

Ergonomics of human-system interaction

Part 171: Guidance on software accessibility (ISO 9241-171:2008)

ICS 13.180; 35.080

National foreword

This British Standard is the UK implementation of EN ISO 9241-171:2008.

The UK participation in its preparation was entrusted to Technical Committee PH/9, Applied ergonomics.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 October 2009 .

© BSI 2009

ISBN 978 0 580 69257 4

Amendments/corrigenda issued since publication

Date	Comments

English Version

Ergonomics of human-system interaction - Part 171: Guidance on software accessibility (ISO 9241-171:2008)

Ergonomie de l'interaction homme-système - Partie 171:
Lignes directrices relatives à l'accessibilité aux logiciels
(ISO 9241-171:2008)

Ergonomie der Mensch-System-Interaktion - Teil 171:
Leitlinien für die Zugänglichkeit von Software (ISO 9241-
171:2008)

This European Standard was approved by CEN on 28 June 2008.

CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration. Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN Management Centre or to any CEN member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the CEN Management Centre has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and United Kingdom.



EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

Management Centre: rue de Stassart, 36 B-1050 Brussels

Foreword

This document (EN ISO 9241-171:2008) has been prepared by Technical Committee ISO/TC 159 "Ergonomics" in collaboration with Technical Committee CEN/TC 122 "Ergonomics" the secretariat of which is held by DIN.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by January 2009, and conflicting national standards shall be withdrawn at the latest by January 2009.

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

According to the CEN/CENELEC Internal Regulations, the national standards organizations of the following countries are bound to implement this European Standard: Austria, Belgium, Bulgaria, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and the United Kingdom.

Endorsement notice

The text of ISO 9241-171:2008 has been approved by CEN as a EN ISO 9241-171:2008 without any modification.

Contents

Page

Foreword.....	iv
Introduction	vi
1 Scope	1
2 Normative references	1
3 Terms and definitions.....	2
4 Rationale and benefits of implementing accessibility	8
5 Principles for designing accessible software.....	8
6 Sources of variation in user characteristics	9
7 How to use this part of ISO 9241	10
7.1 General.....	10
7.2 Conformance	10
8 General guidelines and requirements	11
8.1 Names and labels for user-interface elements	11
8.2 User preference settings.....	14
8.3 Special considerations for accessibility adjustments	16
8.4 General control and operation guidelines.....	17
8.5 Compatibility with assistive technology	21
8.6 Closed systems.....	26
9 Inputs	26
9.1 Alternative input options	26
9.2 Keyboard focus.....	28
9.3 Keyboard input.....	29
9.4 Pointing devices	35
10 Outputs	39
10.1 General output guidelines	39
10.2 Visual output (displays)	39
10.3 Text/fonts	40
10.4 Colour	41
10.5 Window appearance and behaviour	42
10.6 Audio output.....	45
10.7 Text equivalents of audio (captions)	47
10.8 Media	47
10.9 Tactile output	48
11 On-line documentation, “Help” and support services.....	48
11.1 Documentation and “Help”	48
11.2 Support services.....	49
Annex A (informative) Overview of the ISO 9241 series.....	51
Annex B (informative) List of requirements.....	55
Annex C (informative) Sample procedure for assessing applicability and conformance	57
Annex D (informative) Activity limitation issues	68
Annex E (informative) Access features	74
Annex F (informative) Accessibility and usability	83
Bibliography	85

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 9241-171 was prepared by Technical Committee ISO/TC 159, *Ergonomics*, Subcommittee SC 4, *Ergonomics of human-system interaction*.

This first edition of ISO 9241-171 cancels and replaces ISO/TS 16071:2003, 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 3: Visual display requirements*
- *Part 4: Keyboard requirements*
- *Part 5: Workstation layout and postural requirements*
- *Part 6: Guidance on the work environment*
- *Part 7: Requirements for display with reflections*
- *Part 8: Requirements for displayed colours*
- *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*

Guidance on software individualization and human-centred design process for interactive systems are to form the subjects of future parts 129 and 210.

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 110: Dialogue principles*

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

— *Part 171: Guidance on software accessibility*

— *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*

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

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

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

Framework for tactile and haptic interaction is to form the subject of a future part 910.

Introduction

The purpose of this part of ISO 9241 is to provide guidance on the design of the software of interactive systems so that those systems achieve as high a level of accessibility as possible. Designing human-system interactions to increase accessibility promotes increased effectiveness, efficiency and satisfaction for people having a wide variety of capabilities and preferences. Accessibility is therefore strongly related to the concept of usability (see ISO 9241-11).

The most important approaches to increasing the accessibility of a human-system interface are

- adopting a human-centred approach to design (see ISO 13407),
- following a context-based design process,
- providing the capacity for individualization (see ISO 9241-110), and
- offering individualized user instruction and training.

It is important to incorporate accessibility goals and features into the design as early as possible, when it is relatively inexpensive compared to the cost of modifying products to make them accessible once they have been designed. As well as providing guidance for achieving that, this part of ISO 9241 addresses the increasing need to consider social and legislative demands for ensuring accessibility by the removal of barriers that prevent people from participating in life activities such as the use of environments, services, products and information.

This part of ISO 9241 is applicable to software that forms part of interactive systems used in the home, in leisure activities, in public situations and at work. Requirements and/or recommendations are provided for system design, appearance and behaviour, as well as specific accessibility issues, thereby complementing International Standards ISO 9241-11, ISO 9241-12, ISO 9241-13, ISO 9241-14, ISO 9241-15, ISO 9241-16 and ISO 9241-17, ISO 9241-110 and ISO 14915, as well as reflecting the goals outlined in ISO Guide 71 [60]. Conforming with the aforementioned International Standard is also important if the goal of accessibility is to be achieved.

NOTE 1 While the requirements and recommendations of this part of ISO 9241 are generally applicable to all software application domains, additional detailed guidance on the accessibility of Web content (including Web applications) is available from the Web Content Accessibility Guidelines (WCAG) [53].

This part of ISO 9241 is based on the current understanding of the characteristics of individuals who have particular physical, sensory and/or cognitive impairments. However, accessibility is an issue that affects many groups of people. The intended users of interactive systems are consumers or professionals — people at home, at school, engineers, clerks, salespersons, Web designers, etc. The individuals in such target groups vary significantly as regards physical, sensory and cognitive abilities and each target group will include people with different abilities. Thus, people with disabilities do not form a specific group that can be separated out and then disregarded. The differences in capabilities can arise from a variety of factors that serve to limit the capability to engage in the activities of daily living, and are a “universal human experience” [50]. Therefore, accessibility addresses a widely defined group of users including

- people with physical, sensory and cognitive impairments present at birth or acquired during life,
- elderly people who can benefit from new products and services but who experience reduced physical, sensory and cognitive capacities,
- people with temporary disabilities, such as a person with a broken arm or someone who has forgotten his/her glasses, and
- people who experience difficulties in particular situations, such as a person who works in a noisy environment or has both hands occupied by other activities.

