INTERNATIONAL STANDARD

ISO 9241-125

First edition
2017-09

Ergonomics of human-system interaction —

Part 125: Guidance on visual presentation of information

Ergonomie de l'interaction homme-système —

Partie 125: Recommandations relatives à la présentation visuelle d'informations

Reference number
ISO 9241-125:2017(E)

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/TC 159, *Ergonomics*, Subcommittee SC 4, *Ergonomics of human–system interaction*.

This first edition of ISO 9241-125, together with ISO 9241-112, cancels and replaces ISO 9241-12:1998, which has been technically revised with the following changes:

— specific guidance relating to the presentation of visual information has been updated and extended (recommendations for presentation of information in other modalities will be addressed in future parts of ISO 9241);

— the characteristics of presented information have been elaborated with respect to ISO 9241-112;

— textual descriptions of the figures (alt text) have been added to enhance accessibility for sight-impaired users (in PDF, these are “pop-ups” which appear when the cursor is passed over the figure).

A list of all parts in the ISO 9241 series can be found on the ISO website.

Introduction

ISO 9241 is a multi-part International Standard that deals with both the hardware and software ergonomic aspects of human-system interaction.

ISO 9241-125 is intended for use by the following types of users:

a) the user interface designer, who will apply it during the development process.

b) the buyer, who will reference it during the product procurement process, and whose end users will gain from the potential benefits it provides.

c) project managers who are responsible for managing development processes.

d) designers of user interface development tools to be used by interface designers.

e) writers of software industry user-interface guidelines to be used by interface designers, e.g. “interface style guides”.

Guidance relating to presenting information provided in International Standards is intended to be applied to user-interface guidelines published by industry sources.

The ultimate beneficiary of this document will be the end user of the presented information. Although it is unlikely that the end user will read the standard or even know of its existence, its application by designers, buyers, and evaluators should provide user interfaces that are more usable, consistent and that enable greater productivity.

This document consists of general recommendations and conditional recommendations concerning presentation of information. General recommendations apply to most users, tasks, environments, and technology. In contrast, conditional recommendations are recommendations that apply only within the specific context for which they are relevant (e.g. particular kinds of users, tasks, environments, technology). Conditional recommendations have an “if-then” structure. The recommendations were developed primarily by reviewing the existing relevant literature and empirical evidence, then generalizing and formulating this work into recommendations for use by the interface designer and/or evaluator.

Ergonomics of human-system interaction —

Part 125:
Guidance on visual presentation of information

1 Scope

This document provides guidance for the visual presentation of information controlled by software, irrespective of the device. It includes specific properties such as the syntactic or semantic aspects of information, e.g. coding techniques, and gives provisions for the organization of information taking account of human perception and memory capabilities. Those of its provisions that do not apply to specific types of visual interfaces clearly indicate any limitations to their applicability. It does not address specific details of charts, graphs or information visualization.

This document can be utilized throughout the design process (e.g. as specification and guidance for designers during design or as a basis for heuristic evaluation). Its provisions for the presentation of information depend upon the visual design approach, the task, the user, the environment and the single or multiple technologies that might be used for presenting the information. Consequently, this document cannot be applied without knowledge of the context of use. It is not intended to be used as a prescriptive set of rules to be applied in its entirety but rather assumes that the designer has proper information available concerning task and user requirements and understands the use of available technology.

Some of the provisions of this document are based on Latin-based language usage and might not apply, or might need to be modified, for use with languages that use other alphabets. In applying those that assume a specific language base (e.g. alphabetic ordering of coding information, items in a list), it is important that care is taken to follow its intent of the standard when translation is required to a different language.

This document does not address auditory or tactile/haptic presentation of information or modality shifting for the presentation of visual information in other modalities.

NOTE ISO 9241-112 provides high-level ergonomic guidance that applies to all modalities.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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 9241-171:2008, *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.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at http://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

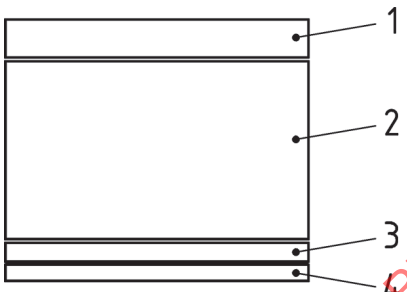
3.1
accessibility
extent to which products, systems, services, environments and facilities can be used by people from a population with the widest range of user needs, characteristics and capabilities to achieve identified goals in identified contexts of use

Note 1 to entry: Context of use includes direct use or use supported by assistive technologies.

[SOURCE: ISO 9241-112:2017, 3.15]

3.2
area
section or region of a display or window

Note 1 to entry: Figure 1 shows an example of a possible layout of different areas.



Key
1 identification area
2 input/output area
3 control area
4 message area

Figure 1 — Possible layout of different areas

3.2.1
identification area
area (3.2) where the title of the displayed information is provided

Note 1 to entry: The identification area can include an indication of the user's current location and task. It may also identify an application, file, or working environment.

3.2.2
input/output area
area (3.2) where information is received from users and/or presented to users

Note 1 to entry: This terms recognizes that an area can be used for both input and output, but does not necessarily have to be used for both.

3.2.3
control area
area (3.2) where control information and/or controls (user-interface elements) for interaction, command entry and command selection is provided

Note 1 to entry: Control information and/or controls (user-interface elements) for interaction, command entry and command selection may be positioned also to other parts of the display, e.g. the input/output area.

3.2.4
message area
area (3.2) where information such as status updates and/or other information e.g. is provided

Note 1 to entry: Information can include error messages, progress indication, feedback.

Note 2 to entry: Messages may be positioned also to other parts of the display, e.g. the input/output area.

3.3
code
technique for representing information by a system of alphanumeric characters, graphical symbols or visual techniques

Note 1 to entry: In general, alphanumeric codes are shorter than the full text needed to express the information content

Note 2 to entry: The term "code" is not to be confused with the terms "code" or "coding" in the computer science context, in which these terms refer to the instructions contained in an executable software program and the process of writing the instructions that comprise a software program.

Note 3 to entry: Visual techniques can include font, colour, highlighting.

3.3.1
mnemonic code
code (3.3) that is meaningful to the user and has some association with the words it represents

Note 1 to entry: Mnemonic codes frequently consist of alphanumeric characters, making them easier to learn and recall. Many mnemonic codes are abbreviations.

3.4
cursor
visual indication of where the user interaction via keyboard (or keyboard emulator) will occur

[SOURCE: ISO 9241-171:2008, 3.10]

3.5
field
user-interface element in which data is entered or presented

3.5.1
entry field
input field
field (3.5) in which users can input data or edit displayed data

3.5.2
read-only field
protected field
field (3.5) that contains data that cannot be modified by the user

3.6
focal colour
colour that is easily remembered and expressed by a short colour word (red, pink, yellow, blue, green, purple, orange, brown, grey, black, and white)

[SOURCE: ANSI HFES 200.5:2008, Clause 4, modified]

3.7
group
set of information items or user interface elements that are semantically related and perceptually distinct

[SOURCE: ISO 9241-112:2017, 3.14]

3.8
highlighting
display technique for emphasizing critical or important information and making it perceptually prominent

EXAMPLE Image polarity reversal, blinking, underscoring, use of colour, contrast enhancement (i.e. brightness coding), addition of graphics (e.g. draw a box around), size.

3.9
hue
attribute of a visual sensation according to which an *area* (3.2) appears to be similar to one of the perceived colours red, yellow, green or blue, or a combination of two of them

[SOURCE: ISO 9241-302:2008, 3.2.18]

3.10
icon
user interface symbol representing an object, action, and/or function

[SOURCE: ISO/IEC 11581-10:2010, 3.4, modified — Removal of the phrase "of the computer" and the Notes to entry.]

3.11
label
short, descriptive title for an entry or *read-only field* (3.5.2), table, control or other user-interface element

Note 1 to entry: In some applications, labels are classified as read-only fields.

Note 2 to entry: Labels include headings, field prompts, descriptive text (e.g. icon labels).

