fields shall be represented so that the differences between them are perceptible.

EXAMPLE 1 An asterisk is placed to the left of the label of required entry fields (making this information also available to blind users via speech output). In addition, a border with a heavier line thickness is placed around required entry fields to distinguish them from optional entry fields.

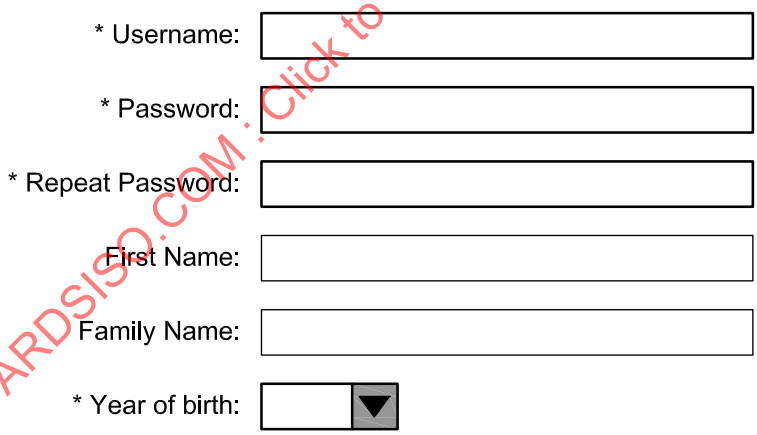

Figure 7 — Representation of required and optional entry fields

EXAMPLE 2 Underscores are used for required entries and periods for optional entries.

EXAMPLE 3 Different colours or shades are used to distinguish required and optional fields, with colours that are distinguishable on a monochrome display.

NOTE See ISO 9241-12 and ISO 9241-171 for guidance on limiting the exclusive use of colour.

5.4.5 Inform on representation used to distinguish between required and optional entries

Information on the representation used to distinguish between required and optional entry fields should be available to the user.

5.4.6 Modifiable versus non-modifiable fields

The user should be able to distinguish easily between fields that can be modified and those that cannot ("read-only" fields) by appropriate coding (see ISO 9241-12).

EXAMPLE 1 Non-modifiable fields have the same background colour as the background of the dialogue box, whereas modifiable (read-write) fields use a different colouring.

EXAMPLE 2 Read-only fields take focus in order for the contents to be read and so that they can be announced as read-only by screen readers.

5.4.7 Cues for entry format

Cues for data entry format (e.g. for time duration "hh:mm:ss") should be displayed within the entry field or in field labels. If abbreviated values are used, it should be clear to the user what the abbreviations mean (e.g. Y/N, for Yes or No).

NOTE See ISO 9241-12:1998, 5.10.3.

5.4.8 Cues for push buttons

Push buttons should have visual cues that indicate whether the command will be carried out immediately or the user will need to provide additional input or confirmation before the command is carried out.

EXAMPLE Push-button labels that will lead to another dialogue or message box before the command is carried out are followed by an ellipsis ("..."), while push-button labels for commands that will be carried out immediately do not have this cue.

5.4.9 Cues for expanding dialogues

If a form element is provided to allow the user to expand a dialogue box or form to reveal additional functionality, the element should contain a visual cue or label indicating that the dialogue box or form will be expanded.

EXAMPLE A push button with a symbol and a label (e.g. Advanced >>) is used to access the hidden portion of a dialogue.

5.4.10 Cues for contracting dialogues

If a form element is provided to allow the user to contract a dialogue box or form in order to hide functionality, the element should contain a visual cue or label indicating that the dialogue box or form will be contracted.

EXAMPLE A push button with a symbol and a label (<< Basic) is used to hide the optional portion of the dialogue.

It is important that the user be able to easily contract the dialogue again after expanding it.

6 Interaction

6.1 Navigation

6.1.1 Method

The user will need navigation methods to access the areas of the form that are required by their tasks. The choice of a navigation method should take into account the particular user population and compatibility with the user's flow of work.

6.1.2 Movements among fields

The user shall be provided with the capability to move from field to field within a group and, if appropriate, to move to nonadjacent fields in other groups.

NOTE 1 A tab key, cursor keys and/or pointing device can be used to move between fields.

NOTE 2 ISO 9241-171:2008, 9.3.2, and ISO 9241-171:2008, 9.3.14, give requirements for keyboard-only navigation and activation.

6.1.3 Quick access

If rapid access to a specific field in the form is required, a quick access mechanism should be provided.

EXAMPLE 1 Fields have implicit designators and the users are able to go to the field by typing in the designator.

EXAMPLE 2 The user presses a designated function key and then types in the label or number in a dialogue box.

6.1.4 Return to initial field

If appropriate to the task, a key or command should be provided so that the user can return to the initial field on the form.

6.1.5 Record cycling

If the data is organized in sequential records and a form represents a view of data from one record, a mechanism should be provided for cycling from record to record, forwards and backwards.

6.1.6 Pointing device and multiple forms

If a pointing device is used for input and the task involves multiple forms, a mechanism for navigating between forms using the pointing device should be provided.

6.1.7 Conditional navigation

If the value of a particular entry makes some of the entries following it unnecessary, the system should move to the next appropriate entry field after the entry is completed or upon tabbing.

Where possible, entry fields that are no longer required should be deleted or protected.

It is important for the user to recognise which fields currently do not require entries based on a previous entry.

6.2 Navigation by tab keys and scrolling

6.2.1 Tab keys or equivalents

A tab key, or equivalent, should be provided for moving from field to field.

NOTE Not every computing device has a tab key for moving from field to field, but in many cases there is another key or command that fulfils the same function. The term “tab key” is used to represent either a real tab key or as the equivalent of a tab key.

6.2.2 Completely filled-in fixed-length fields

If all of the fields on the form are to be completely filled in and the forms are easily learned, auto-skip tabbing from field to field should be provided (i.e. the cursor automatically skips to the next field when the last character position in the field is filled).

Auto-skipping should be used only for data entry tasks when it is consistent with user expectations, when it does not repeatedly create errors or delays, and when reverse tabbing is provided for correcting errors.

6.2.3 Mutually exclusive fields

If mutually exclusive fields are present on the form, skipping remaining choices should be allowed when an entry has been made for one of the choices in the field.

6.2.4 Form sections

If the form is organized into meaningful information groups (sections), the user should be provided with the capability to move from group to group, i.e. the user should not have to tab through all of the fields in a group in order to move to the next group.

6.2.5 Backwards tabbing

If a form or dialogue box supports tab key navigation, a mechanism for tabbing backwards through fields should be provided.

6.2.6 Forward wrapping

When pressed in the last field, the forward tab navigation key should move to the first field in the form or dialogue box.

6.2.7 Backward wrapping

When pressed in the first field, the backward tab navigation key should move to the last field in the form or dialogue box.

6.2.8 Field scrolling

If the maximum length of the data to be presented in a displayed field is longer than the field, a scrolling mechanism should be provided.

Consider providing auto wrap capability for text entry only.

6.2.9 Scrolling by pages

When scrolling by page, there should be a minimum of one unit of overlap between the information that is presented.

EXAMPLE Pages on the form are scrolled by using the up or down double arrows in the form's vertical scroll bar.

NOTE The overlap between one page and the next yields visual continuity for the user.

6.3 Input focus and cursors

6.3.1 Keyboard focus

Only one field at a time within the form or dialogue box shall have keyboard focus for a given user.

NOTE A field will have keyboard focus only when the window, form or dialogue box containing it has keyboard focus.

6.3.2 Cues for keyboard focus

6.3.2.1 The field with keyboard focus should be indicated by the presence of a focus indicator.

6.3.2.2 When a text field no longer has keyboard focus, it should not display a text cursor.

6.3.3 Initial focus position

When the dialogue box or form is first displayed, it should have keyboard focus and the focus indicator should be positioned automatically at the first entry field that must or may be completed by the user.

6.3.4 Initial placement of the text cursor

The location of the text cursor when a text field is first given keyboard focus should support the user's task.

a) If the task flow indicates that a portion of the text field is likely to be modified, then the text cursor should be placed at the end of that portion.

EXAMPLE A server and its fully distinguished name are displayed in a text field. The text cursor is placed at the end of the hostname, before the domain name, as this is the portion of the name most likely to be changed.

b) If the task flow indicates that no specific portion of the text is likely to be modified, the text cursor should be placed at the end of the text in the field.

NOTE If the entire text is selected for replacement it is possible that the text cursor will not be shown, see 6.3.5.

6.3.5 Replacement of text in a field

If the user's task will typically require the replacement of all rather than a portion of the text in a field, the entire text should be selected for replacement when the user first enters that field:

a) if the user enters information into the field, the previously selected text should be replaced with the new text;

b) if the user leaves the field without entering new information, the text that was initially in the field should remain in the field;

c) a mechanism should be provided for positioning the text cursor within the field so that information can be entered without deletion of the previous text, in case the user wants to modify the existing text.

EXAMPLE The user can remove the replacement highlight and position the text cursor by using the arrow keys or clicking within the text field with the pointing device.

6.3.6 Indirect regaining of keyboard focus

If a field has lost keyboard focus and subsequently regains the focus indirectly due to the form or dialogue box regaining focus:

a) if the field is a text field, the text cursor should reappear at the same position it had when the field lost keyboard focus;

b) if the field is not a text field (e.g. a radio button group), the selection indicator(s) and/or selection cursor should reappear at the same position as when the field lost keyboard focus.

6.3.7 Regaining focus by clicking in a field

If a field has lost keyboard focus and subsequently regains the initial focus by a click within the element:

a) if the element is a text field, the focus indicator should move to the field and the text cursor should move to the position in the field where the click occurred;

b) if the element is a check box or a radio button, the focus indicator should move to the specific button clicked and the state of that button should change;

c) if the element is a push button, that push button should be activated, and keyboard focus should change based on the action defined for the push button;

d) if the element is a list box, the focus indicator should move to the list box and the selection indicator should be shown on the item at the position where the click occurred.

EXAMPLE 1 Activation of a push button that results in a new dialogue box moves keyboard focus to the new dialogue box.

EXAMPLE 2 Activation of a push button that is part of a combination box causes a list to be displayed and keyboard focus moves to the first focusable item in the list.

6.3.8 Indicators and cursor for multiple selection

When a multiple-selection list box is entered, the selection cursor should be on the first item in the list box. There should be a selection indicator for each item in the list box that is currently selected.

EXAMPLE 1 In Figure 8, the selection cursor is on “Family”, while “Friends” and “Travels” are shown as currently selected items by the checks in the check boxes.

Figure 8 — Selection cursor and selection indication for check boxes

EXAMPLE 2 In Figure 9, the selection cursor is on “Family”, while “Friends” and “Travels” are shown as currently selected items by highlighting.

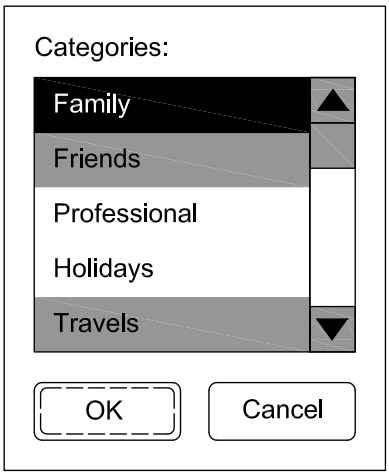

Figure 9 — Selection cursor and selection indication for a list

6.3.9 Indicator for single selection

When a single selection list box initially receives focus, the presence and location of the selection indicator should support the user's task:

a) the selection indicator should be on the current selection or the default selection in the list box — see the recommendation given in 8.9 for selections in a single-selection list box;

b) if there is no current selection and a default selection is inappropriate to the task, no selection indicator should be present; nevertheless, the selection cursor should be on the first item in the list to allow keyboard navigation.

EXAMPLE Figure 10 contains a selection indicator with a default selection. Figure 11 contains a selection cursor but does not have a currently active selection.