When designing and evaluating interactive systems there are other terms that are often associated with accessibility. In Europe, the expression *design for all* or, in North America, *universal design* [9], address the goal of enabling maximum access to the maximum number and diversity of users, irrespective of their skill level, language, culture, environment or disability. This does not mean that every product will be usable by every consumer. There will always be a minority of people with severe or multiple disabilities who will need adaptations or specialized products. Accessibility as defined in this part of ISO 9241 emphasizes the goals of maximizing the number of users and striving to increase the level of usability that these users experience.

This part of ISO 9241 recognizes that some users of software will need assistive technologies in order to use a system. In the concept of designing software to be accessible, this includes the capability of a system to provide connections to, and enable successful integration with, assistive technologies, in order to increase the number of people who will be able to use the interactive system. Guidance is provided on designing software that integrates as effectively as possible with common assistive technologies. It is important to note that accessibility can be provided by a combination of both software and hardware controlled by software. Assistive technologies typically provide specialized input and output capabilities not provided by the system. Software examples include on-screen keyboards that replace physical keyboards, screen-magnification software that allows users to view their screens at various levels of magnification, and screen-reading software that allows blind users to navigate through applications, determine the state of controls, and read text via text-to-speech conversion. Hardware examples include head-mounted pointing devices instead of mice and Braille output devices instead of a video display. There are many others. When users employ add-on assistive software and/or hardware, usability is enhanced to the extent that systems and applications integrate with those technologies. For this reason, platforms (including operating systems) must provide programming services to allow software to operate effectively with add-on assistive software and hardware as specified in this part of ISO 9241. If systems do not provide support for assistive technologies, the probability increases that users will encounter problems with compatibility, performance and usability.

This part of ISO 9241 serves the following types of users:

- designers of user-interface development tools and style guides to be used by interface designers;
- user-interface designers, who will apply the guidance during the development process;
- developers, who will apply the guidance during the design and implementation of system functionality;
- those responsible for implementing solutions to meet end-user needs;
- buyers, who will reference this part of ISO 9241 during product procurement;
- evaluators, who are responsible for ensuring that products are in accordance with this part of ISO 9241.

NOTE 2 In this document the term “developers” is used as shorthand for *all those involved in the development of software design and creation*, which sometimes can span different collaborating or contracting organizations.

The ultimate beneficiary of this part of ISO 9241 will be the end-user of the software. Although it is unlikely that end-users will read this part of ISO 9241, its application by designers, developers, buyers and evaluators ought to provide user interfaces that are more accessible. This part of ISO 9241 concerns the development of software for user interfaces. However, those involved in designing the hardware aspects of user interfaces may also find it useful when considering the interactions between software and hardware aspects.

ISO 9241 was originally developed as a seventeen-part International Standard on the ergonomics requirements for office work with visual display terminals. As part of the standards review process, a major restructuring of ISO 9241 was agreed to broaden its scope, to incorporate other relevant standards and to make it more usable. The general title of the revised ISO 9241, “Ergonomics of human-system interaction”, reflects these changes and aligns the standard with the overall title and scope of Technical Committee ISO/TC 159, SC 4. The revised multipart standard is structured as series of standards numbered in the “hundreds”: the 100 series deals with software interfaces, the 200 series with human-centred design, the 300 series with visual displays, the 400 series with physical input devices, and so on.

See Annex A for an overview of the entire ISO 9241 series.

Ergonomics of human-system interaction —

Part 171: Guidance on software accessibility

1 Scope

This part of ISO 9241 provides ergonomics guidance and specifications for the design of accessible software for use at work, in the home, in education and in public places. It covers issues associated with designing accessible software for people with the widest range of physical, sensory and cognitive abilities, including those who are temporarily disabled, and the elderly. It addresses software considerations for accessibility that complement general design for usability as addressed by ISO 9241-110, ISO 9241-11 to ISO 9241-17, ISO 14915 and ISO 13407.

This part of ISO 9241 is applicable to the accessibility of interactive systems. It addresses a wide range of software (e.g. office, Web, learning support and library systems).

It promotes the increased usability of systems for a wider range of users. While it does not cover the behaviour of, or requirements for, assistive technologies (including assistive software), it does address the use of assistive technologies as an integrated component of interactive systems.

It is intended for use by those responsible for the specification, design, development, evaluation and procurement of software platforms and software applications.

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 9241-11:1998, *Ergonomic requirements for office work with visual display terminals (VDTs) — Part 11: Guidance on usability*

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

ISO 9241-13:1998, *Ergonomic requirements for office work with visual display terminals (VDTs) — Part 13: User guidance*

ISO 9241-14:1997, *Ergonomic requirements for office work with visual display terminals (VDTs) — Part 14: Menu dialogues*

ISO 9241-15:1997, *Ergonomic requirements for office work with visual display terminals (VDTs) — Part 15: Command dialogues*

ISO 9241-16:1999, *Ergonomic requirements for office work with visual display terminals (VDTs) — Part 16: Direct manipulation dialogues*

ISO 9241-17:1998, *Ergonomic requirements for office work with visual display terminals (VDTs) — Part 17: Form filling dialogues*

ISO 9241-110:2006, *Ergonomics of human-system interaction — Part 110: Dialogue principles*

ISO 13407:1999, *Human-centred design processes for interactive systems*

ISO 14915 (all parts), *Software ergonomics for multimedia user interfaces*

3 Terms and definitions

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

3.1
accelerator keys
shortcut keys
key combinations which invoke a menu option without displaying the menu on which the option appears or intermediate menus

[ISO 9241-14:1997]

3.2
accessibility
<interactive system> usability of a product, service, environment or facility by people with the widest range of capabilities

NOTE 1 The concept of accessibility addresses the full range of user capabilities and is not limited to users who are formally recognized as having a disability.

NOTE 2 The usability-orientated concept of accessibility aims to achieve levels of effectiveness, efficiency and satisfaction that are as high as possible considering the specified context of use, while paying particular attention to the full range of capabilities within the user population.

3.3
accessibility feature
feature (etc.) that is specifically designed to increase the usability of products for those experiencing disabilities

3.4
activation
internal state with differential degrees of mental and physical functional efficiency

[ISO 10075:1991]

3.5
assistive technology
AT
hardware or software added to, or incorporated within, a system that increases accessibility for an individual

EXAMPLE Braille display, screen reader, screen magnification software, eye tracking devices.

3.6
chorded key-press
keyboard key or pointing-device button presses where more than one button is held down simultaneously to invoke an action

NOTE This includes both uses of modifier keys with other (non-modifier) keys as well as use of multiple non-modifier keys to enter data or invoke an action.

3.7
closed system
system that does not allow user connection or installation of assistive technology that would have programmatic access to the full user interface

NOTE This can be because of policy, system architecture, physical constraints or for any number of other reasons.