3.12
legibility
ability for unambiguous identification of single characters or symbols that may be presented in a non-contextual format

[SOURCE: ISO 9241-302:2008, 3.3.35]

3.13
list
horizontal or vertical sequential presentation of items in a display

Note 1 to entry: Items in a list can change according to the states of the application.

3.14
marker
symbol that is used for indicating a status or drawing attention to an item

EXAMPLE "*" is frequently used to indicate a required field.

3.15
pointer
graphical symbol that is moved on the screen according to manipulations or movements of a pointing device

Note 1 to entry: Users can interact with elements displayed on the screen by moving the pointer to that location and starting the interaction.

[SOURCE: ISO 9241-16:1999, 3.15]

3.16
saturation
chromaticness or colourfulness of an *area* (3.2) judged in proportion to its brightness

[SOURCE: ISO 9241-302:2008, 3.2.23]

3.17
spectrally extreme colour
extreme blue and extreme red

[SOURCE: ISO 9241-302:2008, 3.2.24, modified — Note to entry removed.]

3.18
table
orderly displayed data

Note 1 to entry: A table is often organized as a number of lists arranged in parallel columns or rectangular arrays, which are related to each other following a specific rule.

3.19
user-interface element
user-interface object
entity of the user interface that is presented to the user by the software

[SOURCE: ISO 9241-171:2008, 3.38, modified — Example and notes to entry removed.]

3.20
window
independently controllable region on the display screen, used to present objects and/or conduct a dialogue with a user

Note 1 to entry: A window is usually rectangular and delimited by a border.

3.20.1
primary window
window (3.20) that represents an operating system, an application, or an object

Note 1 to entry: It is possible to have more than one primary window presented at the same time.

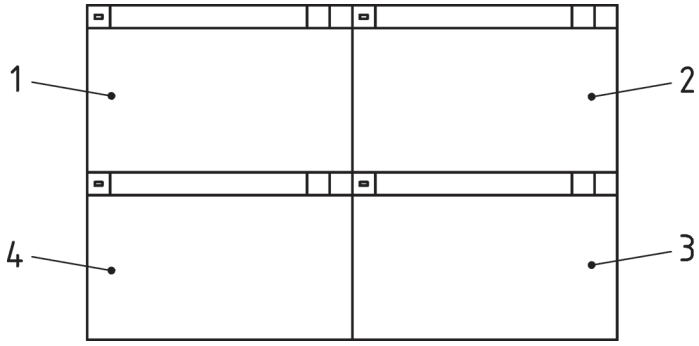
3.20.2
secondary window
window (3.20) arising out of user interaction with a *primary window* (3.20.1) displayed in the course of a dialogue

Note 1 to entry: A secondary window can also be a system initiated window.

3.21
windowing format
arrangement of multiple *windows* (3.20) which are displayed simultaneously

Note 1 to entry: There are several types of windowing formats such as tiled, overlapping, and mixed format.

3.21.1
tiled window format
side by side window format
format in which *windows* (3.20) are placed side by side and do not overlap

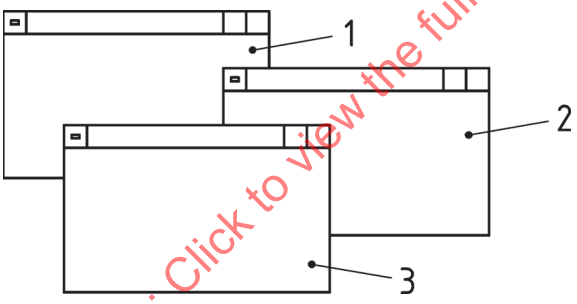


Key

1 window 1
2 window 2
3 window 3
4 window 4

Figure 2 — Illustration of a tiled window format

3.21.2
overlapping window format
format in which *windows* (3.20) partially or completely overlap each other



Key

1 window 1
2 window 2
3 window 3

Figure 3 — Illustration of an overlapping window format

3.21.3
mixed windowing format
format in which *tiled window formats* (3.21.1) and *overlapping window formats* (3.21.2) are combined

Note 1 to entry: The initial format can be tiled, but overlapping windows can be used to display transitory elements such as prompts and advisory messages.

4 Application of ISO 9241-125

4.1 Accessibility

Visual presentation of information shall be in accordance with ISO 9241-171, which gives specific requirements and recommendations for the presentation of information in the visual modality and the presentation of the same information in different modalities.

NOTE Conformity with ISO 9241-171 is achieved by satisfying all of its applicable requirements and by the provision of a systematic list stating how all of its applicable recommendations have been met (ISO 9241-171:2008, 7.2).

4.2 General guidance for presentation of information

The general principles provided by ISO 9241-112 should be applied to the presentation of visual information in the design of user interfaces. Each principle is accompanied by a non-exhaustive list of illustrative recommendations at various levels of detail. The principles are as follows.

— **Detectability**

Presented information is detectable if the information is presented so that it will be recognized as present.

— **Freedom from distraction**

Presented information is free from distractions if the information is presented so that required information will be perceived without other presented information interfering with its perception.

— **Discriminability**

Presented information is discriminable if the information is presented such that discrete items or groups of items can be accurately differentiated and if the items of information are presented in a manner that supports their association with or differentiation from other items or groups of items.

— **Interpretability**

Presented information is interpretable if it will be comprehended as intended.

— **Conciseness**

Information presentation is concise if only the necessary information is presented.

— **Consistency (internal and external)**

Presented information is consistent if items of information with similar intent are presented similarly and items of information with different intent are presented in different style and form within and across the interactive systems and the user's environment.

Guidance on *internal* consistency is related to the consistent use of conventions within the interactive system.

Guidance on *external* consistency is related to conventions known to the user from external sources.

4.3 Guidance on the presentation of visual information

Presentation of information in the visual modality should enable the user to perform tasks (e.g. search for information on the display) effectively, efficiently and with satisfaction.

The provisions for the presentation of visual information of this document support the general principles given in ISO 9241-112.

4.4 General issues for the display of visual information

4.4.1 Legibility of characters and symbols

Presentation of information in the visual modality should enable the user to unambiguously identify characters, icons, symbols and graphic objects in order to ensure legibility.

4.4.2 Character height

Minimum character height shall be used which is appropriate to the language used (e.g. Latin and Hindi/Devnagari characters 16′ of arc, Japanese characters 20′ of arc).

NOTE ISO 9241-303 provides further guidance regarding the interaction between character height, user characteristics, tasks and display imaging properties and the implications of this for legibility (see ISO 9241-303:2011, 5.5.4).

4.4.3 Scaling visually presented information

Users should be able to enlarge or reduce the size of the content that is being presented on a visual display.

NOTE The display size, resolution and viewing distance as well as the task, the users and the environment will determine the need to enlarge or reduce the presented information.

5 Visual structuring of information

5.1 Arranging and labelling information

5.1.1 Information location

Information should be located to meet user expectations and task requirements.

NOTE Information which is located according to user expectations minimizes search and access time.

5.1.2 Required information

All of the information required to perform a given task should be displayed in the input/output area. If this is not possible then the following is recommended.

a) The information required should be structured into subsets corresponding to task steps that each can be completely displayed (separately) in the input/output area.

b) The subsets should support appropriate sub-tasks and be meaningful to the intended users.

c) Splitting of information should not negatively affect task performance.

5.1.3 Consistent location of areas

Areas (e.g. identification, input/output, control, and message area) used in the dialogue within an application should be consistently located.

NOTE 1 The identification area is often located above the input/output area.

NOTE 2 In environments that don't use windows, the control area for command entries is often located at the bottom of the input/output area.

5.1.4 Density of displayed information

The density of displayed information should be such that the information is not perceived as being "cluttered" by the user and does not lead to a degradation of task performance (see ISO 9241-143:2012, 4.2.5).

NOTE For graphical user interfaces, other graphical elements such as lines, push buttons, and icons could be perceived as increasing the density of displayed information.

5.1.5 Distinction of groups

5.1.5.1 Ensuring visual distinction of groups

Groups should be perceptually distinguished by spacing and location. If necessary, other means should be used to improve distinctiveness (e.g. a box around the group). Figure 4 provides an example.