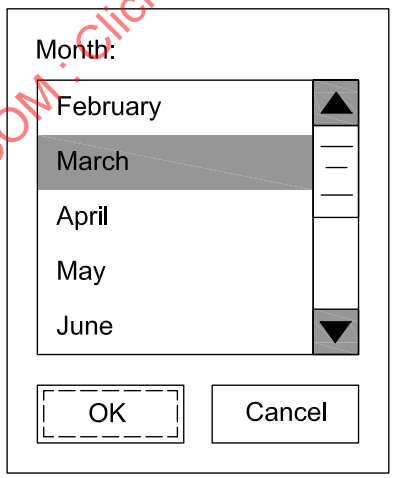

Figure 10 — Default selection

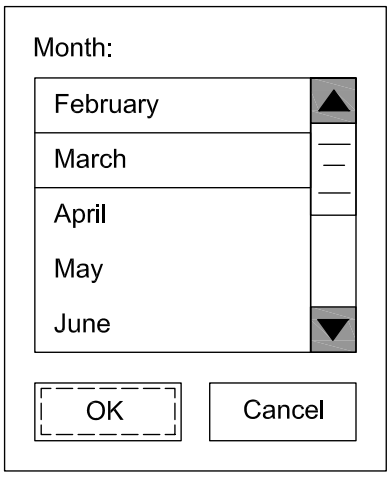

Figure 11 — Selection indicator without a default selection

6.4 Input

6.4.1 Considerations

User input considerations should include

— user control of the dialogue at all times,

— the capability for the user to recover easily from errors, and

— avoiding requiring the user to input more information than is necessary for successful task performance or input information that is currently available in the system.

6.4.2 Minimize cursor movement

The user actions required to move the cursor from one entry field to the next should be minimized.

EXAMPLE A tab key is used to jump from field to field.

6.4.3 Input device independency

If multiple input device access is available to the application, all form elements should be operable by all available input devices (i.e. device independent).

EXAMPLE A user can complete a form field using the keyboard, a mouse or voice input.

6.4.4 Pointing devices

If a pointing device can be used for input in a form it should be usable for navigation as well.

6.4.5 Switching between input devices

If appropriate to the task, the need for the user to switch between different input devices when filling in a form should be minimized.

EXAMPLE 1 Fields that require text entry within a logical group on a form are grouped separately from those that can be completed with a pointing device.

EXAMPLE 2 Multiple data entry methods are available for use in the same field where appropriate.

EXAMPLE 3 Navigation to all form fields with all input devices is provided for a given form.

6.4.6 Incomplete text entry field

If the (required) number of characters entered into the field does not fill the whole field, the user should be allowed to move directly to the next field (without, for example, requiring the user to enter blank spaces to complete the field).

6.5 User control

6.5.1 Changes or corrections

If user changes in a form can impact a database accessed by multiple users and/or this results in negative consequences, the user should be allowed to go back to the initial state of the form or dialogue at any time and start over again, cancel entries or change any entry before the form is processed by the computer.

6.5.2 Immediate processing of user input

6.5.2.1 If a change in a form results in only visual changes on the screen and local storage of the data, and if it can be reversed without negative consequences, user inputs should be processed immediately and then transferred to the database on user confirmation.

EXAMPLE The choice of an option in the first field of a form changes the values visible in the second field, the user can go back and change the selection in the first field without affecting the data in the database.

6.5.2.2 If changes in a dialogue box can be reversed without negative consequences and user performance will benefit, user inputs should be processed immediately without a confirming action by the user. See also 6.5.8.

EXAMPLE When making desktop settings such as background colour and font, the changes are immediately reflected in the desktop as the user makes the selections. No push button confirmation is required.

6.5.2.3 If user input is processed immediately without user confirmation, as recommended by 6.5.2.1 or 6.5.2.2, the user should be allowed to go back to the initial state of the form or dialogue at any time and start over again, cancel entries or change any entry before the form or dialogue box is closed.

6.5.3 Identifying and locating errors

6.5.3.1 Multiple fields

If validation checking of multiple fields detects fields in error and if appropriate to the task, these fields should be indicated and the cursor should be placed on the first field in error and the user should be allowed to easily move through the fields in error to correct the entries.

EXAMPLE 1 All fields in error are highlighted by the reverse video display of these fields.

EXAMPLE 2 Fields for correction are indicated by an asterisk as well as a visual highlight.

6.5.3.2 Dependencies

If there are dependencies between fields, and if it is appropriate to the task, potential errors resulting from such dependencies should be indicated.

6.5.4 Re-entering data

If the field contains an error, the user should be required only to correct the erroneous part of the input.

NOTE In the case of data involving security or authorization information, it could be necessary for an entire field, or fields, to be re-entered.

6.5.5 Disabled areas

The user should not be able to enter information in areas of the screen display not available for user input (e.g. read-only fields), and these areas should give visual cues and information to screen readers indicating that they are disabled.

NOTE 1 A cursor might be placed in a disabled field for the purposes of selection to copy and paste content to other fields.

NOTE 2 If users want to get context-sensitive help on read-only fields, it could be necessary to allow the focus indicator to move to read-only fields, so that keyboard mechanisms for activating the Help can be used (e.g. F1).

6.5.6 Easy transmission

If transmission of the form is required, transmission of the field entries to be processed should be accomplished by means of a simple, explicit action.

The transmission action should take place no matter where the cursor is currently located on the form, i.e. the user is not required to navigate to a particular field in order to use the transmission control key(s).

NOTE If the form is to be completed by a blind user, it is useful to provide a submit button that the user can select at any time, after having had the opportunity to review all the form fields.

6.5.7 User control information

Unless it is obvious to the user, the form should state how to carry out the following actions, if provided:

a) signal completion of the form or dialogue box and redisplay an empty form (with default values, if appropriate) for the entry of new data;

b) signal completion of the form and redisplay the previously completed version of the form or dialogue box or a default version (template) of the form or dialogue box;

c) dismiss the form or dialogue box without changing any data in the system, e.g. by means of the “Escape” or “Cancel” function;

d) use “Undo”.

6.5.8 Dismissing dialogue boxes

6.5.8.1 Single actions

if the expected use of the dialogue box involves the single application of a set of inputs, the dialogue box should contain a button that applies the input and dismisses it.

6.5.8.2 Multiple actions

If the expected use of the dialogue box or form involves multiple applications of sets of user input, the dialogue box should have a form element that applies the input without closing the dialogue box and a form element that closes the dialogue.

EXAMPLE In a dialogue box for typing in names and addresses, the user has a push button labelled “Add” for applying the input and clearing the fields for the next entry. There is also a push button labelled “Close” for closing the dialogue box.

6.5.8.3 Single and multiple actions

If the expected use of a dialogue box or form may involve both single application of input and repeated applications of input, the dialogue should provide a mechanism for applying the input without closing the dialogue, as well as a button for applying the input and closing the dialogue.

EXAMPLE In a dialogue box that allows the user to choose a colour scheme, a push button labelled “Apply” changes the colour scheme without closing the dialogue and “OK” changes the colour scheme and closes the dialogue.

6.5.9 Temporary save

If appropriate to the task and if system constraints allow, a temporary save function should be provided so that the user can leave the form temporarily (because the required item of data is not available, for example) and return to it later without having to re-enter all of the data on the form.

NOTE If a temporary save function is provided, when the form is re-selected and there is already a put-aside form of that type (from a previous save), the user can be given the option of continuing with the saved form or starting a new form.

6.6 Feedback

6.6.1 Information needed

The user should be provided with information allowing him or her to control the dialogue, recognize errors and determine their next course of action.

6.6.2 Typed-in character echoing

Typed-in characters shall be echoed back to the user, character by character, as entered.

NOTE 1 In some cases, echoing can be undesirable (e.g. during the entry of a password, non-identifiable characters such as “•••••••” echoed back to the user instead of the actual characters typed).

NOTE 2 Echoing is effective when it is not delayed by a perceptible lag time.

6.6.3 Cursor position and visibility

6.6.3.1 The cursor position shall always be clearly visible if within the currently displayed portion of the form.

6.6.3.2 Software shall provide at least one mode where keyboard focus cursors and text cursors can be visually located by people with unimpaired vision at a distance of 2,5 m when software is displayed on a 38 cm (15 inch) diagonal screen at 1024 × 768 pixel resolution, without moving the cursor (ISO 9241-171:2008, 9.2.2).

6.6.4 Pointer position

If a pointing device is available, the position of the pointer shall always be clearly visible to the user (see ISO 9241-12:1998, 6.2, and ISO 9241-171:2008, 9.4.13).

NOTE A finger in contact with a touch screen can be considered a pointing device. In some cases, the location of the finger indicates the position.

6.6.5 Focus indicator

An easily discriminable focus indicator shall be provided that allows the user to determine which field or form element in the dialogue box or form currently has keyboard focus (see also ISO 9241-171:2008, 9.2.2).

6.6.6 Field errors

If a field contains an error and, if appropriate to the task and within the system's capabilities, error feedback should be provided as soon as the user completes the field (e.g. by highlighting the error or by providing information on the nature of the error and correct entries) in a manner that is minimally disruptive of the continuation of the task (see also ISO 9241-13).

NOTE Security or safety requirements can necessitate immediate correction.

6.6.7 Transmission acknowledgment

If appropriate to the task, the system should provide an acknowledgment to the user that the transmission of the form entries has been accepted by the system.

6.6.8 Database changes

If the form or dialogue box changes a database, feedback that the database has been updated should be provided to the user.

6.7 Access to forms and dialogue boxes

6.7.1 Access mechanism

If an application contains various forms and dialogue boxes, the user shall be provided with a mechanism to access a particular form or dialogue box.

EXAMPLE A tree-structured list is provided that depicts the various forms and dialogue boxes available in the application from which the user can select.

6.7.2 Direct form access

If appropriate to the task and if forms can be accessed independently, the user should be able to select forms directly, e.g. by naming the form or by selecting it from a menu, or by selecting a form “container object” through direct manipulation.

6.7.3 Movement between forms

If forms can be accessed independently and if appropriate to the task, the user should be able to move from form to form, forwards and backwards, in a predefined sequence and without losing input.

6.7.4 Hierarchical level movement

If the set of forms is hierarchical, the user should be provided with the capability to move to both the next higher and lower levels in the structure.

6.7.5 Returning to the initial form

If the set of forms is hierarchical, the user should be provided with an obvious means of returning to the initial form (i.e. the form at the top of the hierarchy) from any form in the hierarchy.

NOTE It is assumed that the form has not yet been submitted.

6.7.6 Forms in a windows environment

If more than one form can be displayed in a window environment:

a) only the last selected form should be active and ready for input;

b) if appropriate to the task, the user should be provided with the capability to switch to another form to make it active.

6.7.7 Default form

If one form is more likely to be used than another, generally or for a particular task, user, environment and/or technology configuration, that form should be the initial form (i.e. the system should display the form on the screen automatically when the system or forms application is initially activated).

6.7.8 Property dialogues

If a dialogue box is used to present properties of an object or item, the following is recommended.

a) **Access**

The user should have a simple and consistent mechanism for accessing the property boxes.

EXAMPLE 1 The Edit menu contains an item labelled “Properties”. To access a properties box, the user selects the object and then accesses the Edit -> Properties menu item.

b) **Association with object**

The property box should provide a means for the user to identify the object or item with which the properties are associated.

EXAMPLE 2 The title of the properties box provides the name of the object whose properties are being shown.

c) **Multiple properties**

If an object has a large number of properties associated with it, the properties should be categorized and grouped for presentation to the user.

EXAMPLE 3 A document has a number of properties that are grouped under tabs with the following labels: Title, Page Layout, and Permissions.

d) **Templates**

If possible, the user should be able to take the property settings for a given object and apply them to a new or existing object.

EXAMPLE 4 The user can take the properties for an existing paragraph and apply them to a new paragraph so that the new paragraph uses the same font, point size, margins, spacing and text colour.

e) **Modifications**

If the user can modify the properties of objects, the properties box should provide a mechanism for making these modifications.

6.8 Default values

6.8.1 Field default values

Fields should contain default values whenever possible and appropriate for the task.

6.8.2 Choice of system default values

Whenever default values are used, the value should be chosen to support the user's task:

a) system default values should not be destructive (i.e. lead to the loss of data), nor should they lead to undesirably time-consuming activities;

b) in a single-selection list box, the initial default selection should be either the item most likely to be selected or the first item in the list;

c) in a multiple-selection list box, the default values should include a selection of the set of items most likely to be selected by the user;

d) in a group of radio buttons, there shall be a default value initially selected;

e) in a group of radio buttons, the default value should be the choice most likely to be selected;

f) in a group of check boxes, each check box should be set to the value that will most likely be selected, unless it is inappropriate to the task (e.g. the user is required to make an explicit choice);

g) the default value in a text box should be the value most likely to be entered by the user;

h) if stepper buttons are used, the initially displayed choice in a stepper button should be the most logical default choice (i.e. the most likely to be selected).