3.8

colour scheme

set of colour assignments used for rendering user-interface elements

NOTE “Colour” refers to a combination of hue, saturation, and brightness.

3.9

contrast

⟨perceptual sense⟩ assessment of the difference in appearance of two or more parts of a field seen simultaneously or successively (hence: brightness contrast, lightness contrast, colour contrast, etc.)

[CIE 17.4:1987, definition 845-02-47]

3.10

cursor

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

cf. **keyboard focus cursor** (3.22), **text cursor** (3.35), **pointer** (3.30)

3.11

effectiveness

accuracy and completeness with which users achieve specified goals

[ISO 9241-11:1998, 3.2]

3.12

efficiency

resources expended in relation to the accuracy and completeness with which users achieve goals

[ISO 9241-11:1998, 3.3]

3.13

explicit designator

code or abbreviation for a menu option or control label, set apart from the name (usually to the left of it), and typed in for selection

cf. **implicit designator** (3.16)

EXAMPLE “O”, “C”, “S”, “P”, as shown in the menu in Figure 1.

O	Open
C	Close
S	Save
P	Print

Figure 1 — Examples of explicit designators

3.14

focus cursor

location cursor

indicator showing which user-interface element has keyboard focus

cf. **input focus** (3.18) and **cursor** (3.10).

EXAMPLE Box or highlighted area around a text field, button, list or menu option.

NOTE The appearance of this indicator usually depends on the kind of user-interface element that has focus. The user-interface element with focus can be activated if it is a control (e.g. button, menu item) or selected if it is a selectable user-interface element (e.g. icon, list item).

3.15

icon

graphic displayed on the screen of a visual display that represents a function of the computer system

[ISO/IEC 11581-1:2000, 4.7]

3.16

implicit designator

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

EXAMPLE “P” on a screen used for initiating a print job where the control label is displayed as “Print”.

3.17

individualization

modification of interaction and presentation of information to suit individual capabilities and needs of users

3.18

input focus

in relation to a given input device, the indication of the object upon which the user directs input

[ISO 9241-16:1999]

EXAMPLE Pointer focus and keyboard focus are input foci.

3.19

keyboard emulator

software or hardware that generates input identical to that from a keyboard

NOTE A keyboard emulator can provide a representation of keys (e.g. on-screen keyboard) or not (e.g. voice recognition).

EXAMPLE Platform-based on-screen keyboards, speech input, handwriting, wherever their output appears to applications as keystroke input.

3.20

keyboard equivalent

key or key combination that provides access to a function usually activated by a pointing device, voice input or other input or control mechanism

3.21

keyboard focus

current assignment of the input from the keyboard or equivalent to a user-interface element

NOTE For an individual user-interface element, focus is indicated by a focus cursor.

3.22

keyboard focus cursor

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

cf. **keyboard focus** (3.21), **pointer** (3.30), **text cursor** (3.35).

3.23

label

short descriptive title for an entry or read-only field, table, control or object

cf. **name** (3.27)

EXAMPLE 1 Heading, prompt for entry field, text or graphic that accompanies and identifies a control (such as those displayed on the face of buttons), audible prompt used by an interactive voice response system.

EXAMPLE 2 “Creation Time”, as shown in Figure 2.

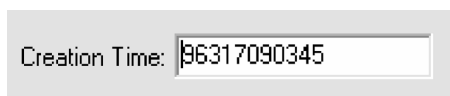


Figure 2 — Example of text field with label

EXAMPLE 3 “Pagination”, “Widow/Orphan control”, “Keep with text”, “Keep lines together”, “Page break before”, as shown in Figure 3.

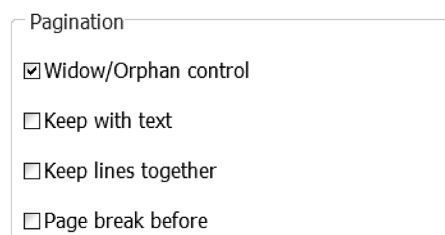


Figure 3 — Example of check box group with labels for group and each check box

EXAMPLE 4 Image of a printer in a window that the user can click to print the current document.

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

NOTE 2 Adapted from ISO 9241-17:1998, definition 3.4.

NOTE 3 For the purposes of this part of ISO 9241, *label* refers to the presented title for a user-interface element — in contrast with the *name* attribute, which might or might not be presented to users but is available to assistive technologies. Textual labels are often a visual display of the name.

3.24

latch

mode in which any modifier key remains logically pressed (active) in combination with a single subsequent non-modifier key-press or pointing-device button action

cf. **lock** (3.25)

3.25

lock

persistent mode in which one or more modifier keys or pointing-device buttons remain logically pressed (active) until lock mode for the key or button is turned off

cf. **latch** (3.24)

NOTE 1 Unlike *latch*, which affects only keyboard and pointing device actions, *lock* will affect any software that uses the modifier key(s) to alter its behaviour.

NOTE 2 Lock mode is usually turned off explicitly by the user, but can also be turned off at other times such as at system shutdown or restart.

3.26

modifier key

keyboard key that changes the action or effect of another key or a pointing device

EXAMPLE 1 Moving the keyboard focus with the shift key held down, thereby extending the current selection in the direction of cursor movement rather than merely moving the position of the cursor.

EXAMPLE 2 Pressing “C”, to obtain the input of that character, pressing “Ctrl+C” to obtain the “Copy” function.

3.27

name

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

cf. **label** (3.23).

NOTE 1 Names are most useful when they are the primary word or phrase by which the on-screen instructions, software documentation and the user refer to the element, and when they do not contain the type or status of the user-interface element.

NOTE 2 The name attribute might or might not be presented to users but is available to assistive technologies — in contrast with *label*. For the purposes of this part of ISO 9241, *label* refers to the presented title for a user-interface element. Textual labels are often a visual display of the name.

NOTE 3 When a textual label is provided it would generally present the name or a shortened version of the name. Not all user-interface elements have labels, however. In those cases, the names would be available to assistive technologies (or sometimes by pop-up tool tips, etc.).

NOTE 4 Names are not to be confused with internal identifiers (ID), which can be used by software and might not be designed to be understood by a human.

3.28

natural language

language that is, or that was, in active use in a community of people, the rules of which are mainly deduced from the usage

3.29

platform software

software that interacts with hardware or provides services for other software

EXAMPLE Operating system, device driver, windowing system, software toolkit.

NOTE 1 A browser can function both as an application and as platform software.

NOTE 2 For the purposes of this part of ISO 9241, *software* refers to both platform software and application software.

3.30

pointer

graphical symbol that is moved on the screen according to operations with a pointing device

NOTE Users can interact with elements displayed on the screen by moving the pointer to that location and starting a direct manipulation.

[ISO 9241-16:1999, 3.15]

3.31

pointer focus

current assignment of the input from the pointing device to a window

NOTE The window with pointer focus usually has some distinguishing characteristic, such as a highlighted border and/or title bar.