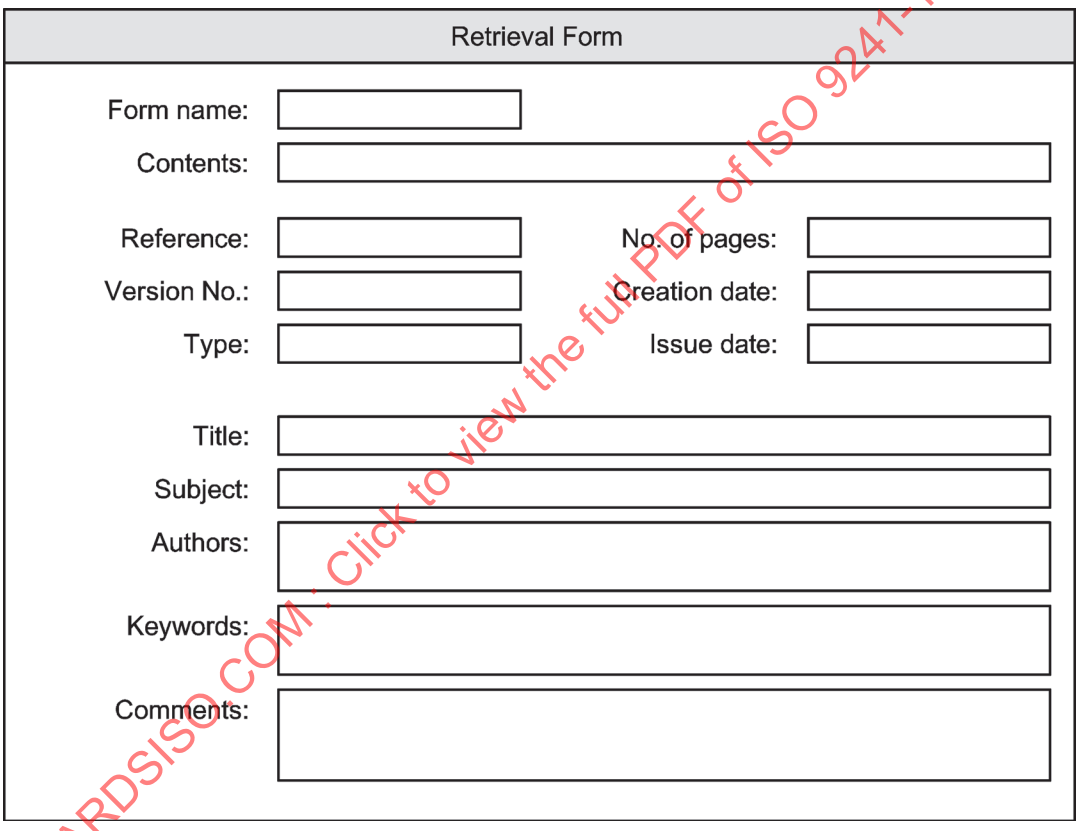


Figure 4 — Example of grouping

5.1.5.2 Application of Gestalt Principles

When grouping information the following Gestalt principles (or *laws*) should be applied to facilitate the distinction of groups containing information from one another.

a) **Law of proximity**

Elements in close spatial proximity are perceived as belonging to each other. This is demonstrated in the example at left in Figure 5 using parallel lines. Lines closer to their neighbours are seen as belonging to each other. The example in the middle shows a field arrangement which uses the effect of proximity in the correct way: labels and its fields are perceived as one visual unit. The example at right in Figure 5 shows an inappropriate arrangement in which Field 1 is seen as belonging to Label 2, which represents an unintended effect caused by the law.

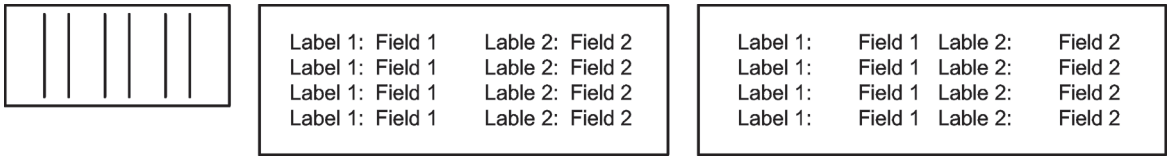


Figure 5 — Illustration of law of proximity — Left: illustration using vertical parallel lines — Middle: good example using user interface objects — Right: poor example

b) **Law of similarity**

Elements are perceived as belonging to each other if they are similar. In the example at left in Figure 6 the observer perceives columns instead of rows, although the information is semantically related across the rows. In the example at right in Figure 6, the column effect is prevented by inserting a dotted horizontal line.

NOTE Similarity applies to all types of user interface elements, not just to characters.

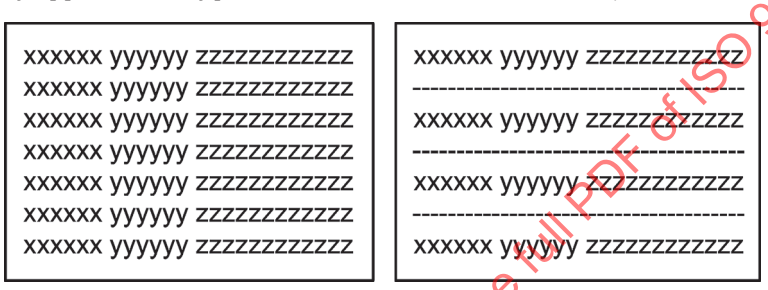


Figure 6 — Illustration of law of similarity — Left: poor design example — Right: good example

c) **Law of closure**

Non-existent parts of a figure are added or incomplete figures are completed automatically. This is the case with all groups of data that are spatially separated and where the observer attempts to build a coherent figure. The example at left in Figure 7 shows how fragments of a rectangle (corners) are sufficient to create the impression of a full rectangle. The example at right in Figure 7 represents a poor one, because of its tendency to prohibit the visual closure.

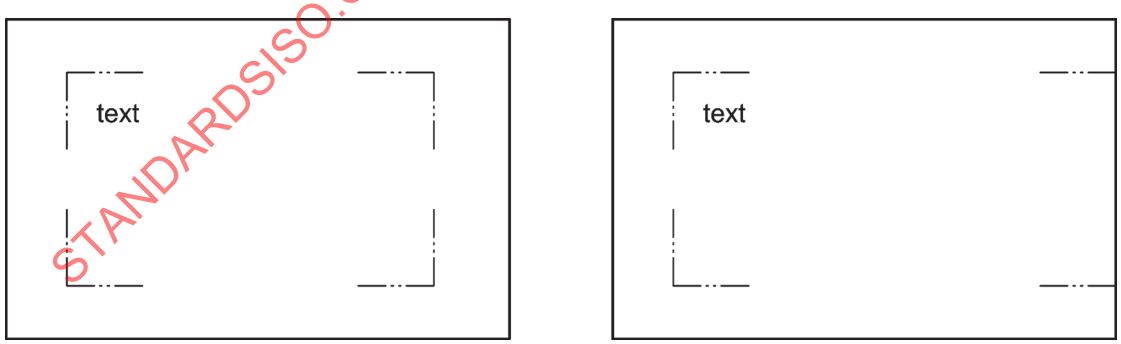


Figure 7 — Illustration of law of closure — Left: good example of using the law of closure — Right: poor example

5.1.6 Sequencing

If the task requires a specific sequence for the presentation of data, visual information should be placed in an order which supports that sequence.

5.1.7 Use of conventions

Groups of information should be arranged to follow common visual presentation formats, conventions and customs (e.g. addresses).

5.1.8 Functional grouping

If the task does not require a specific sequence, task-related information should be organized into groups that are semantically related and are meaningful to the user.

5.1.9 Arrangement of groups for rapid detection and discrimination

If the task requires rapid visual search for a user interface element, group of user interface elements or a piece of information, each user interface element, group and piece of information should, if possible, occupy a portion of the display area that is covered by the visual focus of the eye (i.e. central field of view). If the latter is not possible (e.g. because there is too much information to display) a sequencing of information presentation should be used that is meaningful to the user and facilitates fast searching.

EXAMPLE Depending on the size of the letters, the perception typically extends 12 to 15 letters to the right and 3 to 4 letters to the left in an asymmetrical way when reading Western text.