NOTE "Not answered" or "Not applicable" might be provided as default values [d)].

EXAMPLE If the user opens a document in a directory and the user has not previously accessed the save dialogue box, that directory in which the document was opened can be the default directory for saving the document.

6.8.3 User configured defaults

If default values are likely to vary across users but remain relatively consistent across the tasks of a specific user, methods should be provided to allow the user to customize default values.

6.8.4 Return to system defaults

If an application allows the user to customize default values, some means shall be provided to allow the user to return the settings to the system default values.

6.8.5 Defaults in previously opened dialogue boxes and forms

Dialogue boxes or forms that may be accessed multiple times should provide default values that support the user's task.

a) If likely that retaining values set by the user will minimize user steps or the need for changes in field values, user settings should be presented as the default values the next time the dialogue box or form is accessed.

b) If unlikely that the user will want to retain previously set values, user-customized defaults should be presented as the default values the next time the dialogue box or form is accessed.

NOTE 1 Sometimes user-customized defaults are set on an organizational basis rather than on an individual user basis.

c) If unlikely that the user will want to retain previously set values and user-configured defaults have not been set, system defaults should be presented as the default values the next time the dialogue box or form is accessed.

NOTE 2 The default values that are used for fields in the dialogue box or form can be a combination of previous user values, user-configured values and system default values.

EXAMPLE In the print dialogue box, the default printer is the printer last selected by the user. However, the number of copies returns to one copy based on the system defaults.

6.8.6 Editable default values for text fields

Text default field values should be editable by the user using conventional editing commands.

6.8.7 Default values for a group of radio buttons

If there is a default for a group of radio buttons, the default choice in the set shall be visibly selected when the field is first presented.

6.8.8 Default values for a group of check boxes

If there are default values for a group of check boxes, the default choice for each check box in the group shall be indicated as active or inactive when the field is first presented.

6.9 Default actions for forms elements

6.9.1 Default actions

When default actions will benefit task performance, default actions should be defined for the elements in the dialogue box or form.

EXAMPLE The default action for the Username field in the dialogue box shown in Figure 12, is to move keyboard focus and text cursor to the Password field.

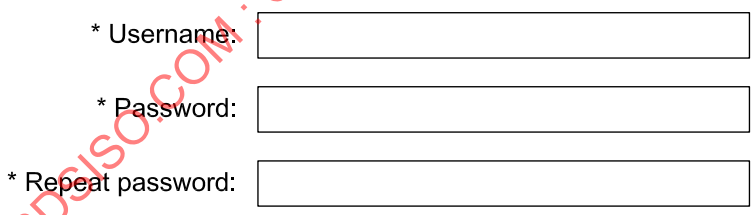

Figure 12 — Login dialogue box

6.9.2 Number of default actions

At any point in time only one default action shall be applicable.

NOTE The default action can change over time within the same dialogue box or form, based on the position of the focus indicator.

6.9.3 Activation of default action

A consistent user action should initiate the default action.

EXAMPLE Pressing the Enter key, Return key or clicking with the mouse button when the pointer is on the element initiates the default action.

6.9.4 Safe default actions

Actions chosen as default actions should be safe (i.e. not destructive or resulting in a time-consuming activity that the user cannot easily cancel).

6.9.5 Cues for default action

If default actions are used and they are associated with push buttons, the current default push button should have a visual cue to indicate that it is the default.

EXAMPLE The current default push button in Figure 13 is indicated by an thicker outer border.

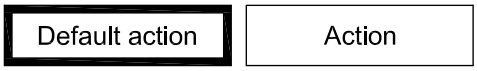

Figure 13 —Default action shown by different border types

6.9.6 Default actions in multi-field dialogues

If the dialogue box or form has more than one input field, the default action should be based on the most likely user action given the location of the focus indicator and the structure of the dialogue:

a) if the focus indicator is on a push button, the default action should be to activate the push button;

b) if the focus indicator is in a single line text field and there is a push button associated with the text field (e.g. Add, Find), the default action should activate the push button and the location cursor should remain in the text field;

c) if the focus indicator is in a single line text field and the field does not have an associated push button, the default action should be to move to the next input field if there is one;

d) if the focus indicator is in a multiple line text field and [Return] is used as the default action mechanism, the default action should be to move the text cursor to the next line in the text field and for the location cursor to remain in the text field;

e) if the focus indicator is in a radio button, the default action should be to select the radio button and move to the next field after the group of radio buttons;

f) if the focus indicator is in a check box, the default action should be to select the check box and to move to the next check box in the group or to the next field if there are no more check boxes in the group;

g) if the focus indicator is in a single or multiple-selection list box, the default action should be to perform the most typical action associated with the list item (e.g. selection or open);

EXAMPLE 1 In a form where the default action for a single-selection list item is selection, the item is selected and the focus indicator is moved to the next field.

EXAMPLE 2 In a form where the default action for a list item was to open a dialogue box, the dialogue box is opened.

h) if the focus indicator is on the last field in the dialogue before the exiting push buttons, and the user will not need to review the values entered in the dialogue, the default action should be to follow recommendations b) to g) above and then to activate the push button to close the dialogue or move to the next page (e.g. tabbed dialogue).

EXAMPLE 3 In a dialogue box for user login to a system, the default action in the text field for login name is to take the entry and move to the password field. When the password field contains the focus indicator, the OK button at the bottom of the dialogue is highlighted as the default action. When the user presses [Return] at the end of the password entry, the password entry is accepted and the dialogue box closes.

6.9.7 Default actions in simple dialogues

In dialogue boxes and forms where the primary user interaction is provided by specific push buttons (e.g. OK and Cancel), one of the push buttons should be defined as the default action for the dialogue when the user first enters the dialogue.

EXAMPLE A user can tell that the OK button is the default action in the button grouping shown in Figure 14.

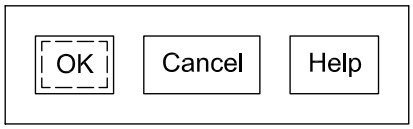

Figure 14 — Default action indicated in dialogue box

7 Validation

7.1 Single-field validation

If system capabilities are available, the data entry system should check the entry in each field before accepting it, based on criteria defined for that field individually. If values in a field are defined from a predefined range or list, the editing criteria should simply verify that the entry matches some item in the predefined range or list.

NOTE Validation can involve both syntax (e.g. date format) and even more importantly semantics (e.g. correct date values).

7.2 Multiple-field validation

If there are dependencies between fields on a form or dialogue box, or between the field on other instances of the same form, the following additional validation checks may be made.

a) Data already entered in other fields of the same form.

EXAMPLE 1 The user is not allowed to enter data in the field “Age of Child” if the entry in the field “Children” is “0”.

b) Data already entered in the same field in other forms.

EXAMPLE 2 A field is a “key field” and is required to be unique. The system checks to make sure this value has not been used before on another instance of the form.

8 Choice of form elements

8.1 Accessibility of form elements

All form elements shall be accessible to screen readers.

8.2 Choice considerations

The more conditions under each of the recommendations given in 8.3 to 8.16 that are met, the more appropriate is the use of the form element. The conditions are not listed in any order of precedence.

NOTE The number of conditions/rules is not a complete/exhaustive set. More choice considerations can be relevant in other dialogue types.

8.3 Push buttons

Push buttons should be used, as appropriate, when

— the desired result is the making of a selection or the setting of a state,

— the desired result is the execution of an action,

— there is a need for quick access,

— there is a need for persistent high recognisability.

NOTE In some cases, a selectable area of a form (e.g. a windows border) that can result in an action acts similarly to a push button.

EXAMPLE In the example in Figure 15, OK is the default push button.

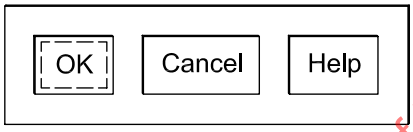

Figure 15 — Dialogue box push buttons

8.4 Toggle buttons

Toggle buttons should be used, as appropriate, when

— the desired result is setting a state,

— the set of choices is binary and can be meaningfully portrayed,

— the user needs to quickly see the current state of the options,

— the user will clearly understand the meaning of the choice when it is selected or not selected,

— the user may need to change settings frequently.

NOTE 1 Toggle buttons can be arranged in groups.

NOTE 2 Toggle buttons are an equivalent of radio buttons.

EXAMPLE As shown in Figure 16, a user has selected “ft” as the measurement unit using the “ft” toggle button.

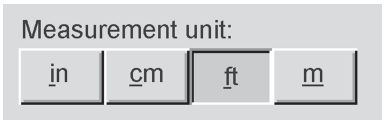

Figure 16 — Measurement unit toggle buttons

8.5 Text entry fields

Text entry fields should be used, as appropriate, when

— the desired result is the making of a selection, the setting of a state or assigning a value,

— the set of possible valid entries is large,

— not all of the entries can be predefined in advance,

— the user can easily enter valid entries,

— the element is used as part of a keyboard intensive task, or

— any input is acceptable (e.g. a comment field).

EXAMPLE A user is able to enter their name in the entry field shown in Figure 17.

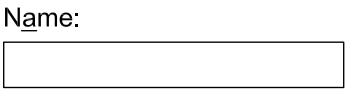

Figure 17 — Text entry field

8.6 Radio buttons

Radio buttons should be used, as appropriate, when

— the desired result is the making of a selection or the setting of a state,

— there are two or more mutually exclusive choices,

— display space is sufficient for all buttons and labels,

— the number of choices is not large (e.g. 5 or fewer),

— the user needs to quickly see which option is currently selected,

— the user will benefit from seeing all potential choices simultaneously,

— the user may need to change settings frequently.

NOTE 1 The logic for radio buttons can be viewed as choosing one of A, B, C, etc.

NOTE 2 Radio buttons can be arranged in labelled groups.

EXAMPLE A user has selected the 50 % zoom level radio button as shown in Figure 18.

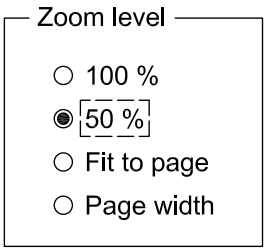

Figure 18 — Radio buttons for selecting zoom level

8.7 Check boxes

Check boxes should be used, as appropriate, when

— the desired result is the making of a selection or the setting of a state,

— each setting is a single two state choice,

— choices can be meaningfully portrayed as an on–off, or yes–no or true–false state,

— the user needs to quickly see the current state of the options,

— the user will clearly understand the meaning of the choice when it is selected or not selected,

— the user will benefit from seeing all potential choices simultaneously.

NOTE 1 The logic for check boxes can be viewed as choosing between A and not A.

EXAMPLE A user has selected Tuesday and Wednesday in the “Day of week” check boxes shown in Figure 19.

Figure 19 — Check boxes

NOTE 2 Check boxes make it apparent that multiple choices are permitted, see example in Figure 19.

NOTE 3 Check boxes can be arranged in labelled groups. Check boxes in a group are mutually independent.

8.8 Stepper buttons

Stepper buttons should be used, as appropriate, when

— choices are mutually exclusive,

— there is limited space,

— there is a familiar sequential order to the items,

— the user does not need to preview the options before making a selection,

— the user will want to make small changes relative to the current value, quickly,

— the user may want to go in either direction within the sequence.

EXAMPLE A user selects the appropriate number by activating the up or down arrows as shown in Figure 20.

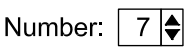

Figure 20 — Stepper buttons

8.9 Single-selection list boxes

Single-selection lists should be used, as appropriate, when

— the desired result is the making of a selection or the setting of a state,

— choices are mutually exclusive,

— there is adequate space to display three or more items simultaneously without scrolling,

— there are more than five items or the number of items may change over time,

— the user may need to change settings frequently,

— there is value in having a large number of choices simultaneously visible (with the number of choices depending on the context of the form layout),

— the number of items changes dynamically.

EXAMPLE In Figure 21, the month of March is selected by the user in the list box.