3.32

pointing device

device that translates a human controlling operation to a controlling operation on the display

NOTE 1 Depending on the applied technology, not only machine devices but also parts of the human body (e.g. fingers, arms) can currently be used as pointing devices.

[ISO 9241-16:1999]

NOTE 2 Pointing devices typically have buttons that are used to activate or manipulate user-interface elements.

NOTE 3 Almost any hardware can be used to control a **pointer** (3.30) with the appropriate software.

3.33

satisfaction

freedom from discomfort, and positive attitudes towards the use of the product

[ISO 9241-11:1998, 3.4]

3.34

screen reader

assistive technology that allows users to operate software without the need to view the visual display

NOTE 1 Output of screen readers is typically text-to-speech or dynamic Braille output on a refreshable Braille display.

NOTE 2 Screen readers rely on the availability of information from the operating system and applications, such as the name or label of the user-interface element.

3.35

text cursor

visual indication of the current insertion point for text entry

cf. **pointer** (3.30), **focus cursor** (3.14).

3.36

usability

extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use

[ISO 9241-11:1998, 3.1]

3.37

user interface

UI

all components of an interactive system (software or hardware) that provide information and controls for the user to accomplish specific tasks with the interactive system

[ISO 9241-110:2006, 3.9]

3.38

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.

4 Rationale and benefits of implementing accessibility

Accessibility is an important consideration in the design of products, systems, environments and facilities because it affects the range of people who are able to use them and use them easily. The more accessible a design, the wider the range of people who will find it usable.

Accessibility can be improved by incorporating features and attributes known to benefit users with special requirements. To determine the achieved level of accessibility, it is necessary to measure the effectiveness, efficiency and satisfaction of users working with a product or interacting with an environment for the widest range of users. Measurement of accessibility is particularly important in view of the complexity of the interactions with the user, the goals, the task characteristics and the other elements of the context of use. A product, system, environment or facility can have significantly different levels of accessibility when used in different contexts.

Planning for accessibility as an integral part of the design and development process involves the systematic identification of requirements for accessibility, including accessibility measurements and verification criteria within the context of use. These provide design targets that can form the basis for verification of the resulting design.

The approach adopted in this part of ISO 9241 has the following benefits:

- the framework can be used to identify the aspects of accessibility and the components of the context of use to be taken into account when specifying, designing or evaluating the accessibility of a product;
- the performance and satisfaction of the users can be used to measure the extent to which a product, system, environment or facility is accessible in a specific context;
- measures of the performance and satisfaction of the users can provide a basis for determining and comparing the accessibility of products having different technical characteristics, which are used in the same context;
- the accessibility planned for a product can be defined, documented and verified (e.g. as part of a quality plan).

5 Principles for designing accessible software

There are different ways of designing accessible software. This part of ISO 9241 does not assume any one specific design method or process, nor does it cover all the different activities necessary to ensure accessible system design. It is complementary to existing design methods and provides a human-centred accessibility perspective, based on ISO 13407, that can be applied — whatever the specific design process or particular context of use — to increase the number of people who are able to use the software. The guidance provided in this part of ISO 9241 is applicable at any stage in the development of an interactive system.

Designing accessible software should adhere to the following principles.

— Equitable use

Equitable solutions provide the same means of use for all users: identical whenever possible; equivalent when not. Achieving equitable use will ensure that solutions designed to increase accessibility do not result in such things as loss of privacy, increased risks to personal safety or security, or the stigmatization of individuals.

— **Suitability for the widest range of use**

Suitability for the widest range of use involves designing with the objective of producing solutions that will be useful, acceptable and available to the widest range of users within the intended user population, taking account of their special abilities, variations in their capabilities, the diversity of their tasks, and their differing environmental, economic and social circumstances.

— **Robustness** (WCAG 2.0 Principle No. 4) ^[53]

Software should be designed to be as *robust* as possible to allow it to work with current and future assistive technologies. Although it is not feasible to make all software accessible without add-on assistive technologies, these guidelines should help designers develop software that increases accessibility without the use of assistive technologies, and, by providing the necessary interface information, enables assistive software and devices to operate effectively and efficiently when used. The software can promote integration of assistive technologies by providing information that can be read by assistive technologies, and by communicating through standard application-to-application communication protocols.

EXAMPLE Systems that provide built-in screen magnification can enable many more users to read the text and see the images that are presented. However, if the necessary integration information is available, users can attach the screen magnification program of their choice to suit their particular needs.

The development of solutions that will result in accessible software should be based upon the application of ergonomic design guidance: ISO 9241-12, ISO 9241-13, ISO 9241-14, ISO 9241-15, ISO 9241-16, ISO 9241-17 and ISO 9241-110, and ISO 14915-1, ISO 14915-2 and ISO 14915-3, shall be consulted. The guidance in these International Standards is also included in the provisions of other standards and publications such as the Web Content Accessibility Guidelines 2.0 (WCAG) ^[53], universal design principles/design for all (DFA) ^[9], which specifically aim to increase accessibility. Issues that represent ergonomics principles and good practice that are particularly important as the basis for achieving accessibility are that

- information should be perceivable by the user (ISO 9241-12, WCAG 2.0 Principle No 1);
- content and control should be understandable (ISO 9241-12, ISO 9241-110, WCAG 2.0 Principle No 3);
- interface elements should be operable (ISO 9241-110, WCAG 2.0 Principle No 2);
- software should be error tolerant (ISO 9241-110, DFA);
- software should be flexible in use, enabling users to choose from a wider range of input and output alternatives. (ISO 9241-110, DFA).

The additional guidance provided in this part of ISO 9241 specifically addresses issues that arise when providing design solutions that will satisfy the needs of users with a great variety of capabilities.

6 Sources of variation in user characteristics

All user populations vary significantly in terms of their characteristics, capabilities and preferences. Any interactive system will include, within the user group for which it is designed, people with very different physical, sensory and cognitive abilities. These differences will have many different sources including innate characteristics, culture, experience and learning, as well as changes that occur throughout life. While the requirements and recommendations in this part of ISO 9241 are based on the current understanding of the individual characteristics of people who have particular physical, sensory and cognitive impairments, their application addresses the diversity of abilities within any intended user population that may lead to limitation on activities.

Disabilities considered include not only those due to restrictions on mobility or physical performance, such as loss of a limb or tremor, but also those associated with sensory impairment, such as low vision or hearing loss, as well as cognitive factors, such as declining short-term memory or dyslexia. Some disabilities can be

experienced intermittently and appear unpredictably or very suddenly. In addition, a disability can be experienced as an entirely unique event for an individual (e.g. having a broken arm). Annex D provides an outline of some of the limitations typically encountered by individuals with various types of disability, but does not constitute an exhaustive account of all the issues that can arise. In addition, limitations on activities can be created by the physical environment (e.g. poor lighting and noise), social environment (e.g. confidential tasks carried out in the presence of other people) or by the need to deal with other tasks in parallel, and these are also taken into account.