5.1.10 Consistency with paper forms

If paper forms are used in the task, the information display formats and the paper forms should be designed to match each other (see ISO 9241-143:2012, 5.2.1).

5.1.11 Labelling user interface elements

User interface elements (e.g. fields, items, icons and graphs) or groups of user interface elements (e.g. groups of check boxes or radio buttons) should be labelled unless their meaning is obvious and can be understood clearly by the intended users (for icon labels, see ISO 9241-14:1997, 8.4.1 and ISO 9241-143:2012, 5.3.2, and ISO/IEC TR 11580).

NOTE If labelling of icons is not practicable (e.g. due to space limitation), system initiated object identification (text appearing when the mouse cursor is moved over the icon, e.g. tool tip, quick info, balloon help) is an alternative.

5.1.12 Label designation

Labels should name the purpose and content of the designated information item.

5.1.13 Grammatical construction of labels

Labels should be grammatically consistent, e.g. consistent use of noun-verb combinations.

5.1.14 Label position

Labels should be consistently located adjacent to the designated user-interface element.

Label positioning should be consistent within types of user-interface elements (such as text boxes, list boxes, combination boxes, icons).

EXAMPLE 1 In an application, all the field labels are placed consistently to the left of the displayed field.

EXAMPLE 2 In an application, all the icon labels are placed consistently below the displayed icon.

5.1.15 Label position for individual check boxes or radio buttons

Labels for check boxes or radio buttons should be consistently located. In Western languages labels should be located to the right.

NOTE The recommended location can be affected by different cultural conventions, e.g. radio buttons or check boxes might be located on top of a label, when the convention is to read from the bottom up.

5.1.16 Separation of labels and associated information

Labels should be distinguishable from the information which they are designating (e.g. entry fields, items, icons, and graphs).

EXAMPLE 1 In an application, labels and the associated information are distinguished clearly by spaces.

EXAMPLE 2 In an application, colons are used to separate labels from entry fields.

5.1.17 Label format and alignment

Labels and fields should be consistently formatted (e.g. font, size, shape) and aligned (left or right-justified) (for alignment see ISO 9241-143:2012, 5.3.1).

5.1.18 Labels for units of measurement

The units of measurement for displayed information should be either included in the label or added to the right of the read-only or entry field unless the unit is obvious to the intended user.

EXAMPLE 1 Distance (km): 1,5.

EXAMPLE 2 Distance: 1,5 km.

5.1.19 Differentiating objects of identical type

If identical graphical representations (icons) are used for different objects, each representation should be given a unique identity by a text label.

EXAMPLE To differentiate between three printers, the system provides unique names attached to the printer icons, such as “Printer 1”, “Printer 2”.

5.2 Using windows to present information

5.2.1 Appropriateness of windows

The use of windows is more appropriate as more of the following task characteristics and system capabilities apply.

NOTE 1 The alternative to windows is to present the information in a browser-like single document interface, where there is no opportunity to manipulate the size and the location of windows individually.

a) **Task characteristics**

- The user evaluates, compares or manipulates multiple sources of information or multiple views of a single source of information (e.g. moving or copying information from one application to another).
- The user frequently alternates between tasks, systems, applications, files, sections or views.
- The user needs to preserve the broader task context while performing individual subtasks (e.g. accessing a customer's credit rating while processing a customer order).

— The user needs to attend to system or application events before primary task operations can continue (e.g. use of a “pop-up” window to display a caution or error message and request user acknowledgement).

b) **System capabilities**

Display size and resolution: the combined size and resolution of the display allow users to view meaningful amounts of information in multiple windows without requiring the users to perform numerous moving, resizing or scrolling/paging operations.

NOTE 2 Smartphones and screens for small devices might not be sufficiently large to allow for the display of multiple simultaneously visible windows (due to screen resolution and the amount of pixels available).

5.2.2 Multiple windows

5.2.2.1 Considerations for multiple windows

If information from different sources needs to be displayed or manipulated, multiple windows or a single window with multiple input/output areas (e.g. multiple panes) should be considered for use.

NOTE These sources might include different operating systems, applications, files within the same application, sections of the same file (e.g. beginning or end of a text file), views or versions of the same information (e.g. character-based and graphical view), or different parts of an application.

5.2.2.2 Optimizing use of space using multiple windows

If the task requires a large number of primary windows that the user needs to independently control, and display space is minimal, multiple windows should be considered for use.

5.2.2.3 Providing easy access to multiple windows

Each of the primary windows should have an easily available representation (e.g. in a menu bar or task bar) and an easy way to switch between them.

EXAMPLE The user uses a keyboard shortcut to select another window from a list of all windows available.

5.2.2.4 Overlapping window format

An overlapping window format can be used in cases where the task requires variable or unconstrained types, sizes, numbers, contents, and/or arrangement of windows.

An overlapping window format should be avoided on displays that are small or of such low resolution that users cannot view meaningful amounts of information in individual tiled windows.

5.2.2.5 Tiled window format

A tiled format should be used in cases where

— the task requires little or no variation in the size, numbers, contents, and arrangement of windows,

— continuous visual access to the information currently displayed (e.g. critical information, information necessarily used for the task) is required, or

— the processing required for the rapid manipulation and display of overlapping windows degrades system response time and user task performance.

5.2.3 Selection of window format

If appropriate for the task, users should be allowed to select the preferred windowing format and save the selected format as “default”.

6 Using user interface elements to organize information

6.1 Lists

6.1.1 List structure

Lists should be organized in a logical or natural order appropriate to the task.

NOTE If no such order is applicable, alphabetic ordering can be considered.

6.1.2 Item separation

Items and groups of items in a list should be visually distinct from one another to support visual scanning.

6.1.3 Alphabetic information

The format of lists of alphabetic information should depend on language conventions, e.g. vertical lists of alphabetic information are left-justified for languages which read from left to right.

NOTE Indentation can be used to indicate subordination in hierarchical lists (Figure 8 provides an example).

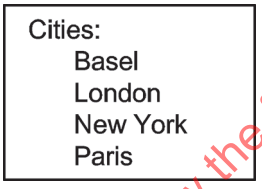


Figure 8 — Illustration of left-justified and indented alphabetic information

6.1.4 Numeric information

Numeric information without decimals should be right-justified (Figure 9 provides an example).

Numeric information containing decimals should be aligned with respect to the decimal sign (comma or point) (see Figure 9).

345	34,500
34	0,34
32345	323,4500

Figure 9 — Illustration of alignment of numeric information — Left: numeric information without decimal signs — Right: with decimal signs (using comma)

NOTE 1 Numeric information includes numbers having a comparable value. It does not include, e.g. phone numbers or product numbers.

NOTE 2 It is not necessary to right-justify column headers of columns containing right-justified numbers.

6.1.5 Fixed font size

A fixed font size with constant spacing should be used in numeric lists.

NOTE 1 Fonts with proportional spacing usually have constant spacing for numbers, whereas alphabetical characters have proportional spacing.

NOTE 2 Highlighting numbers by making them bold alters the breadth of a character so that it might no longer align with other numbers that have not been highlighted in this way.

6.1.6 Item numbering

When listed items are numerically labelled, the numbering should begin with “1” (one), not “0” (zero), unless this conflicts with user expectations.

6.1.7 Indication of relative position of displayed information

If the information exceeds the available input/output area, the relative position and proportion of the information currently displayed should be indicated in relation to the total amount of information (e.g. a scroll bar, slider, or “page x of y” indication).

6.1.8 Indication of list continuation

If a list extends beyond the available display area, an indication of list continuation should be provided.

EXAMPLE 1 The information “page 1 of 3” indicates which page is shown and how many pages exist (see left example of Figure 10).

EXAMPLE 2 A scroll bar indicates the list continuation (see right example of Figure 10).

NOTE Non-scrollable lists are often used in browser-oriented user interfaces.