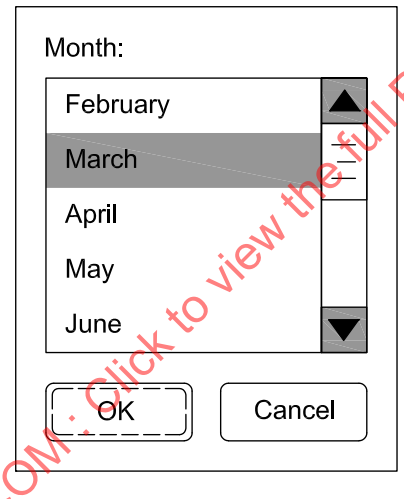

Figure 21 — Single-selection list box

NOTE 1 If a scroll bar is to be used with the list box, it is important that the list box is large enough to show at least 3 items.

NOTE 2 List boxes are harder to use by elderly users than groups of radio buttons or check boxes.

8.10 Multiple-selection list boxes

Multiple-selection lists should be used, as appropriate, when

— the desired result is the making of a selection or the setting of one or more states,

— choices are *not* mutually exclusive,

— there is adequate space to display three or more items simultaneously without scrolling,

— there are more than five items or the number of items may change over time,

— the user may need to change settings frequently,

— there is value in having a large number of the choices simultaneously visible,

— the user will clearly understand the meaning of the choice when it is selected or not selected,

— the number of items changes dynamically.

EXAMPLE 1 In the check box list shown in Figure 22, a user can select the items he or she is interested in (selected items are checked).

Figure 22 — Multiple-selection list box using check boxes

EXAMPLE 2 In the list box shown in Figure 23, a user can select the items he or she is interested in (selected items are highlighted).

Figure 23 — Multiple-selection list box using highlighting

NOTE When the users need to make multiple selections, it is especially important to make the box as large as space will allow.

8.11 Pop-up/drop-down list

Pop-up or drop-down lists should be used, as appropriate, when

— the desired result is the making of a selection or the setting of a state,

— choices are mutually exclusive,

— there is very limited space,

— the settings associated with the element are not changed frequently,

— except when changing the selection, the user needs to see only the item which is currently selected,

— there are four or more items or the number of items may change over time,

— all values can be supplied by the application.

EXAMPLE 1 When the user activates the “Networks” button, a pop-up list is displayed as shown in Figure 24.

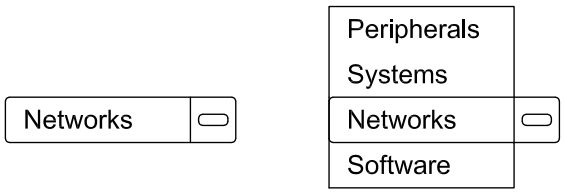

Figure 24 — List button and associated pop-up list box

NOTE A pop-up list allows the user to pick the depicted value without moving the selection cursor.

EXAMPLE 2 When the user activates the “down-arrow” in the “Networks” field, a drop-down list is displayed as depicted in Figure 25.

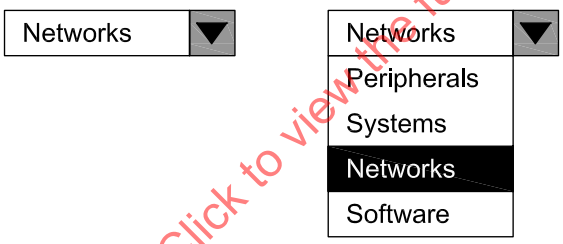

Figure 25 — Drop-down list entry and associated list box

8.12 Combination boxes

Combination boxes should be used, as appropriate, when

— the desired result is the making of a selection or the setting of a state or assigning of a value,

— choices are mutually exclusive,

— there is limited space,

— the user needs to see which option is currently selected,

— there are five or more items or the number of items may change over time,

— the user may be able to type the entry more quickly than they can select it,

— the element is used as part of a keyboard intensive task,

— the user may have to type values that cannot be supplied by the application.

EXAMPLE In the combination box in Figure 26, the user can type in the country name or select it from the list or type an item that is not in the list.

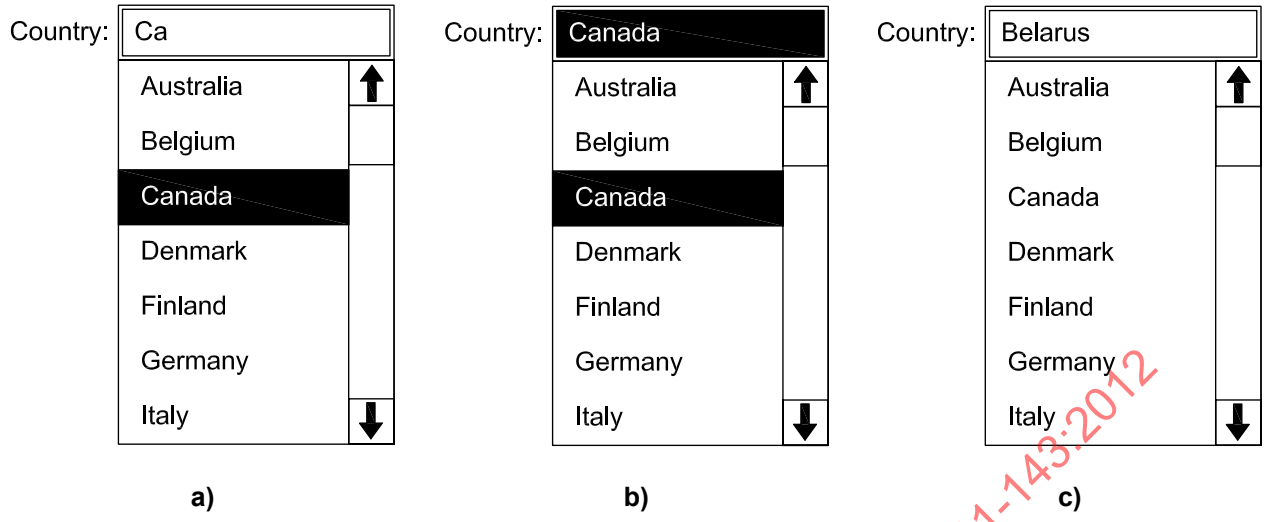

Figure 26 — Combination box

8.13 Single-selection hierarchical lists

Single-selection hierarchical lists should be used, as appropriate, when

— the desired result is the making of a selection,

— choices are mutually exclusive,

— there is adequate space to display three or more options simultaneously without scrolling,

— there are a large number of items,

— the options can be meaningfully grouped in a hierarchy,

— the user needs to implicitly select all of the subordinates of a level of the hierarchy.

EXAMPLE 1 In Figure 27, a hierarchical list of staff information is depicted from which only one selection can be made.

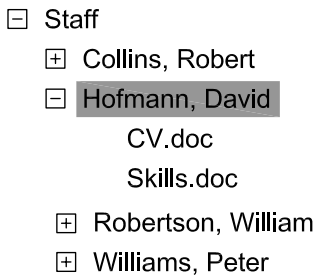

Figure 27 — Single-selection hierarchical list

NOTE 1 When an item is selected at the higher level category, all of the subordinate levels are automatically selected.

EXAMPLE 2 When selecting a department to receive copies of a form then all members of the department are automatically selected to receive the form.

NOTE 2 When the user needs to make selections from a hierarchical list, it is especially important to make the box as large as space will allow.

NOTE 3 If hierarchical selection and details of the selected item are to be displayed simultaneously, both areas can be scrolled independently.

NOTE 4 Hierarchical lists are often referred to as *tree structures*.

8.14 Multiple-selection hierarchical lists

Multiple-selection hierarchical lists should be used, as appropriate, when

— the desired result is the making of a selection,

— choices are not mutually exclusive,

— there are a large number of items,

— there is adequate space to display three or more options simultaneously without scrolling,

— the options can be meaningfully grouped in a hierarchy,

— the user needs to simultaneously select all of the subordinates of a level of the hierarchy.

EXAMPLE In Figure 28, a hierarchical list of staff information is depicted form which only multiple selections can be made.

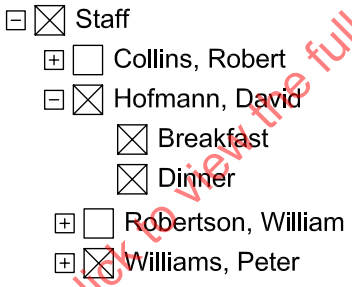

Figure 28 — Multiple-selection hierarchical list

NOTE 1 When the user needs to make selections from a hierarchical list, it is especially important to make the box as large as space will allow.

NOTE 2 If hierarchical selection and details of the selected item have to be displayed simultaneously, both areas might have to scroll independently.

8.15 Analogue form elements (slider, rotary dials and equivalents)

Analogue elements should be used, as appropriate, when

— the desired result is the making of a selection or the setting of a state,

— choices are mutually exclusive,

— display space is sufficient,

— there is a need to present and manipulate a variable along a continuum,

— there is a need for large imprecise changes with minimal effort,

— there is a need to show the current value relative to the possible range of values.

EXAMPLE In the “volume” slider shown in Figure 29, a user moves the slider to set the volume.

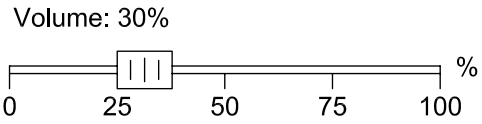

Figure 29 — Example of a numeric slider

8.16 Tabbed form elements

Tabbed elements should be used, as appropriate, when

— the desired result is the presentation of multiple pages of information in a limited display space,

— there is more information than can be presented in a single dialogue,

— there are multiple forms which need to be visible, one at a time, simultaneously with another user-interface element,

— there are multiple settings that can be grouped into meaningful non-overlapping categories;

— there is no sequential dependency between groups of elements (placed on separate tabs);

— the user does not need to see all settings simultaneously.

EXAMPLE 1 A user can step through a list of items, with item details displayed on tabs and, for each tab, can inspect the respective tab content.

EXAMPLE 2 A user can select from the “Type” or “Options” tab, as shown in Figure 30.

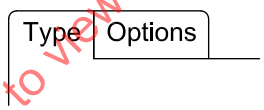

Figure 30 — Tabbed elements

9 Form element design

9.1 Alphanumeric text entry

9.1.1 Justification of entries

If the entries need to be justified within the field, the system (not the user) should do the justification.

9.1.2 Leading zeros

If leading zeros are needed for numeric entries, the system (not the user) should provide the leading zero entries (e.g. the system provides appropriate zeros in the entry field).

EXAMPLE Since the character string “0015” is not equivalent to “15”, the system places “00” in front of “15” when the user only enters “15” into the field.

9.1.3 Multiple lines

If a text field contains multiple lines of text (i.e. sentence or a paragraph), the following is recommended.

a) **Input area size**

The size of the multiline input area should be clearly indicated.

EXAMPLE Borders are placed around the text area, as shown in Figure 31.

Comments (max. 110 characters):

Data in this area is bounded by the border of the field. The text entry area cannot exceed the area.

Figure 31 — Bordered text area

b) **Auto wrap**

An auto-wrap capability should be provided and individual words should not be broken.

c) **Editing and navigating**

Navigating and editing text should be consistent within the application.

d) **Scrolling**

If the length of the text is longer than the text area, a scrolling mechanism should be provided.

9.1.4 Mutually exclusive fields

If fields are mutually exclusive, a perceptible cue shall be provided to indicate that only one of the fields can be used.

EXAMPLE Radio button in a dialogue box or the use of the word “or”.

9.1.5 Interdependency rules

The use of complex “if/then” interdependency rules among entry fields should be avoided, or else they should automatically be handled by the system through constraining user choices and visible fields.

EXAMPLE If “Y” is entered in field 1, then fields 2 and 4 must be left blank; if “N” is entered in field 1, then field 2 should contain ----, etc.

NOTE If interdependencies between fields are necessary, consider providing automatic movement through the required completion sequence by the system and visually coding fields available for input due to interdependencies.

9.1.6 Text entry field area

Each text entry field should be large enough to accommodate the majority of anticipated entries without scrolling.

9.1.7 Numeric entry field area

Each numeric entry field shall be large enough to accommodate the largest valid numeric entry without scrolling.