The extent to which the particular source of any disability creates limitations varies and some of the guidance provided is specific to the degree of disability experienced. Thus, impairments in vision can range from declining ability to resolve small detail to having been blind from birth. Different provisions in the design of the interactive system could be needed to deal with different degrees of disability: for example, the facility to enlarge the size of detail presented on a screen will not address the problems of those people who are blind.

It is also important to recognize that people can experience multiple disabilities. Guidance that is appropriate to addressing a specific type of disability might not work if somebody who has that disability also has some other type of disability. For example, auditory output of written text will not provide support for people who are deaf-blind. Combinations of different disabilities and variations in the levels of disabilities experienced can have unexpected effects. It is therefore important that different approaches to access be supported so that interfaces can be individualized to the user and their task.

7 How to use this part of ISO 9241

7.1 General

In order to achieve accessibility, it is necessary to provide support in different parts of the system of software, which includes platform software (the operating system and associated layers, and toolkits) and other software (such as most applications) that run on and make use of services provided by platform software.

While much can be done to improve accessibility in the design of an application, it is not possible to provide all of the input and output support that users require in every circumstance at the application level alone. To the extent that any particular part of the software is dependent upon a level below it for its operational characteristics, it will be necessary to ensure that the lower levels enable the implementation of recommended accessibility characteristics in any layers that depend upon them. Similarly, accessibility characteristics implemented by the platform can require cooperation from layers running on top of them in order to be fully effective. The majority of the requirements and recommendations given in Clauses 8, 9, 10 and 11 demand that the issue be addressed at more than one level of the software system if the particular requirement or recommendation is to be satisfied.

These dependencies can occur in relation to different layers in the platform (e.g. window management on top of process management and screen drawing, which are on top of hardware drivers) and in relation to the applications that are mounted on the platform. Applications themselves can have layers that result in dependencies arising within different levels of the application.

Most of the provisions given in this part of ISO 9241 apply to all software that implements or contributes to the software user interface — regardless of whether or not it is part of the platform. Some are only applicable to portions of platform software (such as guidelines about low-level input, window management or system-wide behaviour): for example, the platform is the general means by which accessibility features that involve control of hardware devices, in particular those involving input, are implemented and controlled. Similarly, other provisions might only apply to software that displays user-interface elements, generates sounds, or exhibits other specific behaviours. In these cases, the applicable layers or type of software is indicated in the text of the requirement or recommendation, and in some cases are further elaborated upon in accompanying Notes.

7.2 Conformance

Conformance with this part of ISO 9241 is achieved by satisfying all the applicable requirements and by the provision of a systematic list of 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. For reference purposes, all the clauses including requirements are listed in Annex B.

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 (“may” statements give permission). If a product is claimed to have met the applicable recommendations in ISO 9241-171, the procedure used in establishing requirements for, developing and/or evaluating, the software accessibility shall be specified. The level of specification of the procedure is a matter of negotiation between the involved parties.

ISO 13407:1999, 7.2, and ISO 9241-11:1998, 5.3, shall be consulted for guidance on the identification and specification of the context of use; see Annex C for guidance on assessing applicability.

Annex C provides a means both for determining and recording the applicability of all the requirements and recommendations and for reporting that they have been followed. Other equivalent forms of report are acceptable.

Server software (used in client-server and mainframe environments) shall be evaluated in conjunction with the client (including terminal) software that would be used with it.

Software used on or intended to be used on closed systems shall be evaluated in conjunction with the intended hardware configuration and shall be in accordance with all requirements of Clauses 8, 9, 10 and 11, except for those given in 8.6.

8 General guidelines and requirements

8.1 Names and labels for user-interface elements

8.1.1 Provide a name for each user-interface element

Software shall associate an identifying name with every user-interface element except where the name would be redundant.

NOTE 1 A name conveys the identity of the user-interface element to the user. It complements the role attribute that tells an element’s function (such as that it acts as a push button) and the description attribute that summarizes the element’s visual appearance.

EXAMPLE 1 The application provides a label showing the name “File name” for a static text field that shows the name of the file being described in the fields below.

EXAMPLE 2 Dialog boxes or windows have meaningful names, so that a user who is hearing rather than seeing the screen gets appropriate contextual information.

NOTE 2 If some names are missing, assistive technology might be unable to sufficiently identify or operate the user-interface elements for the user.

NOTE 3 Names would be redundant for user-interface elements whose entire informational content is already conveyed by their role attribute (such as a horizontal rule), static text elements that serve to name other elements, and elements that only serve as an integral portion of a parent element (such as the rectangular border around a button).

EXAMPLE 3 The software does not need to provide a name for a static text field that says “Last name:” and serves to identify text box that follows it, as that string would be exposed using the field’s value attribute.

EXAMPLE 4 When a script or object hosted within a Web browser attempts to set the window’s title to a string that is already used by another of the browser’s windows, the browser modifies this string to be unique.

NOTE 4 In some cases, the name will be displayed visibly, but in other cases it will only be provided programmatically for use by assistive technology as described in 8.6.

EXAMPLE 5 A control is listed in the product documentation as “the Print button”; therefore, its identifying name in the software is “Print” (regardless of whether the word “Print” appears on the visual representation of the button or not).

NOTE 5 Such elements can be containers that serve to group one or more sub-elements. In a typical graphical user interface, examples of UI (user interface) elements include basic elements such as window title bars, menu items, push buttons, images, text labels and editable text fields; while examples of containers include windows, list boxes, grouping boxes, menu bars, menus, groups of mutually-exclusive option buttons, and compound images that are made up of several smaller images. In a typical audio user interface, examples of interactive UI elements include menus, menu items, messages, prompt tones and pauses.

EXAMPLE 6 Compound user-interface elements that consist of a collection of other user-interface elements have a group name. A Web page image composed of a series of smaller image files provides a group name (“Construction site”) in addition to the names of the individual component images (“building”, “bulldozer”, “dump truck”, “crane”, etc.).

8.1.2 Provide meaningful names

Names of user-interface elements should be comprised of natural language words that are meaningful to the intended users.

NOTE 1 This means that each word in the name would occur in a standard dictionary or in electronic documentation for end-users included with the software.

NOTE 2 The names are most useful when they are the primary name by which the software, its documentation and its users refer to the element, and do not contain the type or status of the user-interface element.

EXAMPLE 1 The name of a checkbox is “Gender” and not “Gender checkbox.”

NOTE 3 User-interface elements that represent a real entity (such as a document, location or person) can be provided with the name of that entity even if the name is too lengthy or cryptic to be read easily.

NOTE 4 Names can use terms that are specific to a particular task domain, provided that they have established meanings for the intended users.

EXAMPLE 2 A control is listed in the product documentation as “the Print button”; therefore its identifying name in the software is “Print” (regardless of whether the word “Print” appears on the visual representation of the button or not).