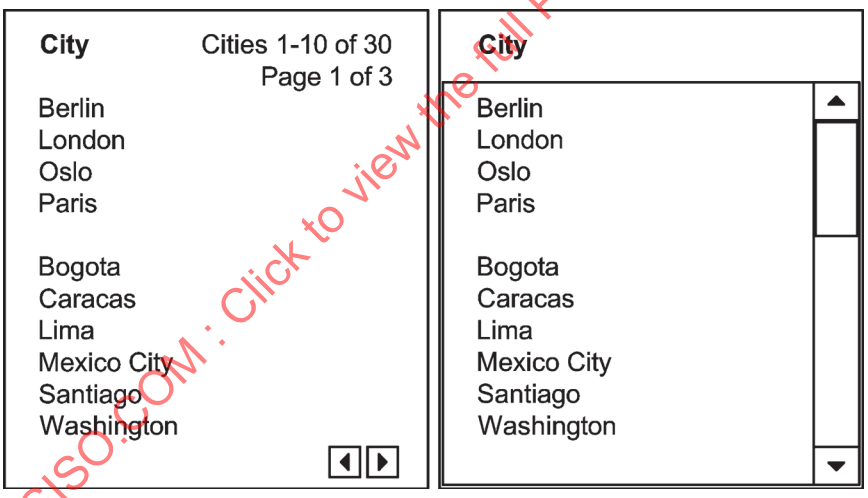


Figure 10 — Examples of indication of list continuation — Left: total number of pages and number of displayed page shown — Right: scrollbar indicates position and list continuation

6.2 Tables

6.2.1 List organization in tables

Tabular information should be arranged such that the material most relevant to the user or with the highest priority is displayed in the left most column and associated but less significant material in columns further to the right, unless this conflicts with user expectations (Figure 11 provides an example).

NOTE This applies to languages which are read from left to right.

Name	*Tel*	*Town*
Adena	40 12 03 89	Nantes
Bocage	40 34 90 00	Rennes
Colin	97 23 32 00	Paris
Diderot	82 32 32 04	Nancy

Figure 11 — Example of a list organization in a table

6.2.2 Maintaining column and row headings

If a table uses column and row headings and it extends beyond the available display then the headings associated with the visible columns and/or rows should always be visible.

6.2.3 Spacing as a visual technique for grouping information

Spacing should be used as a distinctive technique to facilitate visual scanning.

EXAMPLE Blank line (as spacing) approximately every five lines are used (Figure 12 provides an example).

City	***Country***	***Telephone Code***
Berlin	Germany	+4930
Bern	Switzerland	+4131
Brussels	Belgium	+322
Copenhagen	Denmark	+45
Lisbon	Portugal	+3511
London	United Kingdom	+44171
Madrid	Spain	+341
Oslo	Norway	+47
Paris	France	+331
Rome	Italy	+396
Vaduz	Liechtenstein	+4175
Valetta	Malta	+356
Vienna	Austria	+431

Figure 12 — Example of facilitating visual scanning by inserting blank lines

6.2.4 Column spacing

Columns in tables should be perceptually distinct.

NOTE Techniques for making columns perceptually discriminable include providing approximately 3 to 5 spaces on the left, lines between columns, colours, etc.

6.3 Entry fields

6.3.1 Entry field format

If a data entry field requires a specific format then the formats of the entry field should be clearly indicated (e.g. by a prompt or field help) unless obvious (Figure 13 provides an example).

NOTE This recommendation is most relevant for users not familiar with field format.

Figure 13 — Example of format indication

6.3.2 Entry field length

The length of a non-scrolling fixed length entry field should be clearly indicated (Figure 14 provides an example).

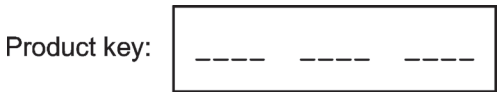


Figure 14 — Example of indicated field length

6.4 Windows

6.4.1 Unique window identification

A unique window identification (e.g. window name or file name or application name) should be provided for each window.

EXAMPLE In a specific office application, a window is identified by one or more of the following system specifications: system name, application, function, file name.

NOTE It could be useful to include an indication of the user's current location and task within the window identification.

6.4.2 Application information

Users should have access to information about the application and version (if applicable) that is running within each primary window.

6.4.3 Default window parameters

Default window sizes and locations should be designed to minimize the number of operations users need to perform to complete a task (e.g. windows are positioned where they will not obscure task-critical information in other windows).

6.4.4 Consistent window appearance within an application

Within an application, all windows of the same type should have a consistent appearance, if appropriate to the task.

EXAMPLE All windows for a particular help system have a consistent appearance.

6.4.5 Consistent window appearance within a multi-application environment

Within a multi-application environment, all windows of the same type should have a consistent appearance if they are used together and if appropriate to the task.

6.4.6 Indication of primary/secondary window relationships

The relationship between a primary window and its secondary windows should always be visually apparent.

EXAMPLE 1 In a specific office application, secondary windows are contained in the primary window.

EXAMPLE 2 Primary and secondary windows have common window border style, highlighting, colour.

EXAMPLE 3 Primary and secondary windows have a common identifying text label.

6.4.7 Identification of window control elements

Window control elements that perform different functions (e.g. a close-window control and a resize-window control) should be visually discriminable from each other and consistently placed in the same location in each window.

6.4.8 Location of window control elements

If windows contain window-control elements, those elements should be displayed in the same location in each window.

6.4.9 Size and location across sessions

If windows are retained from session to session, the user should be able to specify whether those windows will appear in the same size and location they had at the end of the previous session.

6.4.10 Duplicate or split windows

If users will need to simultaneously see two portions of application data at the same time, applications should provide a means to duplicate the window or to split the application window into multiple panes.

6.4.11 Resizing windows

If resizing will not prevent users from accessing necessary information and controls, applications should provide a means to resize both primary and secondary windows.

6.4.12 Resizing window panes

If separate panes are provided in the window, a means should be provided for the user to resize the panes to distribute the window space in accordance with task needs.

6.4.13 Predictable new window location

Newly created windows should appear in a predictable location on the display.

NOTE The location is predictable based on the context of existing windows and the logic of the application or operating system.

6.4.14 New windows offset

Unless specified otherwise by the user, the default position for newly created overlapping windows should be slightly offset from the previously created window. The degree of offset should be sufficient to allow users to select the previously created window.

6.4.15 New window on top

If windows are displayed in an overlapping format and the user has not specified otherwise, a newly created window should appear in the front-most layer.

6.4.16 Placement and size of new window

When a user takes an action that causes a new primary window to be opened,

— the window should be sized and positioned so that it is completely visible within the display area,

— the window should be created in accordance with user specifications, if any, but if there are no user specifications for size, the window should be created large enough so that the user can perform useful work in it.

6.4.17 Placement of secondary windows

Secondary windows should be positioned so that they will draw a user's attention without impeding the user's work.

EXAMPLE 1 A help window is placed next to the window from which it was requested, not on top of that window.

EXAMPLE 2 A message window or dialogue box is placed near the pointer or near the text cursor.

6.4.18 Warning for application termination

If removing a window causes the system to terminate an application, users should be warned if unsaved work will be lost and given an opportunity to save it (unless work is saved automatically).

6.4.19 Restoring hidden windows

If removing a window does not terminate the application, a mechanism should be provided that allows users to quickly identify and restore the window.

EXAMPLE A menu of "hidden" windows is provided so users can access windows that are not currently visible on the screen.

6.4.20 Identification cues for iconified window

The system should help the user locate and identify the icon that results from iconifying a window.

EXAMPLE 1 The icon is labelled.

EXAMPLE 2 The transition from window to icon is animated.

6.4.21 Predictable position of an icon representing a window

When a window is iconified, the default position for the icon should be predictable by the user and should not obscure other icons.

6.4.22 Visual cue in icon for important events

If an iconified window continues to receive or process data, a visual cue should be provided to signal important application events to the user.

EXAMPLE The icon for the window changes colour when an important event occurs.

6.4.23 Discriminable cue for application status

Icons representing terminated or non-running applications should be discriminable from those representing running applications.

6.4.24 Discriminable cue for restored windows

If a window's icon remains visible after the user has restored the window, the appearance of the icon should change to indicate that it represents a restored window.

EXAMPLE The icon is greyed when the application has been restored to the screen.

6.4.25 Identification cues for restored window

The system should provide a visual cue to help the user identify the restored window.