9.2 Choice

9.2.1 Choice categories

Choices are typically grouped into the following categories:

a) exclusive choice (a group of choices in which only one choice can be selected at a time);

b) binary choice (a choice between two settings, e.g. on/off, activated/deactivated);

c) non-exclusive choice (a group of choices in which more than one choice can be selected, i.e. the choices are independent of each other).

9.2.2 Limited choice options

If the number of acceptable entry options is limited and predetermined, a mechanism should be provided to enable the user to view and select available options.

EXAMPLE 1 Providing pop-up list boxes.

EXAMPLE 2 The use of soft machine controls which are graphical representations having an explicit or historical analogy to hardware controls or other entities (e.g. buttons, sliders, radio buttons, check boxes).

9.2.3 Discriminable visual cues

If the form is rendered visually, discriminable visual cues shall be used to discriminate among different logical types of visually presented choice entries in an application.

EXAMPLE 1 Circles are used to indicate exclusive choices (e.g. radio buttons) and square boxes (e.g. check boxes) are used to indicate non-exclusive choices.

EXAMPLE 2 Diamonds are used to indicate exclusive choices and square boxes are used to indicate non-exclusive choices.

9.2.4 Push buttons (Screen buttons)

If the user must select a small number of values (2 to 5) and the values become effective immediately after selection, the use of push buttons should be considered.

9.2.5 Groups of radio or toggle buttons

Exclusive choice buttons (e.g. radio buttons, toggle buttons) should be presented in sets of two or more choices and the group labelled.

NOTE A descriptive group label is particularly import for screen reader users.

9.2.6 Multiple binary state settings

When using check boxes the following is recommended.

a) **Group presentation**

Binary state buttons (e.g. toggle buttons) related to the current task should be presented in one or more context-related groups rather than as individual items.

b) **Indication of state**

When the form is presented, binary state buttons should provide a visual indication of their current state.

9.2.7 Layout of radio buttons and check boxes

If display space is available, all choice buttons (i.e. radio buttons or check boxes) within a group should be vertically aligned.

NOTE Vertically aligned choice buttons are easier for the user to scan.

9.2.8 Use of check boxes for multiple objects

If a check box represents a setting shared by multiple objects, the check box setting should provide visual cues to indicate a summary of the settings across all objects.

EXAMPLE When all of the objects in the group have the check box setting on, the setting for the group is shown as checked. When all of the objects in the group have the check box setting off, the setting for the group is shown as an empty box. When some objects have the setting turned on and others have the setting turned off, the setting for the group is shown as greyed.

9.3 List-based elements for choice

9.3.1 Types of list-based elements

List-based elements include menu buttons, list boxes, combination boxes, stepper buttons, and drop-down lists.

9.3.2 Fields with text and list-based elements

If the user must select a field entry option from a known set of options, a list-based element should be considered as the entry technique. When a list-based element is used, the following is recommended.

a) **Visual cues**

A visual cue that a list element is associated with the field should be provided, unless the selection list associated with the list element is continuously visible.

EXAMPLE 1 An arrow in the field label or in the button surrounding the data value indicates a list is available, as shown in Figure 32.

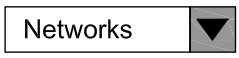

Figure 32 — Field label with arrow

EXAMPLE 2 A bar on the button indicates a list button and not a typical push button, see Figure 33.

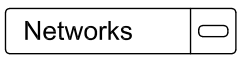

Figure 33 — Button with bar

b) **Field value**

The form field should present the most recent or the default selection from the list as the current value for the field.

NOTE It is useful to provide a means to differentiate defaults from user-selected values.

9.3.3 List elements that include a text field

If the user will be familiar with the values and typing would improve performance, the element should allow the user to set the value for the field by typing into the text field of the element.

9.3.4 Non-scrolling or scrolling lists

If the user must select one or more entries from a large and/or variable set of values, or if the user can customize a list, the use of non-scrolling or scrolling lists should be considered. When lists are used, the following is recommended.

a) **Visual cues**

A visual cue should be provided that allows the user to discriminate selected from non-selected values.

EXAMPLE 1 Check marks or highlighting of selected values.

b) **Long lists**

If a list is extremely long, it should be presented in the appropriate logical order (e.g. alphabetical order, numeric, date order) and a mechanism should be provided so that the user can rapidly navigate through the list.

EXAMPLE 2 The user is allowed to skip to parts of a list by typing in the initial letter(s) of list items.

9.3.5 Hierarchical list elements

If hierarchical list elements are used, the following is recommended.

a) Separate mechanisms should be provided for the user to expand items and to select items.

EXAMPLE 1 A hierarchical element is used in a file browser. If the user clicks on the button with a plus sign, a list of sub-items is presented. If the user clicks on the name of the item, the item becomes selected.

b) Items at the same level in the hierarchy should be consistently sequenced according to some logical, task-oriented, categorization scheme.

EXAMPLE 2 Items are sequenced by functional, alphabetical order, or date of creation.

c) Items at one level of the hierarchy should be distinguishable from items at other levels and cues should be provided to indicate the levels to which the items belong.

EXAMPLE 3 Sub-items in the hierarchy are indented relative to items at the level above them.

d) Items that are expanded to show their sub-items should be distinguishable from those that have not been expanded.

e) Items that contain sub-items should be distinguishable from those that do not contain lower level items.

f) If different types of items are presented in the same hierarchy, cues should be provided so that the user can differentiate among different types of objects (e.g. mail folders, discussion threads and documents).

9.3.6 Multiple column lists

If multiple column list elements are used, the following is recommended.

a) **Column labels**

The labels for the columns in the list should be placed above the list so that they are continuously visible even when the user scrolls the list of items.

b) **Too many columns**

If a list element has more columns than can be displayed in the dialogue box or form, a mechanism should be provided that enables the user to see all of the fields.

EXAMPLE 1 A horizontal scroll bar allows the user to view columns that are currently not visible.

c) **Discriminable columns**

The presentation of the columns should facilitate the user's identification of the column to which any item of information belongs (see ISO 9241-12:1998, 5.8.5);

EXAMPLE 2 Adequate spacing is provided so the user can tell where each column of information begins and ends.

EXAMPLE 3 A vertical line is used to separate each of the columns.

d) **Sorting of lists**

If it will support the user's task, a mechanism to enable the user to sort the list by any of the columns should be provided.

9.3.7 Cues for single or multiple-selection list boxes

A list box should provide a cue to indicate whether it allows multiple selection or only single selection.

EXAMPLE The label for the list box indicates that the user can select multiple items.

9.3.8 Selection of all items in a list

If the user needs to select all items in the multiple-selection list, a mechanism for quickly selecting all items should be provided.

EXAMPLE A button is presented next to the list that is labelled "Select All".

9.3.9 Cues for type of multiple selection

Information shall be provided that indicates whether or not multiple discrete items can be selected from a multiple-selection list or if the multiple selected items must be in a range.

EXAMPLE The multiple-selection list box contains a scrollable list of check boxes, showing the selection state of each item and allowing users to manipulate the selection state of each item independently.

9.3.10 Mechanisms for multiple selection

A multiple-selection list box that allows both range and discrete selection should provide separate mechanisms to allow a range to be extended and discrete selections to be made.

EXAMPLE Shift + Click extends the selection to include all items between the previous selection and the position of the pointer. Ctrl + Click adds the item under the pointer to the group of selected items.

9.3.11 De-selection in multiple-selection lists

Multiple-selection list boxes shall provide the user with mechanisms for de-selecting individual items.

9.3.12 Quick de-selection access in multiple-selection lists

A quick mechanism should be provided for de-selecting all items in the list with a single action.

9.3.13 Range selection in multiple-selection lists

If range selection is allowed, a mechanism should be provided for decreasing the range.

9.3.14 Quick access for list boxes

If the dialogue box or form allows the user to execute commands on list box items and the user will benefit from quick access, a mechanism for combining both selection of the item and execution of the selected action (which could be the default) should be provided.

EXAMPLE When the user double clicks on one item in a list of message types, a dialogue box for modifying the properties of the message type appears.

9.3.15 Combination boxes

If combination boxes are used and space is not an issue, combination boxes in which the list is always visible should be considered.

NOTE Because a combination box includes a list box and a text field, the recommendations for those list boxes and text fields also apply to combination boxes.

9.3.16 Stepper buttons

If the user must select one or more entries and display space is limited and/or the options need not be changed frequently, the use of stepper buttons (e.g. cycle buttons, spin buttons) or elements that cycle through a list of choices should be considered.

If stepper buttons are used, the following is recommended.

a) **Content**

Stepper buttons should be used only for sets of options that have a logical sequence.

b) **Typing values**

The user should be allowed to type in values in order to set a value quickly.

c) **Directionality**

The stepper button should provide a mechanism for moving through the values in either direction in the sequence (e.g. up or down, forwards or backwards).

EXAMPLE 1 In a box containing multiple fields, when the user selects the “minute” field, the minutes can be changed by the stepper arrow buttons, see Figure 34.

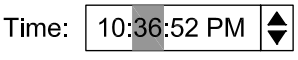

Figure 34 — Multiple stepper buttons used in a clock

d) **Clarity of direction**

The mapping between the labels on the buttons and the direction of change in values should be consistent with user population conventions.

EXAMPLE 2 Data are consistently sequenced so that using an up arrow increases the value for the data while using a down arrow decreases the value for the data, up to the point where the data wrap.

EXAMPLE 3 Days of the week are presented in a stepper and the left arrow moves to an earlier day and the right arrow moves to a later day in the week.

9.4 Tabs

9.4.1 Use

Tabbed elements provide a means of presenting a large amount of information in a small amount of space. These elements present multiple pages of settings using a metaphor of a set of index cards with attached tabs.

9.4.2 Logical sequence of tabs

The sequence of tabs within a tabbed element should match the logical order required by the user's task.

9.4.3 Preservation of ordering

The ordering of tabs in a row should be preserved as the user accesses different tabs in the tab set.

9.4.4 Rows of tabs

The number of rows of tabs presented in a tabbed dialogue should be minimized.

NOTE Multiple rows of tabs are confusing to the user when navigating between tabs in the different rows.

9.4.5 Alternatives to rows of tabs

If there are more categories than there is space for tabs in a single row, the following alternatives should be considered in the order in which they are given:

a) shrinking the tabs to fit the labels;

b) using the shortest possible meaningful tab labels;

c) restructuring the categories so there are fewer categories;

d) replacing the tabs with a different type of element;

e) scrolling of the tabs if the user will not need to see all tabs simultaneously to know the categories.

EXAMPLE One way of avoiding multiple rows of tabs is to have tabs that scroll as shown in Figure 35.

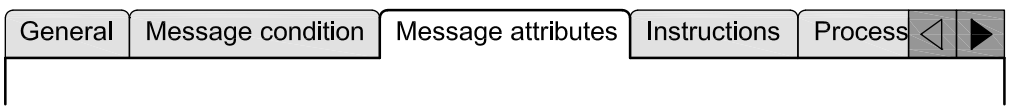

Figure 35 — Scrolling tabs

NOTE If the user expands the text on the tab through any means (operating system, browser, within the application, screen magnification), the tab will increase to fit the text.

9.4.6 Vertically displayed tabs

If vertically displayed tabs are used, the labels should be rotated as a whole rather than the characters being shown vertically displayed.

NOTE Vertically displayed tabs can be difficult for people to use.

Since vertically displayed tabs can create an accessibility barrier, where possible, consider using alternative methods.

EXAMPLE In Figure 36, the “Type” and “Options” vertically displayed tabs are oriented properly, while in Figure 37 they are not.

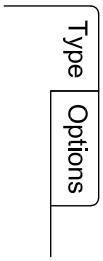

Figure 36 — Label meeting guideline

Figure 37 — Label not meeting guideline

9.4.7 Visual cues in tab elements

If beneficial to task performance, visual cues should be provided to indicate

a) the user has accessed the tab, and/or

b) the user has changed the information on a tab.

9.4.8 Hierarchies of tabs

If hierarchical tabs are used, different levels of the hierarchy should be easily distinguishable.

9.4.9 Push buttons in a dialogue box for a single tab

Push buttons that act on a single tab within the set of tabs shall be placed within the tab on which they act.

9.4.10 Push buttons in a dialogue box for all tabs

Push buttons that act on all of the tabs within the dialogue box shall be placed outside of the tabs.

EXAMPLE In Figure 38, the “OK”, “Cancel” and “Help” buttons are located outside the tabs.