EXAMPLE 3 Dialog boxes or windows have meaningful names, so that a user who is using speech output because he/she cannot see the screen gets appropriate contextual information.

8.1.3 Provide unique names within context

Each name of a user-interface element specified by software developers should be unique within its context.

NOTE 1 Users will not be able to use the name to identify an element if several elements have the same name within the same context.

NOTE 2 A name is considered unique if no other user-interface element with the same name and role attributes shares the same container or parent element (such as a window, group box, section, etc.).

NOTE 3 User-interface elements that represent a real entity (such as a document, location or person) can be provided with the name of that entity even if the name is not unique in its context.

EXAMPLE 1 A form displays areas containing fields with a customer’s home and business details. Each area has an associated “Change” button. Rather than duplicate the name, the buttons are named “Change Home” and “Change Business”.

EXAMPLE 2 A form for a purchasing application has several rows of items, each containing a text field displaying the title of a book, followed by “Buy” button that is used to purchase that book. Even though the face of each button looks identical, the form is implemented so that each provides a unique name for use by assistive technology, such as “Buy *The Grapes of Wrath*” and “Buy *Pride and Prejudice*”.

EXAMPLE 3 A user opens a second window of the same document using a word processing application. Both windows are of the same document and both are editable. The word processor adds a “:1” to the end of the document name to form the name of the first window. It names the second window with the same document name, except that it appends a “:2” to the end of the second window name so that the two windows have unique names.

EXAMPLE 4 When a script or object hosted within a Web browser attempts to set the window's title to a string that is already used by another of the browser's windows, the browser modifies this string to be unique.

8.1.4 Make names available to assistive technology (AT)

Each name of a user-interface element and its association shall be made available by the software system to assistive technology in a documented and stable fashion.

NOTE 1 In a platform that does not provide a standard service for the association of names and elements, application developers document how assistive technologies can access that information.

NOTE 2 Services for the association of names and elements are part of those described in 8.1.6.

8.1.5 Display names

If a user-interface element has a visual representation and is not part of standard components of the user interface, software should present its name to users (either by default or at the user's request).

NOTE Standard components of the user interface are components that are provided by the platform and that look and behave the same across applications.

EXAMPLE 1 In an application, the window scroll up and scroll down buttons do not have a label or pop-up text because these buttons are standard across applications on the platform. At the bottom of the scrollbar, however, are special “jump to next find” and “jump to last find” arrows that do have pop-up text that describes their function.

EXAMPLE 2 A print button has a picture of a printer with a textual name that pops up when the user pauses a pointer over the button and also when the user moves the keyboard focus to the button and presses a specific keyboard command.

8.1.6 Provide names and labels that are short

Each name or label of a user-interface element specified by the software developers should be short enough to be rendered succinctly.

NOTE 1 If developers are encouraged to put the most distinctive parts of the name first, users can skip over the latter parts once they have read enough to identify the element.

NOTE 2 User-interface elements that represent a real entity (such as a document, location or person) can be provided with the name of that entity even if the name is too lengthy or cryptic to be read easily.

NOTE 3 Using brief labels also benefits users of auditory, visual and tactile output.

EXAMPLE 1 “Print” is used instead of “Print button” or “This button prints the current document.”

EXAMPLE 2 An icon representing a document is labelled with the file name or title of the document, even if that string is too lengthy or cryptic to be read easily, because that string was determined by the document author rather than by software developers.

8.1.7 Provide text label display option for icons

Software should allow users to choose to display either the icon images, the icon images with text labels or the icon text labels only.

NOTE It is useful for the user to be able to adjust the font size (see also 10.3.1).

8.1.8 Properly position the labels of user-interface elements on screen

The labels for user-interface elements provided by software should be consistently positioned, relative to the elements that they are labelling, on the display (see also ISO 9241-12:1998, 5.9.4 and 5.9.6).

If the platform software has conventions for positioning labels relative to the elements that they are labelling, these conventions should be followed.

NOTE This helps AT correctly associate the labels with their corresponding controls, and helps users of screen magnification software know where to look for a label or control.

8.2 User preference settings

8.2.1 Enable individualization of user-preference settings

When the software enables the user to set personal preferences, these settings should be easily adjustable.

EXAMPLE 1 A software application allows users to configure and save settings for font size and style within a particular window.

NOTE 1 System-wide user preference settings provided by the platform need to be used in addition to any preference settings for product-specific options.

EXAMPLE 2 A software application allows a user with cognitive disabilities to choose the number and size of icons displayed at any one time.

NOTE 2 Requiring users to hand edit a configuration file is not an easy method for individualizing preference settings because it is too easy for the user to accidentally enter invalid values or otherwise corrupt the file.

EXAMPLE 3 A user chooses preference settings through a graphical user interface, rather than directly editing the configuration files.

NOTE 3 Business considerations related to consistency of operations, performance-based considerations, safety, privacy, and security concerns can all lead to some necessary restriction by system administrators of the user's capability to modify the behaviour and appearance of user-interface elements in certain contexts. Administrators need to show restraint in limiting user control. Not all options/preferences settings are appropriate for such administrative control.

NOTE 4 Administrators within this business environment can make specific permission profiles for users that require more flexibility in their options/preference settings for usability and accessibility.

8.2.2 Enable adjustment of attributes of common user-interface elements

Software should enable users to adjust the attributes of common user-interface elements if applicable to the task.

NOTE 1 Common attributes for a visual interface could include, but are not limited to, font type, font size, and font colour. For an auditory interface they could include, but are not limited to, aural cue type, rate, volume, pitch, position in 3D audio space. For a tactile interface they could include, but are not limited to, haptic object size, texture, xy- or xyz-position, pressure sensitivity, solidity.

NOTE 2 Platform software often supports these options for standard user-interface elements it provides. To enhance user experience, applications can use the settings defined at the platform level.

EXAMPLE Software retains user preference for window size and location between sessions.

8.2.3 Enable individualization of the user interface look and feel

Software should provide a mechanism enabling users to individualize the interface "look and feel", including the modification or hiding of command buttons.

EXAMPLE 1 A user with a cognitive disability can, when using a given application, change the interface to simplify the application's look and feel.

EXAMPLE 2 A word processor allows users to temporarily hide menu items and tool bar buttons that they do not find useful for a given situation.

8.2.4 Enable individualization of the cursor and pointer

If the hardware supports the service, software shall enable users to individualize attributes of all keyboard focus cursors, text cursors and pointers, including, but not limited to, shape, size, stroke width, colour, blink rate (if any), and pointer trails (if any).

NOTE 1 Platform software often supports these options for standard cursors and pointers it provides, and software using these cursors and pointers can automatically comply with this requirement.

NOTE 2 The ability to set the cursor to non-blinking is important for users with attention deficits, who may be easily distracted.

NOTE 3 The colour aspect of this provision is not applicable if the presentation of the cursor or pointer is an inversion of the image and it has no colour.