EXAMPLE The expansion from icon to window and/or bringing the restored window to the front is animated to help the user identify the location.

6.4.26 Input cursor location

If the system assigns input focus to a window, and it is appropriate to have an input cursor, the input cursor should automatically appear in that window at the place where users are most likely to perform an input action (e.g. the first data input field).

6.4.27 Visual cues for window with focus

Windows to which input focus has been assigned should have visual cues that visually distinguished them from other windows.

EXAMPLE 1 The window with input focus has a different border colour than other windows.

EXAMPLE 2 The window with input focus has a visual indicator in the title bar.

6.4.28 Manipulation of the size of windows

The system should provide mechanisms that allow a window to be sized either horizontally, vertically or in both dimensions simultaneously.

EXAMPLE Changing height and width is achieved by dragging window borders (for changes of a single dimension) or corners (for simultaneous change of both dimensions) to different screen positions.

6.4.29 Feedback provided during resize

The system should provide visual feedback indicating the dynamically changing size of the window during a resizing operation.

EXAMPLE During a resizing operation using the mouse, a rectangle indicates the shape the window will assume when the mouse button is released.

6.4.30 Original size indicator during resize

If users need to monitor the original size of a window while they perform a resizing operation, an indicator of the window's original size should remain visible until the resizing operation is completed.

EXAMPLE A user needs to maintain the original proportions of a window while shortening its width to one half of the original width. An outline of the original window remains visible during the resizing operation.

6.4.31 Independence of resize dimensions

If the window is resizable in two dimensions, then users should be able to resize the window along a single dimension (independently), or both dimensions simultaneously.

6.4.32 Scaling

If appropriate to the task, users should be allowed to scale the working area continuously or in one step.

NOTE This is not always technically feasible.

6.4.33 Sizing effects on window contents

When a window is resized, the contents should be adapted according to the method that is most appropriate for the user's task. The following are examples of how the contents are adapted as the window is resized.

EXAMPLE Three alternatives for resizing the window and its content:

a) the displayed window content is scaled in the same proportion as the window itself, and the identical content remains visible, but in a different size;

b) the displayed content keeps its size, but is rearranged to fit into the new window size as much as possible, while only parts, however, may remain visible;

c) the displayed window content is not scaled, but keeps its original size and arrangement and therefore may be only partly visible if the window is resized.

6.4.34 Scrolling and paging

If the amount of information to be displayed exceeds the available input/output area, the user should be provided with an easy means (e.g. horizontal or vertical paging/scrolling) of viewing portions of the information not currently displayed.

NOTE When the user needs to discern some relationship between separately displayed sets of information, it is desirable to display the two sets of information on a single screen, and scrolling/paging is not always appropriate. Techniques such as windows, split-screens, keywords, indexing, etc. can facilitate the overview of two sets of information.

6.4.35 Moving window contents in multiple units

If a window allows users to move its contents horizontally or vertically, and the associated information greatly exceeds that area which can be displayed within the window at its current size, a mechanism should be provided to scroll in steps corresponding to meaningful multiple and single units.

EXAMPLE Text can be scrolled in steps corresponding to single pages or to the size of the current window pane as well as in steps of single lines or columns.

6.4.36 Provide scrolling by smallest meaningful unit

If a window supports scrolling along a dimension, a mechanism should be provided that allows users to scroll in either direction in the smallest meaningful unit (e.g. a single line or column).

6.4.37 Provide scrolling by large units

If a window supports scrolling along a dimension, and the associated information greatly exceeds what can be displayed within the window at its current size, a mechanism should be provided that allows users to scroll in large units (e.g. page-sized vertical jumps).

6.4.38 Consistent direction of scrolling

The direction of scrolling initiated by pointing device inputs should be consistent with the direction of scrolling initiated by other input devices (e.g. up and down arrow keys).

6.4.39 Visual cue for group membership

A visual cue should be provided to indicate windows that are included in a window group.

7 Graphical objects

7.1 Cursors and pointers

7.1.1 Designation of cursor and pointer position

Cursors and pointers should indicate their position with distinctive visual features (e.g. shape, blink, colour and brightness).

NOTE For pointer appearance, format and shape, see ISO/IEC 11581-3.

7.1.2 Cursor occlusion of characters

Cursors should not obscure any character displayed in the position of the cursor.

7.1.3 Cursor and pointer location

Cursors or pointers should remain stationary until a change of position is initiated by the user or it is obvious to the user that the system changes the position according to the needs of the task.

7.1.4 Cursor "home" position

If there is a predefined home position for the cursor, that position should be consistent within active input/output areas.

7.1.5 Initial position for entry fields

When entry fields are first displayed, the cursor should be automatically positioned in the most appropriate entry field for the user's current task and expectations. The position of the cursor should be apparent to the user.

NOTE For languages read from left to right, the top left-hand entry field is the common default position for the cursor, when there is no other more appropriate entry field.

7.1.6 Point designation accuracy

If positioning accuracy is required, as in graphic interactions, the displayed pointer should include a point designation feature (e.g. cross-hair or V-shaped symbol).

7.1.7 Different cursors/pointers

Cursors and pointers used for different functions (e.g. text entry versus direct manipulation) should be visually distinct (see Figure 15).

Figure 15 — Example of visually distinct graphics for cursors and pointers for different functions — Left: vertical cursor for text entry — Middle: pointer over button (arrow) — Right: pointer over link (hand)

7.1.8 Active cursor/pointer

If more than one cursor or pointer is simultaneously displayed (e.g. in computer supported cooperative work), the active cursor/pointer should be visually distinct from those which are currently not active.

7.1.9 Multiple cursors and pointers

If the same displayed information is used by more than one user/operator simultaneously interacting, a visually distinct cursor and/or pointer for each individual user should be provided.

7.2 Icons

7.2.1 Guidance on design and use of icons

Guidance on the design and use of icons is provided in ISO/IEC 11581-10.

NOTE This standard will be used to provide additional guidance for inclusion in this clause.

7.2.2 Text alternative for icons

Icons should be provided with a text alternative (see ISO/IEC 11581-10:2010, 8.2.8 to 8.2.13 for further guidance).

NOTE ISO/IEC TS 20071-11 provides guidance on text alternatives for images.

7.2.3 Neutral icons

In choosing graphical representations for objects and actions, gender, racial, or other stereotyping should be avoided (see ISO/IEC 11581-10:2010, 8.3.2 to 8.3.6 for further guidance).

EXAMPLE In a medical application, the appearance of a doctor's head in an icon is gender neutral because it depicts the general features of a human face without suggesting which gender the face belongs to.

7.2.4 Cross-cultural icons

If icons need to be used in different countries, cultures, or presented to individuals speaking different languages, then the graphical images should avoid being culturally inappropriate. Otherwise, alternate, locally appropriate images should be used (see ISO/IEC 11581-10:2010, 8.3.2).

NOTE It is good practice to test or review icons with the different target populations for effectiveness and acceptability. The following graphical components are particularly likely to confuse or offend users in some cultures:

— hand and arm gestures;

— body parts and positions;

— facial expressions;

— words or alphabetic characters;

— punctuation or commercial symbols;

— animals;

— concepts represented by colours with culturally-specific meanings;

— puns and humour;

— religious symbols;

— stars or crosses that can be interpreted as religious symbols.

7.2.5 Ease of learning

To facilitate learning, concrete objects, familiar symbols, or metaphors commonly associated with what is being represented should be used (see ISO/IEC 11581-10:2010, 7.1.2).

7.2.6 Icons containing consistent cues

Icons that have concepts in common should use the same visual cues to represent those common concepts (see ISO/IEC 11581-10:2010, 7.2 and 8.3.7).

EXAMPLE Icons have consistent visual cues (arrows) to show options for controlling page views (see Figure 16).

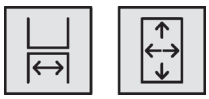


Figure 16 — Example of consistent cues (arrows)

7.2.7 National or international graphics

When a standard national or international graphic or symbol exists for the represented object or action, then it should be used (see ISO/IEC 11581-10:2010, 8.3.2).

EXAMPLE Using a circle with a slash through it to indicate that the user cannot drop a directly manipulated object at a specific location.