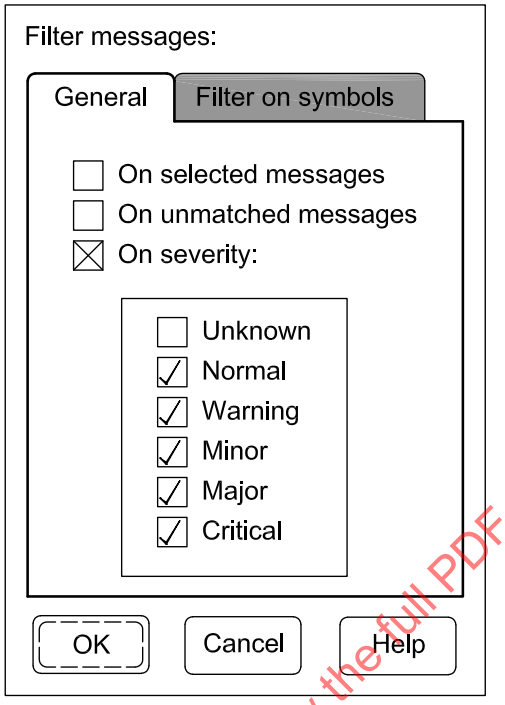

Figure 38 — Push buttons in a tabbed dialogue box

9.5 Scroll bars

9.5.1 General design considerations

ISO 9241-16 shall be consulted for information on the design of elements used by direct manipulation devices (e.g. mouse).

9.5.2 Presentation

Scroll bars should be provided only along the dimension in which additional information is available but not visible.

9.5.3 Meaningful units

Scrolling should be in increments that are meaningful to the user's task.

EXAMPLE In a word processing application, a single line of text is a meaningful unit for vertical scrolling.

9.5.4 Upward scroll button

When the user clicks once on the up scroll button, the information should scroll by the smallest meaningful vertical scrolling increment towards the bottom of the window (i.e. the user's view of the information will appear to move upwards).

9.5.5 Downward scroll button

When the user clicks once on the down scroll button, the information should scroll by the smallest meaningful vertical scrolling increment towards the top of the window (i.e. the user's view of the information will appear to move downwards).

9.5.6 Right scroll button

When the user clicks once on the right scroll button, the information should scroll by the smallest meaningful horizontal scrolling increment towards the left of the window (i.e. the user's view of the information will appear to move to the right).

9.5.7 Left scroll button

When the user clicks once on the left scroll button, the information should scroll by the smallest meaningful horizontal scrolling increment toward the right of the window (i.e. the user's view of the information will appear to move to the left).

9.5.8 Scroll box

9.5.8.1 The information should to be continually updated as the user drags the scroll box.

9.5.8.2 Alternatively, if it is not possible to continually update the information as it is being scrolled, the updated information should be displayed when the user pauses in the scrolling action.

9.5.9 Scroll shaft

9.5.9.1 When the user clicks once on the scroll shaft above the scroll box (elevator) in a vertical scroll bar, the information should scroll by a larger meaningful vertical scrolling increment towards the bottom of the window (i.e. the user's view of the information will appear to move upwards).

EXAMPLE In a word processing application, a click on the up or down scroll button scrolls by one line; whereas, a click on the scroll shaft scrolls by a window full of information.

9.5.9.2 When the user clicks once on the scroll shaft below the elevator in a vertical scroll bar, the information should scroll by a larger meaningful vertical scrolling increment towards the top of the window (i.e. the user's view of the information will appear to move downwards).

9.5.9.3 When the user clicks once on the scroll shaft to the right of the scroll box in a horizontal scroll bar, the information should scroll by a larger meaningful horizontal scrolling increment towards the left of the window (i.e. the user's view of the information will appear to move to the right).

EXAMPLE In a spreadsheet application, a click on the horizontal scroll button scrolls by a column; whereas, a click on the scroll shaft scrolls by a window full of information.

9.5.9.4 When the user clicks once on the scroll shaft to the left of the scroll box in a horizontal scroll bar, the information should scroll by a larger meaningful horizontal scrolling increment towards the right of the window (i.e. the user's view of the information will appear to move to the left).

9.5.10 Cues for scrolling availability

Cues shall be provided to indicate when there is no more information to be scrolled in a given direction.

EXAMPLE 1 The scroll buttons are displayed with an unavailable-state emphasis when there is no more information in the direction indicated by the button.

EXAMPLE 2 An auditory code is presented when the user scrolls to the end of the information.

EXAMPLE 3 The elevator touches the end of the scroll shaft to indicate the user is viewing the end of the information.

9.5.11 Cues for information to be scrolled

Cues should be provided to indicate how much information is currently being displayed in relation to the total amount of information.

EXAMPLE 1 In a page-based application, static text is presented to indicate that the current page is number three of ten.

EXAMPLE 2 The size of the elevator is adjusted so that it is proportional to the amount of information that is currently displayed. However, the size of the elevator is not reduced to such a small size, so that it can not easily be manipulated.

9.6 Push buttons and tool palettes

9.6.1 Cues for scope of action for push buttons

Consistent cues shall be provided to differentiate push buttons that affect the form or dialogue box as a whole from ones that affect a specific field or set of fields in the dialogue box or form. See also 5.4.8.

EXAMPLE All push buttons that affect the dialogue box as a whole are placed in a separate area at the bottom of the dialogue box.

9.6.2 When to use tool palettes

If there are objects or actions that the user will need to access frequently and quickly, those objects and actions should be made available to the user in a tool palette.

EXAMPLE In Figure 39, the user can select common word processing attributes in the tool palette.

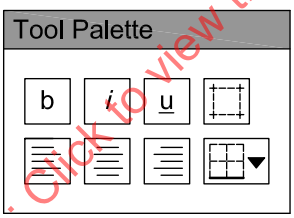

Figure 39 — Tool palette

9.6.3 Regaining tool palette space

If the user might need to use the display space occupied by the tool palette for other purposes, a means of hiding or removing the tool palette should be provided.

9.6.4 Alternative access to functionality

If the tool palette can be hidden or removed, or if it presents icons without a visible label, the objects and actions in the tool palette should be available through a redundant means (e.g. menus).

9.6.5 Separation of objects and actions

If a tool palette contains both objects and actions, the object icons should be grouped and separated from the actions.

9.6.6 Cascading palettes

If the available display area does not permit the display of all required items within a single palette, or if there is a very large number of items to be presented in the palette, then

a) items should be grouped into meaningful categories and placed into cascading palettes,

b) the cascading palettes should be accessible from the primary tool palette, and

c) a visual indicator should be provided to indicate which items provide access to cascading palettes.

EXAMPLE In the tool palette shown in Figure 40, when the user activates the down arrow, additional selections are provided on a cascading sub-palette.

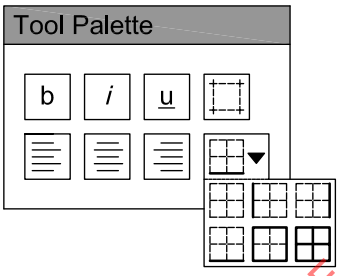

Figure 40 — Cascading tool palette

9.6.7 Keeping cascading tool palettes open

If the user will need to continually access items in a cascading palette, he or she should have a means of controlling whether the palette stays open or closes immediately after an item has been selected in the palette.

EXAMPLE The cascading palette can be dragged away from the primary palette that caused it to open. When dragged to another part of the desktop, the palette stays open.

9.6.8 Closing cascading tool palettes

If the user can leave a cascading palette open after selecting an item from within it, a means of closing the palette should be provided.

EXAMPLE The user can click on a graphic in a corner of the palette window, causing the palette to close.

9.6.9 Position of tool palette

Where feasible, the application should allow the user to choose where to position the tool palette.

9.6.10 Customization of tool palettes

If different users might use different groups of tool palette items or the use of items could vary across user tasks, a mechanism should be provided that allows the user to create custom tool palettes.

9.6.11 Context-sensitive help

If items will not have visible textual labels, easy access to a description of the object or an action associated with the tool palette item should be provided.

EXAMPLE Bubble help or tool tips that are presented when the mouse pointer rests over the tool palette item for a set amount of time.

10 Conformance

Conformance with this part of ISO 9241 is achieved by satisfying all the applicable requirements and by specifying all the recommendations that have been satisfied. Any requirements that have been determined not to be applicable shall also be listed, together with a statement of the reasons why they are not applicable.

Users of this part of ISO 9241 shall evaluate the applicability of each requirement (a “shall” statement) and should evaluate the applicability of each recommendation (a “should” statement) to determine whether it is applicable in the particular context of use that has been established for the interactive system that is being designed.

If a product is claimed to have met the applicable requirements and recommendations in this part of ISO 9241, the procedure used in establishing requirements for, developing and/or evaluating, forms shall be specified. The level of detail of the specification of the procedure is a matter of negotiation between the involved parties. Users of this part of ISO 9241 can either utilize the procedures provided in Annex B, or develop another procedure tailored to their particular development and/or evaluation environment.

Annex A
(informative)

Overview of the ISO 9241 series

This annex presents an overview of the structure of ISO 9241. For an up-to-date overview of its structure, subject areas and the current status of both published and projected parts, please refer to:

ISO 9241 series

The structure reflects the numbering of the original ISO 9241 standard; for example, displays were originally Part 3 and are now the 300 series. In each section, the “hundred” is an introduction to the section; for example, Part 100 gives an introduction to the software-ergonomics parts.

Table A.1 — Structure of ISO 9241 — Ergonomics of human–system interaction

Part	Title
1	Introduction
2	Job design
11	Hardware and software usability
20	Accessibility and human–system interaction
21-99	Reserved numbers
100	Software ergonomics
200	Human–system interaction processes
300	Displays and display-related hardware
400	Physical input devices — Ergonomics principles
500	Workplace ergonomics
600	Environment ergonomics
700	Control rooms
900	Tactile and haptic interactions

Annex B
(informative)

Checklist for applying this part of ISO 9241

B.1 Introduction

This annex provides an example of a procedure for determining whether the applicable requirements and recommendations in this part of ISO 9241 have been met. It is important to note that the procedure described below is provided as guidance and is not a rigid process to be used as a substitute for the standard itself. This approach provides a two stage process for 1) determining which requirements and recommendations are relevant, and 2) whether those relevant requirements and recommendations have been complied with.

Interface design depends upon the task, the user, the environment, and the available technology. Consequently, this part of ISO 9241 cannot be applied without knowledge of the design and use context of the interface and it is not intended to be used as a prescriptive set of rules to be applied in its entirety. Rather, it assumes that the designer has proper information available concerning task and user requirements and understands the use of available technology (this could require consultation with a qualified ergonomics professional as well as empirical testing with real users).

It is important that the evaluation procedure be based on an analysis of typical users, their typical and critical tasks and their typical usage environments. Forms evaluations generally fall into the two following categories:

— when users and user tasks are known, evaluators evaluate the product or observe representative users of the product in the context of accomplishing typical and critical user tasks in a typical usage environment;

— when specific users and user tasks are not known, evaluators evaluate all forms used in the product being evaluated.

Determination of whether a product meets a given requirement or recommendation can be based on the set of forms encountered during the evaluation described above. Forms that can be shown to be better than ones that meet the recommendations described in this part of ISO 9241 would also be accepted as meeting the requirements/recommendations of the standard.

Users of this part of ISO 9241 could demonstrate how they met its provisions by listing the forms evaluated (e.g. all forms or a task-derived sub-set of forms) and the results.

B.2 Applicability

Requirements are applicable unless the requirement involves form elements, or their attributes, not utilized in the particular form being designed and evaluated.

EXAMPLE 1 If form elements do not have different states, requirement 4.2.4 *would not be* applicable

The applicability of a recommendation is based on two factors.

a) Whether the conditional, if included as part of the statement, is true: a particular recommendation is (or is not) applicable when the conditional if-statement is (or is not) true.

EXAMPLE 2 If further information is not needed for completion, recommendation 4.2.9 *would not be applicable.*

b) Context of use: a particular recommendation may not be applicable because of user, task, environment and/or technology constraints, such as unknown user community, variations in tasks, noisy office, screen

resolution or lack of a pointing device. However, if the context of use did involve user characteristics, tasks or technology features addressed by a particular recommendation, that recommendation would be applicable. For example, if form input was allowed by means of choice entries, the recommendations in 9.2 should be evaluated to determine their applicability.