EXAMPLE 1 Users with low vision can change a text cursor from non-blinking to blinking, and adjust the size to be more readily visible given their visual capabilities.

EXAMPLE 2 Users with low vision and/or a colour deficiency can change the thickness and colour of the keyboard focus cursor so that they can more easily see the current input focus.

EXAMPLE 3 Users with low vision can make the pointer larger so that they can more readily locate it.

8.2.5 Provide user-preference profiles

Software should enable users to create, save, edit and recall profiles of preference settings, including input and output characteristics, without having to carry out any restart that would cause a change of state or data.

NOTE 1 For systems that provide access for multiple users, such as library systems, conversion back to a default profile can be advisable.

NOTE 2 It is often useful to be able to access the preference settings over a network. Doing this in a secure way would preserve privacy especially for people who are worried about revealing the fact that they have a disability.

NOTE 3 It is preferable to minimize the need to restart the system or application in order for changes in user interface settings to become effective.

EXAMPLE 1 Platform software allows each user to save global settings for font size, sound volume, and pointer-control settings that apply everywhere on the system.

EXAMPLE 2 A software application allows users to configure and save settings for font size and style within a particular window.

EXAMPLE 3 The profile for a public library system is modified for the needs of a current user but returns to default values when that user is finished.

EXAMPLE 4 For a person completing an on-line process who has to make adjustments to the accessibility feature to reduce errors, restarting the operating system or the user agent would cause loss of work.

8.2.6 Provide capability to use preference settings across locations

Software should permit users to transfer their preference settings easily onto a compatible system.

NOTE 1 Portability is important for users with disabilities because they could find a system difficult or impossible to use without preferences set to meet their interaction needs. The overhead and effort required to create preference settings can be a significant hindrance to system usability if it must be repeated at every location.

NOTE 2 User preferences profiles are sometimes made publicly available, e.g. for download from the Internet. Because people can be concerned about others knowing of their disability, it would be helpful if their use of these resources could be kept private.

NOTE 3 Some platform software can provide a general mechanism for transferring their preference settings; in such cases, software might not have to implement this feature itself, as long as it follows platform conventions for storing its user preference settings.

EXAMPLE 1 A user visiting a different building on the company network logs in and the system automatically locates and uses his or her personal preference settings from the network without having to edit configuration files.

EXAMPLE 2 A user loads a preference settings file from a USB (universal serial bus) drive onto a new computer.

EXAMPLE 3 A user's preference settings are loaded from a smart card onto a new system.

8.2.7 Enable user control of timed responses

Unless limits placed on the timing of user responses are essential to maintaining the integrity of the task or activity or are based on real life time constraints (e.g. an auction), software shall allow users to adjust each software-specified user response time parameter in one or more of the following ways:

- the user is allowed to deactivate the time-out;
- the user is allowed to adjust the time-out over a wide range which is at least ten times the length of the default setting;
- the user is warned before time expires, allowed to extend the time-out with a simple action (for example, "hit any key") and given at least 20 s to respond.

EXAMPLE A log-on prompt requires the user to enter their password within 30 s. The remaining time is shown on the screen and a control provided to stop the time decrementing.

8.3 Special considerations for accessibility adjustments

8.3.1 Make controls for accessibility features discoverable and operable

Software shall enable any "On"/"Off" controls and adjustments for accessibility features to be discoverable, and operable, by those who need that feature.

NOTE Features are considered *discoverable* if their settings and description can be found by browsing the user interface (including browsing "Help" available through the application).

EXAMPLE 1 Rather than needing to use a combination of keys, a user can turn the StickyKeys (see Annex E) accessibility feature on and off by pressing the Shift key five times in succession.

EXAMPLE 2 Accessibility features are turned on by use of a single toggle key held down at system start-up.

EXAMPLE 3 Controls for individualizing low-vision options are shown in large type by default.

8.3.2 Safeguard against inadvertent activation or deactivation of accessibility features

Software should prevent inadvertent activation or deactivation of accessibility features.

EXAMPLE The software system requests confirmation before activating or deactivating accessibility features.

8.3.3 Avoid interference with accessibility features

Software shall not disable or interfere with the accessibility features of the platform.

EXAMPLE Software that intercepts keyboard input does not defeat the operation of keyboard filters such as key latching and locking.

8.3.4 Inform user of accessibility feature “On”/“Off” status

Software should allow the user to identify the current status of accessibility features.

EXAMPLE 1 A control panel shows the current state of all accessibility features.

EXAMPLE 2 A small icon on the screen indicates that an accessibility feature is switched on.

8.3.5 Inform user of accessibility feature activation

When an accessibility feature can be activated unintentionally, software should inform users and provide an opportunity to accept or cancel the activation.

NOTE 1 The alert for an individual accessibility feature could be disabled at user request but would be “on” by default.

NOTE 2 It is good practice to provide an alert whenever the SlowKeys¹⁾ feature (see Annex E) is activated via a keyboard shortcut method, since an accidental activation could lead an uninformed user to believe that the keyboard was broken. Regular users of SlowKeys, however, prefer to have a way to defeat the alert so that they do not have to dismiss the alert each time they turn their SlowKeys feature on.

8.3.6 Enable persistent display

When users can activate a menu, control, or other user-interface element to display additional information or controls, software should allow that information or control to persist while the user engages in other tasks, until the user chooses to dismiss it [44], if it is appropriate to the task.

NOTE Persistent display of frequently used windows and controls may be helpful for users who have physical, language, learning or cognitive disabilities, and reduces the number of steps required to access them.

EXAMPLE 1 Users can keep a “Help” window available as they go through the tasks described.

EXAMPLE 2 The user can “tear off” one or more menus and continue to view and/or use them while navigating and using other menus.

EXAMPLE 3 The user can add a toolbar button that duplicates the function of a specific menu command. By doing so, this menu command is persistently displayed.

8.4 General control and operation guidelines

8.4.1 Enable switching of input/output alternatives

Platform software should enable users to switch among the available input/output alternatives without requiring them to reconfigure or restart the system or applications, unless there would be no change of state or data.

NOTE This capability aids users with different abilities who are working together on the same system.

1) SlowKeys™ is a trademark of the University of Wisconsin. This information is given for the convenience of users of this part of ISO 9241 and does not constitute an endorsement by ISO of the product named. Equivalent products may be used if they can be shown to lead to the same results.

EXAMPLE 1 A person who is blind uses his or her system only through the keyboard, using keyboard substitutes for mouse actions. A sighted user working on the same system can use the mouse and type in text. The system does not have to be restarted in between each session.

EXAMPLE 2 One user points with a mouse to an icon for a document on their screen and says “print” to print the document while another clicks on the document and keys in CTRL+P to print it. Still another uses the “Print” item in the “File” menu to print the document.

EXAMPLE 3 To implement a setting change, an application must restart, but it restores all data including position of the keyboard focus cursor.

8.4.2 Optimize the number of steps required for any task

Software should be designed to optimize the number of steps that the user has to perform for any given task.