7.2.8 Reuse of existing icons

If there are existing icons for representing concepts in a given environment or application, the existing symbols should be used, rather than creating new icons for representing the same concept (see ISO/IEC 11581-10:2010, 8.3.4).

NOTE 1 Reuse of existing graphics or symbols in an icon will minimize the amount of learning that the user will have to undergo for the application or set of applications within an environment.

Existing icons should not be used to represent concepts that are not compatible with the concepts for which the icon is conventionally used.

NOTE 2 Some icons can be the intellectual property of corporations and are therefore protected.

7.2.9 User guidance for icons

If objects or actions are represented by an icon, but will not have a continuously visible textual label, then the application should provide easy access to a label or description that indicates what the graphic represents (see ISO/IEC 11581-10:2010, 6.6).

EXAMPLE When the pointer is over the graphic, a tool tip appears that indicates the object, type of object, or action that the graphic represents.

8 Coding techniques

8.1 General recommendations for codes

8.1.1 Introduction

Using codes can help designers reduce the amount of clutter (lack of order, poor spacing and displaying unnecessary information) on a visual display by representing information in “short form”

(or abbreviated form) using text and/or graphics (e.g. icons). Employing codes can also improve user performance in information entry by increasing speed and reducing errors. Poorly coded information causes a user's dialogue with the system to be slower and error-prone.

8.1.2 Meeting user expectations

Code constructions or rules of code should be designed together with the intended users and according to their expectations and tasks.

8.1.3 Explaining codes

General rules for code construction should be explained to users and should be easily available on request.

8.1.4 Distinctiveness of codes

Codes should be used which are perceptually distinct from each other.

8.1.5 Consistent coding

Codes should be used consistently with the same meaning or the same function.

NOTE If different applications are employed by the same user, then it is beneficial to task performance for codes to be used consistently with the same meaning or same function across applications.

8.1.6 Meaningfulness

Meaningfulness should be built into codes whenever and however possible.

NOTE 1 Meaningfulness is increased when clear associations exist between coded information and its intended meaning.

Preference should be given to mnemonic codes over arbitrary codes because mnemonic codes are meaningful.

NOTE 2 Task performance is more accurate and rapid, if codes are meaningful (Figure 17 provides an example).

UN - United Nations

© - Copyright

Figure 17 — Examples for meaningful coding

NOTE 3 Mnemonic codes frequently consist of alphanumeric characters which aid learning and recall. Many mnemonic codes are abbreviations.

NOTE 4 When designing mnemonic codes, it is important to consider whether or not codes and the information coded are subject to translation. Translation could stop mnemonic effects of the code.

8.1.7 Access to meaning of code

When the meaning of a code is not obvious to the user, then information about the meaning of the code should be easily available.

8.1.8 Use of standards or conventional meaning

Codes should be assigned according to established standards or conventional meanings for an intended user group.

EXAMPLE 1 In the United States: “Off”-position = switches in down position.

EXAMPLE 2 The maximum value of a horizontally oriented slider is at the rightmost position.

8.1.9 Rules of code construction

Codes should apply relevant rules for code construction consistently and unambiguously.

8.1.10 Codes for missing information

If it is important for the user's task to know that information is missing, then a code should be used to indicate the absence of this information.

EXAMPLE If a network connection is no longer available, the icon representing the network connection is shown crossed out rather than removed from the screen.

8.1.11 Partitioning long information items

Long information items should be partitioned into groups with a specified number of characters which are consistently used for entry and display.

EXAMPLE 1 A 10-digit telephone number is represented as 10 00 33 45 35 or 100 033 4535.

A space should be used as a separator, unless this conflicts with existing conventions or user expectations.

EXAMPLE 2 A 6-digit bank code is represented as 339 456.

8.2 Alphanumeric coding

8.2.1 Length of character strings

Simple codes (representing a single attribute of information) should be short, preferably 6 or fewer characters (consistent with providing meaningfulness, unique codes and ability to add additional codes).

NOTE There are inevitable trade-offs among these factors (for example, using the fewest number of characters conflicts with the goal of supporting the ability to add additional codes).

8.2.2 Alphabetic vs. numeric codes

Alphabetic codes should generally be used rather than numeric codes, unless it can be shown that numeric codes offer greater meaningfulness to the intended users for a particular task.

EXAMPLE http://www.iso.org is used instead of http://138.81.11.132.

8.2.3 Same meaning for upper and lower case characters

If alphabetic coding is used for input, upper case and lower case letters should have the same meaning, unless this is contrary to user expectations.

8.2.4 Visually similar alphanumeric characters

When using alphanumeric characters such as O (upper case) and 0 (zero), B and 8, S and 5, upper case "i" and lower case "L" (i.e. "I" and "l"), a type of font should be used which ensures the unambiguous visual differentiation between those characters.

NOTE The "i" and the "L" are to be understood as the opposite example to ensure unambiguity for the reader. Abbreviations for alphanumeric codes.

8.2.5 Length of abbreviations

The length of abbreviations should be as short as possible.

NOTE The length will depend on the number and similarity of words to be abbreviated.

8.2.6 Abbreviations of different length

If in a set of abbreviations of equal length some abbreviations can be shortened without ambiguity, this should be permitted to minimize required keystrokes.

8.2.7 Truncation

Truncation to construct codes should be used, when this can be done without ambiguity.

EXAMPLE The first three letters "abb" of the command "abbreviate" are used for truncation.

NOTE When the truncation leads to identical characters, a meaningful method can be applied to make truncated codes different (e.g. an additional character is used which makes all codes different from each other).

8.2.8 Deviation from the rules of code construction for abbreviations

If an abbreviation must deviate from the rule used for code construction (e.g. identical words, misleading), then the extent of the deviation should be minimized. If more than 10 % of all abbreviations are deviations, the rule used for code construction should be changed.

8.2.9 Conventional and task related abbreviations

Conventional and task related abbreviations should be used to meet user expectations.

8.3 Graphical coding

8.3.1 Distinctive states of user-interface elements

Graphical coding techniques should be used to indicate different states of user-interface elements.

EXAMPLE 1 The active window is visually distinguished from other windows by a different pattern of the window border.

EXAMPLE 2 The state of pressed push buttons is visually distinguished from the non-pressed push buttons by different shadows.

NOTE See ISO 14915-2:2003, 7.9.

8.3.2 Levels of graphical codes

The number of levels or degrees of graphical coding that need to be perceived should be limited.

EXAMPLE In an application, no more than three size codes are used.

8.3.3 Three-dimensional coding

The use of graphical techniques to create the perception of three dimensions should be considered if it would add additional information to a graphical element and not distract from the task.

EXAMPLE Buttons are represented three-dimensionally to create the impression that they can be activated.

8.3.4 Coding with geometric shapes

For every category of information, a unique and discriminable geometric shape should be used. The overall number of different categories and geometric shapes to be displayed should be minimized.

8.3.5 Coding with different types of lines

If coding by different appearances of lines is used, variations in line type (e.g. solid, dashed, dotted) and line width (boldness) should be clearly discriminable.

NOTE Line coding can be used e.g. on maps and graphs.

8.3.6 Line orientation

If line orientation is used for coding a direction or value, then contextual information should be provided, so that direction or values are accurately identifiable (Figure 18 provides an example).

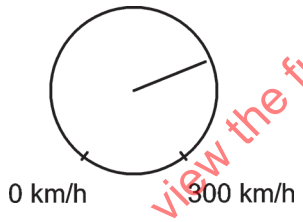


Figure 18 — Example of contextual information to facilitate discrimination of line orientation

8.4 Colour coding

8.4.1 Redundant colour coding

Colour shall never be the only means of coding.

NOTE 1 Some people discriminate certain colours poorly or cannot discriminate on the basis of colour at all.

NOTE 2 See ISO 9241-171:2008, 10.4.1.

Colour should be used together with some other coding techniques (see 9.2.1).

NOTE 3 Redundant coding is particularly important when colour is used to signal, or to enhance searching or classification. The goal is that when a specific colour is important for the user's task, the colour is not the only method of coding the particular piece of information.

NOTE 4 Memory for items that have to be recalled a certain amount of time after being removed from the screen can be enhanced by providing redundant colour coding.