Typical methods for determining applicability include

a) system documentation analysis,

b) documented evidence,

c) observation,

d) analytical evaluation, and

e) empirical evaluation.

B.3 Compliance

If a recommendation is applicable on the basis of the criteria described above, it is then necessary to determine whether or not the recommendation has been met. Compliance is generally determined by means of one or more of the methods listed below:

a) measurements;

b) observation;

c) documented evidence;

d) analytical evaluation;

e) empirical evaluation.

B.4 Checklist[1)]

The checklist on the following pages is intended as an aid for both designers and evaluators of forms in evaluating both the applicability of, and compliance with, the requirements and recommendations in this part of ISO 9241. This checklist contains all of the requirements and recommendations and provides a logic structure to assist users in determining applicability. Many of the requirements and recommendations allow a number of alternative solutions.

B.4.1 Description of the checklist

B.4.1.1 Requirements/recommendations column

The first column of the checklist contains the text (sometimes shortened due to space limitation) of the requirements and recommendations. It should be noted that in the requirements statements, the “shalls” are shown in bold (i.e. **shall**). Since each requirement and recommendation is numbered with its clause number, users can look up the full text easily in the relevant clause or subclause.

1) Users of this part of ISO 9241 may freely reproduce the checklist in this annex so that it can be used for its intended purpose and may further publish the completed checklist.

B.4.1.2 Applicability columns

The two columns of the applicability portion of the checklist are provided for recording the result of the applicability determination by a checkmark in the “Y” or “N” column. The “Y” column for those requirements that are applicable for any form application are pre-checked.

B.4.1.3 Compliance columns

This part of the checklist is used to specify whether or not a particular requirement or recommendation has been met. If the result is positive, the “Y” column is checked; if the result is negative, the “N” column is checked.

B.4.1.4 Comments

The comment column provides space for additional statements and comments pertaining to each of the requirements and recommendations and can be used to indicate the source of the assessment (e.g. name of expert, title of documented evidence) as well as to describe unique methods when used. This description can include how these solutions relate to the form design requirements and recommendations and appropriate dialogue principles.

Table B.1 — Applicability and compliance checklist

Requirement/recommendation	Applicable		Compliance		**Comments** (including data sources)
	Y	N	Y	N	
4 Forms					
4.1 Selection Forms should be used for structured data entry tasks requiring input or modification of multiple data items.					
4.2 General requirements and recommendations					
4.2.1 Titles Forms **shall** be titled unless a title would be redundant. Any title **shall** clearly indicate the purpose of the form and differentiate it from other forms.	X				
4.2.2 Relationship of form title to structure The title should provide the user with a sense of the location of the form within the application structure.					
4.2.3 Visual coding Distinctive visual coding should be used to depict user entries, defaults, and previously entered data.					
4.2.4 Appearance of form elements If elements have different states, the current state of the element **shall** be clearly indicated by a perceptible cue.					
4.2.5 Form display density Unless required by the task or context of use, density of the textual information displayed should be limited.					
4.2.6 Complexity Complexity appropriate to the task and appropriate means should be used to reduce complexity where necessary.					
4.2.7 Restricting use of expanded dialogues Dialogue box expansions and extensions of dialogue should be restricted to functions needed by a subset of users or not needed for typical task.					

Table B.1 (*continued*)

Requirement/recommendation	Applicable		Compliance		Comments (including data sources)
	Y	N	Y	N	
4.2.8 Instructions Access to instructions should be provided.					
4.2.9 Help If further information is needed for completion, access to completion assistance or help should be provided.					
4.2.10 Overview of structure An overview or visual presentation of the structure should be provided.					
4.2.11 Modal dialogue boxes Unless necessary for command completion or to prevent further interaction, modeless dialogue boxes should be used.					
4.2.12 Accessibility Forms, dialogue boxes and their elements **shall** be in accordance with ISO 9241-171.	X				
4.2.13 Consistency Behaviour (conceptual, semantic and lexical) of forms should be consistent through platforms and systems.					
5 Information presentation					
5.1 General ISO 9241-12 **shall** be consulted for general guidance.	X				
5.2 Layout					
5.2.1 Paper document source If used, forms screens should be designed to be consistent with the layout of the paper source document.					

Table B.1 (*continued*)

Requirement/recommendation	Applicable		Compliance		**Comments** (including data sources)
	Y	N	Y	N	
5.2.2 No source document a) Elements should be arranged based on the sequence appropriate for the written language of the user, or b) Entry fields should be grouped by function, importance, etc. or input sequence optimized from the user's point of view.					
5.2.3 Order of required and optional fields Required fields should be positioned first unless such positioning is inappropriate to the user's task.					
5.2.4 Alphanumeric field alignment If appropriate to language content, entry fields should be aligned vertically in columns and left-justified.					
5.2.5 Numeric field alignment If groups of entry fields are all numeric and field lengths are different, these fields should be displayed right-justified. If numerical fields contain decimal markers, they should be aligned to the decimal marker.					
5.2.6 Allowable field values Information should be provided indicating allowable field values.					
5.3 Names and labels					
5.3.1 Consistency Naming of labels should be consistent through platforms and systems.					
5.3.2 Naming elements and groups of elements All elements and groups of elements **shall** have a name, whether or not the label is visually displayed.	X				
5.3.3 Label visual design ISO 9241-12:1998, 5.9, **shall** be consulted for general guidance on visual design of labels.	X				

Table B.1 (*continued*)

Requirement/recommendation	Applicable		Compliance		**Comments** (including data sources)
	Y	**N**	**Y**	**N**	
5.3.4 Differing label lengths (fields aligned vertically) If label lengths differ significantly and the task involves sequential data entry, labels should be right-justified and fields should be left-justified, or the fields should be left-justified and the labels placed above the fields and left-justified with the fields.					
5.3.5 Similar label lengths (fields aligned vertically) If field label lengths do not differ significantly, field labels and fields both may be left-justified.					
5.3.6 Label placement for search tasks If the user's task involves searching a group of labels for a particular label, field labels should be left-justified.					
5.3.7 Consistent label positions Labels should be consistently positioned for type of user-interface element or group of elements within a given context.					
5.3.8 Label position for check boxes or radio buttons Labels for check boxes or radio buttons should be consistently located to the right.					
5.3.9 Labels for screen reader use Individual check box or radio button label names should be included in the label name of the group.					
5.3.10 Groups of fields If fields are grouped, the label of the group should be placed at the top of the group.					
5.3.11 Multiple instances of a field If a label is used for multiple instances of a field, the label should be located above the column, or to the left of the row.					

Table B.1 (*continued*)

Requirement/recommendation	Applicable		Compliance		**Comments** (including data sources)
	Y	N	Y	N	
5.3.12 Descriptive field labels All fields should be clearly and unambiguously labelled to describe the purpose or function of content of each.					
5.3.13 Distinctive field labels Distinctive words and/or codes should be used for entry field labels, and the approach used applied consistently.					
5.3.14 Consistent labels Mapping between labels and functions for form elements should be consistent across software application, product or service.					
5.3.15 Symbols or units Symbols or units should be displayed as an additional label.					
5.3.16 Initial upper-case letter for field labels Text field labels should begin with an upper-case letter and the rest of the label should be in lower-case letters (for English).					
5.3.17 Implicit designator for elements Implicit designator should be assigned to each form element.					
5.3.18 Multiple pages 5.3.18.1 In sequence of multiple forms, location within the sequence **shall** be provided and placed at same location on each form.					
5.3.18.2 If the form is columnar, labels of the columns should be redisplayed.					
5.4 Visual cues in fields and forms elements					
5.4.1 Design considerations Cueing should be provided appropriate to the modality of presentation.					

Table B.1 (*continued*)

Requirement/recommendation	Applicable		Compliance		Comments (including data sources)
	Y	N	Y	N	
5.4.2 Fields with fixed length entry If the text entry fields are of fixed length, lengths should be explicit.					
5.4.3 Fields with maximum length If the text entry fields have a maximum length it should be explicit.					
5.4.4 Representation of optional and required entries Required and optional entry fields **shall** be represented so that the differences between them are perceptible.					
5.4.5 Inform on representation of required and optional entries Information on the representation used to distinguish between required and optional entry fields should be available to the user.					
5.4.6 Modifiable versus non-modifiable fields The user should be able to distinguish easily between fields that can be modified and those that cannot, by means of appropriate coding.					
5.4.7 Cues for entry format Cues for data entry format should be displayed within the entry field or in field labels (with abbreviations clear to user).					
5.4.8 Cues for push buttons Push buttons should have visual cues that indicate whether the command will be carried out immediately or if the user needs to provide additional input or confirmation before the command is carried out.					
5.4.9 Cues for expanding dialogues If a form element is provided to allow the user to expand a dialogue box or form to reveal additional functionality, the element should contain a visual cue or label indicating that the dialogue box or form will be expanded.					

Table B.1 (*continued*)

Requirement/recommendation	Applicable		Compliance		**Comments** (including data sources)
	Y	N	Y	N	
5.4.10 Cues for contracting dialogues If a form element is provided to allow the user to contract a dialogue box or form to hide functionality, the element should contain a visual cue or label indicating that the dialogue box or form will be contracted.					
6 Interaction					
6.1 Navigation					
6.1.1 Method The choice of method should take into account the particular user population and compatibility with the user's flow of work.					
6.1.2 Movement among fields The user **shall** be provided with the capability to move from field to field within a group and, if appropriate, to move to non-adjacent fields in other groups.	X				
6.1.3 Quick access If rapid access to a specific field in the form is required, a quick access mechanism should be provided.					
6.1.4 Return to initial field If appropriate to the task, a key or command should be provided to allow the user to return to the initial field on the form.					
6.1.5 Record cycling If the data is organized in sequential records and a form represents a view of data from one record, a mechanism should be provided for cycling from record to record, forwards and backwards.					
6.1.6 Pointing device and multiple forms If a pointing device is used for input and the task involves multiple forms, a mechanism to navigate between forms using the pointing device should be provided.					

Table B.1 (*continued*)

Requirement/recommendation	Applicable		Compliance		**Comments** (including data sources)
	Y	N	Y	N	
6.1.7 Conditional navigation If the value of a particular entry makes some of the entries following it, unnecessary, the system should move to the next appropriate entry field after entry is completed or upon tabbing. If possible, fields no longer needed should be deleted or protected.					
6.2 Navigation by tab keys and scrolling					
6.2.1 Tab keys or equivalents Manual tabbing by a tab key mechanism (or equivalent) should be provided for moving from field to field.					
6.2.2 Completely filled-in fields If all of the fields on the form are to be completely filled in and the forms are easily learned, auto-skip tabbing from field to field should be provided.					
6.2.3 Mutually exclusive fields Skipping remaining choices should be allowed when an entry has been made for one of the choices in the field.					
6.2.4 Form sections If the form is organized into meaningful information groups (sections), the user should be provided with the capability to move from group to group.					
6.2.5 Backwards tabbing If a form or dialogue box supports tab key navigation, a mechanism for tabbing backwards through fields should be provided.					
6.2.6 Forward wrapping When pressed in the last field, the forward tab navigation key should move to the first field in the form or dialogue box.					

Table B.1 (*continued*)

Requirement/recommendation	Applicable		Compliance		**Comments** (including data sources)
	Y	N	Y	N	
6.2.7 Backward wrapping When pressed in the first field, the backward tab navigation key should move to the last field in the form or dialogue box.					
6.2.8 Field scrolling If the maximum length of the data to be presented in a displayed field is longer than the field, a scrolling mechanism should be provided.					
6.2.9 Scrolling by pages When scrolling by page, there should be a minimum of one unit of overlap between the information that is presented.					
6.3 Input focus and cursors					
6.3.1 Keyboard focus Only one field at a time within the form or dialogue box **shall** have keyboard focus for a given user.	X				
6.3.2 Cues for keyboard focus 6.3.2.1 The field with keyboard focus should be indicated by the presence of a focus indicator.					
6.3.2.2 When a text field no longer has keyboard focus, it should not display a text cursor.					
6.3.3 Initial focus position When the dialogue box or form is first displayed, it should have keyboard focus and the focus indicator should be positioned automatically at the first entry field that must or may be completed by the user.					
6.3.4 Initial placement of text cursor a) If the task flow indicates that a portion of the text field is likely to be modified, then the text cursor should be placed at the end of that portion.					
b) If the task flow indicates that no specific portion of the text is most likely to be modified, the text cursor should be placed at the end of the text in the field.					

Table B.1 (*continued*)

Requirement/recommendation	Applicable		Compliance		**Comments** (including data sources)
	Y	N	Y	N	
6.3.5 Replacement of text in a field If the user's task will typically require the replacement of all, rather than a portion, of the text in a field, the entire text should be selected for replacement when the user first enters that field:					
a) if the user enters information into the field, the previously selected text should be replaced with the new text;					
b) if the user leaves the field without entering new information, the text that was initially in the field should remain in the field;					
c) a mechanism should be provided for positioning the text cursor within the field so that information can be entered without deletion of the previous text, in case the user wants to modify the existing text.					
6.3.6 Indirect regaining of keyboard focus If a field has lost keyboard focus and subsequently regains the focus indirectly due to the form or dialogue box regaining focus:					
a) if the field is a text field, the text cursor should reappear at the same position it had when the field lost keyboard focus;					
b) if the field is not a text field (e.g. a radio button group), the selection indicator(s) and/or selection cursor should reappear at the same position as when the field lost keyboard focus.					

Table B.1 (*continued*)

Requirement/recommendation	Applicable		Compliance		**Comments** (including data sources)
	Y	N	Y	N	
6.3.7 Regaining focus by clicking in a field If a field has lost keyboard focus and subsequently regains the initial focus by a click within the element:					
a) if the element is a text field, the focus indicator should move to the field and the text cursor should move to the position in the field where the click occurred;					
b) if the element is a check box or a radio button, the focus indicator should move to the specific button clicked and that button selected;					
c) if the element is a push button, that push button should be activated and keyboard focus should change, based on the action defined for the push button;					
d) if the element is a list box, the focus indicator should move to the list box and the selection indicator be shown on the item at the position where the click occurred.					
6.3.8 Indicators and cursor for multiple selection When a multiple-selection list box is entered, the selection cursor should be on the first item in the list box. There should be a selection indicator for each item in the list box that is currently selected.					
6.3.9 Indicator for single selection When a single-selection list box initially receives focus, the presence and location of the selection indicator should support the user's task:					
a) the selection indicator should be on the current selection or the default selection in the list box;					
b) if there is no current selection and a default selection is inappropriate to the task, no selection indicator should be present; nevertheless, the selection cursor should be on the first item in the list to allow keyboard navigation.					

Table B.1 (*continued*)

Requirement/recommendation	Applicable		Compliance		**Comments** (including data sources)
	Y	**N**	**Y**	**N**	
6.4 Input					
6.4.1 Considerations Considerations should include: user control of the dialogue at all times, the capability for the user to recover easily from errors, and the avoidance of requiring the user to input more information than necessary or input currently available information.					
6.4.2 Minimize cursor movement The user actions required to move the cursor from one entry field to the next should be minimized.					
6.4.3 Input device independency If multiple input device access is available to the application, all form elements should be operable by all available input devices (i.e. device independent).					
6.4.4 Pointing devices If a pointing device can be used for input in a form it should be usable for navigation as well.					
6.4.5 Switching between input devices The need for the user to switch between different input devices when filling in a form should be minimized.					
6.4.6 Incomplete text entry field If (the required) number of characters entered into the field does not fill the whole field, the user should be allowed to move directly to the next field.					
6.5 User control					
6.5.1 Changes or corrections If the user changes can impact a database accessed by multiple users and/or this results in negative consequences, the user should be allowed to go back to initial state of form or dialogue at any time and start over again, cancel entries, or change any entry before the form is processed by the computer.					

Table B.1 (*continued*)

Requirement/recommendation	Applicable		Compliance		**Comments** (including data sources)
	Y	N	Y	N	
6.5.2 Immediate processing of user input 6.5.2.1 If a change in a form results in only visual changes on the screen and local storage of the data, and if it can be reversed without negative consequences, user inputs should be processed immediately and then transferred to the database on user confirmation.					
6.5.2.2 If changes in a dialogue box can be reversed without negative consequences and user performance will benefit, user inputs should be processed immediately without a confirming action by the user.					
6.5.2.3 If user input is processed immediately without user confirmation, as in 6.5.2.1 and 6.5.2.2, the user should be allowed to go back to the initial state of the form or dialogue at any time and start over again, cancel entries, or change any entry before the form or dialogue box is closed.					
6.5.3 Identifying and locating errors					
6.5.3.1 Multiple fields If validation checking of multiple fields detects fields in error and if appropriate to the task, these fields should be indicated and the cursor should be placed on the first field in error and the user allowed to easily move through the fields in error to correct the entries.					
6.5.3.2 Dependencies If there are dependencies between fields, and if it is appropriate to the task, potential errors resulting from such dependencies should be indicated.					
6.5.4 Re-entering data If the field contains an error, the user should be required only to correct the erroneous part of the input.					
6.5.5 Disabled areas The user should not be able to enter information in areas of screen display not available for user input, and these areas should have visual cues and information to screen readers indicating that they are disabled.					

Table B.1 (*continued*)

Requirement/recommendation	Applicable		Compliance		Comments (including data sources)
	Y	N	Y	N	
6.5.6 Easy transmission If transmission of the form is required, transmission of the field entries to be processed should be accomplished by means of a simple explicit action. Transmission should take place no matter where the cursor is currently located on the form.					
6.5.7 User control information Unless it is obvious to the user, the form should state how to carry out the following actions, if provided:					
a) signal completion of the form or dialogue box and redisplay an empty form (with default values, if appropriate) for the entry of new data;					
b) signal completion of the form and redisplay the previously completed version of the form or dialogue box or a default version (template) of the form or dialogue box.					
c) dismiss the form or dialogue box without changing any data in the system;					
d) use “Undo”.					
6.5.8 Dismissing dialogue boxes 6.5.8.1 Single actions If involving the single application of a set of input, the dialogue box should contain a button that applies the input and dismisses it.					
6.5.8.2 Multiple actions If involving multiple applications of sets of user input, the dialogue box should have a form element that applies input without closing the dialogue box and a form element that closes the dialogue.					
6.5.8.3 Single and multiple actions If use may involve both a single application of input and repeated applications of input, the dialogue should provide a mechanism for applying the input without closing the dialogue, as well as a button for applying the input and closing the dialogue.					

Table B.1 (*continued*)

Requirement/recommendation	Applicable		Compliance		**Comments** (including data sources)
	Y	N	Y	N	
6.5.9 Temporary save If appropriate to task and constraints, temporary save function should be provided so the user can leave the form temporarily and return to it later without having to re-enter all the data.					
6.6 Feedback					
6.6.1 Information needed The user should be provided with information allowing him/her to control dialogue, recognize errors, determine the next course of action.					
6.6.2 Typed-in character echoing Typed-in characters **shall** be echoed back to the user, character by character, as entered.	X				
6.6.3 Cursor position and visibility 6.6.3.1 The cursor position **shall** always be clearly visible if it is within the currently displayed portion of the form.	X				
6.6.3.2 The software **shall** provide at least one mode where keyboard focus cursors and text cursors can be visually located by people with unimpaired vision without moving the cursor.	X				
6.6.4 Pointer position Position of the pointer **shall** always be clearly visible to user.	X				
6.6.5 Focus indicator An easily discriminable focus indicator **shall** be provided that allows the user to determine which field or form element in the dialogue box or form currently has keyboard focus.	X				
6.6.6 Field errors If a field contains an error, error feedback should be provided as soon as the user completes the field in a manner that is minimally disruptive of the continuation of the task.					

Table B.1 (*continued*)

Requirement/recommendation	Applicable		Compliance		**Comments** (including data sources)
	Y	**N**	**Y**	**N**	
6.6.7 Transmission acknowledgment If appropriate to the task, the system should provide an acknowledgment to the user that the transmission of the form entries has been accepted by the system.					
6.6.8 Database changes If a form or dialogue box changes a database, feedback that the database has been updated should be provided to the user.					
6.7 Access to forms and dialogue boxes					
6.7.1 Access mechanism If an application contains various forms and dialogue boxes, the user **shall** be provided with a mechanism to access a particular form or dialogue box.					
6.7.2 Direct form access If appropriate to the task and if forms can be accessed independently, the user should be able to select forms directly.					
6.7.3 Movement between forms If forms can be accessed independently, the user should be able to move from form to form, forwards and backwards, in a predefined sequence and without losing input.					
6.7.4 Hierarchical level movement If the set of forms is hierarchical, the user should be provided with the capability to move to both the next higher and lower level in the structure.					
6.7.5 Returning to initial form If the set of forms is hierarchical, the user should be provided with an obvious means of returning to the initial form (i.e. the form at the top of the hierarchy) from any form in the application.					

Table B.1 (*continued*)

Requirement/recommendation	Applicable		Compliance		**Comments** (including data sources)
	Y	N	Y	N	
6.7.6 Forms in a windows environment If more than one form can be displayed in a window environment:					
a) only the last selected form should be active and ready for input;					
b) the user should be provided with the capability to switch to another form to make it active.					
6.7.7 Default form If one form is more likely to be used than another, generally or for a particular task, user, environment and/or technology configuration, that form should be the initial form.					
6.7.8 Property dialogues If a dialogue box presents the properties of an object or item...					
a) Access The user should have a simple and consistent mechanism for accessing the property boxes.					
b) Association with object The property box should provide a means for the user to identify the object or item with which the properties are associated.					
c) Multiple properties If an object has a large number of properties associated with it, the properties should be categorized and grouped for presentation to the user.					
d) Templates If possible, the user should be able to take the property settings for a given object and apply them to a new or existing object.					
e) Modification If the user can modify the properties of objects, the properties box should provide a mechanism for making these modifications.					

Table B.1 (*continued*)

Requirement/recommendation	Applicable		Compliance		**Comments** (including data sources)
	Y	**N**	**Y**	**N**	
6.8 Default values					
6.8.1 Field default values Fields should contain default values whenever possible and appropriate to the task.					
6.8.2 Choice of system default values Whenever default values are used, the value should be chosen to support the user's task:					
a) system default values should not be destructive or lead to undesirably time-consuming activities;					
b) in a single-selection list box, initial default selection should be either the item most likely to be selected or the first item in the list;					
c) in a multiple-selection list box, default values should include the selection of the set of items most likely to be selected by the user;					
d) in a group of radio buttons, there **shall** be a default value initially selected;					
e) in a group of radio buttons, the default value should be the choice most likely to be selected;					
f) in a group of check boxes, each check box should be set to the value that will most likely to be selected;					
g) the default value in a text box should be the value most likely be entered by the user;					
h) if stepper buttons are used, initially displayed choice in a stepper button should be the most logical default choice.					
6.8.3 User-configured defaults If default values likely to vary across users but remain relatively consistent across tasks of a specific user, methods should be provided to allow the user to customize default values.					

Table B.1 (*continued*)

Requirement/recommendation	Applicable		Compliance		**Comments** (including data sources)
	Y	**N**	**Y**	**N**	
6.8.4 Return to system defaults If an application allows the user to customize default values, some means **shall** be provided to allow the user to return the settings to the system default values.					
6.8.5 Defaults in previously opened dialogue boxes and forms Dialogue boxes or forms that may be accessed multiple times should provide default values that support user's task.					
a) If likely that retaining values set by the user will minimize user steps or the need for changes in field values, the user settings should be presented as default values the next time the dialogue box or form is accessed.					
b) If unlikely that the user will want to retain previously set values, user-customized defaults should be presented as the default values the next time the dialogue box or form is accessed.					
c) If unlikely that the user will want to retain previously set values and user-configured defaults have not been set, system defaults should be presented as the default values the next time the dialogue box or form is accessed.					
6.8.6 Editable default values for text fields Text default field values should be editable by the user using conventional editing commands.					
6.8.7 Default values for groups of radio buttons If there is a default for a group of radio buttons, the default choice in the set **shall** be visibly selected when the field is first presented.					
6.8.8 Default values for groups of check boxes If there are default values for a group of check boxes, the default choice for each check box in the group **shall** be indicated as active or inactive when the field is first presented.					