8.4.2 Colour coding for people with colour deficits

When it is not possible to make the colour code redundant, desaturated colours or discriminable different grey levels for people with colour deficits should be used.

8.4.3 Legend showing colour codes

A legend showing colour codes and their associated meanings should be displayed or should be made easily available.

NOTE 1 Generally, six or more levels of colour coding necessitate a persistent legend.

NOTE 2 To create a consistent and usable order of colour codes in the legend, context of use information can be useful.

8.4.4 Redundant codes displayed in legends

Where redundant coding techniques, including colour, are used on a display (e.g. dot patterns) the redundant coding techniques should be shown together in the legend in order to clearly show their association.

EXAMPLE Legends for age groups are shown adjacent to the colour and pattern combinations used to code them.

8.4.5 Different use of one colour

Within an application, cues should be provided so that the user is aware that the context in which a colour is being used has changed and in the new context the meaning of the colour is different from that in the previous context.

EXAMPLE On one screen red is used as warning text for an error and on another it is used to indicate a hot temperature in a graphic. In the graphic, a legend is provided to explain the meaning of red in its context.

NOTE 1 The specific situations in which the colour is used might be very diverse.

NOTE 2 It is important to convey to the user how colour is being used when it is being used to mean different things in different contexts.

8.4.6 Limiting the number of colours

In general, the number of colours simultaneously used for coding should be minimized.

8.4.7 Colour assignment to categories of information

If colour is used for categories of information, then all the items from the same category should be displayed in the same colour.

EXAMPLE 1 A screen may have a set of meters implemented in software showing the states of various subsystems in a plant. Use one colour for all the meters that have passed critical, another for those reflecting systems that are normal, and a third for those that are approaching critical. The meters themselves will have colour, and if appropriate for the task, subsystems that are interrelated might all have the same colour (e.g. a dark grey). Independent, but interrelated, subsystems might have a different colour (e.g. a different shade of grey).

NOTE 1 The association mechanism that groups texts and/or figures in the same colour works best when not more than three or four colours are present on the screen.

NOTE 2 Another technique for showing that objects should be treated together is to use colour to demarcate the area in which they are placed on a display.

EXAMPLE 2 In a particular system all types of messages indicating dangerous situations have been identified as a single category of information. The colour red is used as the background colour for these messages.

8.4.8 Colour coding conventions

Familiar colour coding conventions should be followed, taking the context into account (e.g. red = warning; yellow = caution; green = OK or available).

NOTE The conventions that may be familiar to the user include

— task conventions, such as the coding (e.g. black = 0 and brown = 1) used for resistors,

— cultural conventions, such as red for warning,

— concrete identification, such as “blue sky” and “red hot”, and

— linguistic conventions and metaphorical associations, such as “blue Monday”.

8.4.9 Number of colours used

If colour coding is used, then the colours should be readily distinguishable by the user. It is preferable to use no more than six colours in addition to black and white (see ISO 9241-303:2011, 5.7.6.4).

NOTE This maximum does not refer to colours within images or graphical representations also on a display.

8.4.10 Assignments based on cultural conventions

Colour assignments should be used in accordance with cultural conventions.

EXAMPLE Western cultural conventions for the use of colour include

— Stop, Hot, Danger, Error, Extreme Warning, Severe Alert, Emergency, Alarm = Red.

— Caution, Potential or Mild Warning, Requires Attention, Slow, Moderate Alert = Yellow.

— Go, Safe, Normal, Good, Proceed = Green.

— Cold = Blue.

— Inactive, Unavailable Option or Choice = Grey.

8.4.11 Cross-cultural design

If an application is going to be used internationally or is intended to support more than a single user population, then the colour coding should either be culturally neutral or allow each user to easily select a palette that automatically maps the culturally appropriate colours to the relevant functions and/or meanings.

8.4.12 Colour assignments for special and temporary states

High contrast colours and colours that fit the conventions held by the user population should be used to signal a special and temporary state, such as a warning, or to identify emergency buttons.

NOTE This is one area where it is appropriate to use a highly saturated colour, because the state is temporary and the contrast with the generally less saturated interface is part of what will draw the user's attention to the warning.

EXAMPLE Change the colour of the background of an entry field to red to indicate that it contains an error.

8.4.13 Continuous scales

If colour is used to show values of a continuous variable, then changes in the current value should be shown by a corresponding change in one or more colour attributes.

NOTE 1 If colour is used to code values of a more or less continuous variable, then it helps if there is an apparent sequence to the colour code.

NOTE 2 It might be difficult for some users to discriminate between subtle changes in colour. Colour changes might not be appropriate when users need to identify the exact value that is being represented.

8.4.14 Coding differences

To emphasize that items or categories of items are distinctly different, the colours used to code the items or categories should be distinctly different.

8.4.15 Coding relative values

When the relative — rather than the absolute — values of a variable are important, gradual colour changes in hue, saturation, or brightness should be used to show the relative values of the single variable.

EXAMPLE 1 For reading topographic maps, the variation in hue might be used to show the relative height of displayed areas.

EXAMPLE 2 Under appropriate circumstances, users view a spectral order as a natural one and intuitively select colours in a spectral progression.

NOTE Small changes in extreme reds, extreme blues, and purples are more difficult to detect than corresponding changes in other colours, such as yellow and green. When large or abrupt changes in a variable should be indicated, large hue changes are necessary.

8.4.16 Ordered coding

When variation in hue, saturation, or brightness is used for coding relative values, the assigned code values should be ordered so that the darkest and lightest shades correspond to the extreme values of the coded variable.

EXAMPLE A continuous increase in temperature is represented by a change in hue from blue to red.

NOTE Different hues are most appropriate for discrete rather than for continuous data; general changes in saturation or brightness are best to indicate continuous changes in data.

8.4.17 Consistency of colour codes

If categories of information are logically similar, then the same or related colours should be used when assigning colours to the categories.

EXAMPLE 1 The same colour is used as a background colour in a dialogue box and as the background for read-only text fields.

8.4.18 Change of state

If colour is used to indicate a state, then a change in colour should be used to indicate a change in state.

EXAMPLE 1 Changing a green word reading “Normal” to a red word reading “Danger” when an alarm is triggered.

EXAMPLE 2 Changing a yellow area that is potentially selectable to a darker yellow when it has been selected and when typing.

While the user will focus their attention on a change of colour if they see the change of colour as it is happening, they might not notice the change of colour if not looking at the screen. Therefore, if the state change is important, make the redundant coding noticeable (e.g. use an auditory signal).

8.4.19 Additional guidance on the use of colour

Additional guidance on the use of colour can be found in Clause 9.

8.5 Markers

8.5.1 Special symbols for markers

Markers (e.g. *) should be considered for drawing attention to selected alphanumeric items (Figure 19 provides an example).

NOTE Markers are used for permanent selection.

Region	Country	City
Europe	Germany	Berlin
	UK	*London
	Norway	Oslo
	France	*Paris
	Italy	*Rome

Key

* cities with more than six million inhabitants

Figure 19 — Illustration of usage of markers

8.5.2 Markers for multiple selection

A marker should be used to indicate multiple selection.

EXAMPLE Check boxes can be used to indicate multiple selection in a list.

8.5.3 Unique use of symbols for markers

Markers should be used consistently.

Where possible, these symbols should not be used for any other purpose or displayed under conditions where confusion with other markers might occur.

8.5.4 Positioning of markers

Markers should be positioned close to the items marked. However, the markers should not appear to be part of the displayed items. Markers and items should be designed and positioned in a way which allows them to be identified clearly by the users (Figure 20 provides an example).

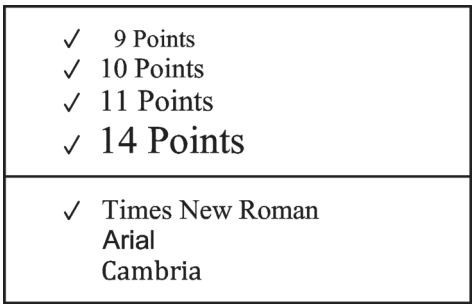


Figure 20 — Illustration of positioning of markers