NOTE It is important to find a balance between reducing the steps to improve efficiency and adding steps to allow for sufficient explanation of an infrequent task. ISO 13407 provides guidance on human centred design processes that can help determine the optimal design.

EXAMPLE 1 A user who wants to print one document can do so in only two steps. Once he/she selects the “Print” icon on the toolbar, a dialogue is displayed (which can, if desired, be used to change various print settings) and the user then simply selects the “OK” button.

EXAMPLE 2 A user with cerebral palsy types very slowly, and so finds it much more convenient when he/she can save a document by pressing a single key combination rather than having to navigate menus and dialog boxes.

8.4.3 Provide “Undo” and/or “Confirm” functionality

Software should provide a mechanism that enables users to undo at least the most recent user action and/or cancel the action during a confirmation step ^[44].

NOTE 1 Although this is a general ergonomic principle, “Undo” mechanisms are particularly important for users who have disabilities that significantly increase the likelihood of an unintentional action. These users can require significant time and effort to recover from such unintentional actions.

NOTE 2 A macro is considered to be one user action.

NOTE 3 Generally, the more consecutive actions the user can undo, the better.

NOTE 4 It is preferable that undo operations themselves can be undone.

NOTE 5 However, this might not be possible for such interactions as operations causing fundamental transformation of logical or physical devices, or could involve a data exchange with third parties that are out of the software’s control, etc.

NOTE 6 The default configuration can provide a confirmation step for any action that the user cannot undo with a single “Undo” command.

NOTE 7 Software could allow the user to disable the confirmation for specific actions.

EXAMPLE 1 A user with Parkinson’s disease might inadvertently input a sequence of keystrokes that activates several dialogues that then need to be undone. The use of several steps of the undo function permits the user to go back to the original state.

EXAMPLE 2 A user is about to format a hard disk. As this is an operation that cannot be undone, the software shows a confirmation dialog before the formatting begins.

8.4.4 Provide alternatives when assistive technology is not operable

If a task requires user interaction when the state of the software prevents the use of AT or speech output (such as during system start-up), the software shall provide the user with an alternative means for completing the task that does not require user interaction during that period.

NOTE System start-up and restart include operations prior to the stage where the user's accessibility aids and preference settings are available. They can be either accessible or non-interactive.

EXAMPLE 1 A computer is configured with a small "boot loader" that lets the user choose between two or more operating systems present on the system. Because the boot loader's menu is run before a full operating system or any assistive technology is running, it provides a mechanism by which the user can, during a normal session and using AT, specify which operating system will be loaded the next time the system starts.

EXAMPLE 2 A computer does not request any password or other user input until after it has loaded access features.

EXAMPLE 3 A public information kiosk restarts automatically and comes up accessible. There is no log-on before access features work.

8.4.5 Enable software-controlled media extraction

If the hardware allows for it, software shall enable the user to perform software-controlled media extraction.

NOTE 1 Those media include, but are not limited to, floppy disks, CD-ROM and DVD.

NOTE 2 In most cases, the user can use this feature as provided by the platform software without explicit support from software applications.

EXAMPLE A user who cannot press physical buttons on the computer or handle a CD-ROM can nevertheless use the on-screen keyboard to instruct the operating system to eject the disc, so that it will not interfere with the computer's next restart.

8.4.6 Support "Copy" and "Paste" operations

Software should support "Copy" and "Paste" operations for all user-interface elements that support text input.

NOTE 1 "Copy" and "Paste" operations enable users with disabilities to avoid awkward, slow and error-prone manual re-entry of potentially large amounts of data.

NOTE 2 "Copy" operations from password entry fields and similar secure objects can copy the text that is displayed on the screen rather than the actual text.

EXAMPLE As the user types into a password entry field, the field displays a dot to represent each character typed. The user can select this text and copy it to the clipboard, but the appropriate number of asterisk characters are copied rather than the typed password.

NOTE 3 "Cut" operations can also be provided as they enable faster user work.

8.4.7 Support "Copy" operations in non-editable text

Software should support "Copy" operations for all user-interface elements that display text.

NOTE The ability to copy non-editable text onto the clipboard can help users with disabilities avoid awkward, slow, or error-prone manual input of that text into other locations.

EXAMPLE 1 A user who communicates with the user assistance team by email provides examples of problems by copying the text from an error dialog box and pasting it into the email rather than having to retype it.

EXAMPLE 2 An operating system provides a function whereby the user can hold down the "Control" and "Alt" keys and select any text that was drawn to the screen using the OS (operating system) text drawing routines.

8.4.8 Enable selection of elements as an alternative to typing

Where the user can enter commands, file names, or similar choices from a limited set of options, software should provide at least one method for selecting or choosing that does not require the user to type the entire name.

NOTE 1 This reduces cognitive load for all users and reduces the amount of typing for users for whom spelling or typing is difficult, slow or painful.

EXAMPLE 1 An application prompts the user for a filename by presenting a dialog box in which the user can type in a filename or choose from a list of existing files.

EXAMPLE 2 At a command line prompt, the user can type the first letter or letters of a file name and then press TAB to complete the name. Repeatedly pressing TAB allows cycling through the names of additional files that match the string the user entered. This same mechanism can be used to enter command names as well as file names.

NOTE 2 In many cases, these features are supported automatically when software incorporates standard user-interface elements provided by the platform software.

8.4.9 Allow warning or error information to persist

Software shall ensure that error or warning information persists or repeats in a suitable manner as long as the source of the error or warning remains active, or until the user dismisses it [44], [49], [51].

EXAMPLE A dialog box indicating that a file was not saved remains visible until the user presses a “Close” button.

8.4.10 Present user notification using consistent techniques

Alerts, warnings, and other user notification should be presented by software using consistent techniques that enable a user to locate them and identify the category of the information (e.g. alerts versus error messages).

EXAMPLE 1 Following a “beep”, a positioning message is provided in a consistent place that allows the user having low vision to look for and find the error message more easily.

EXAMPLE 2 Every error message occurs in a dialog box, while every informative (non-error) message appears in the bottom left of a window. The consistent position of error messages allows users who are viewing only part of the screen through magnification to predict where particular types of information are likely to be found.

NOTE A screen reader can be programmed to read a message automatically as long as it appears in a consistent manner in a particular screen location.

EXAMPLE 3 Notification that a form field is mandatory or read-only is displayed on the status bar. On tabbing to the control, the message is automatically picked up by a screen reader that has been programmed to monitor this area of the window.

8.4.11 Provide understandable user notifications

Alerts, warnings and other user notifications provided by software should be short, simple and written in a clear language.

NOTE 1 Short messages do not preclude the provision of additional details on request.

NOTE 2 ISO 9241-13 provides detailed recommendations on user guidance.

EXAMPLE 1 Notifications are presented in the language of the user, avoiding internal system codes, abbreviations and developer-oriented terminology.

EXAMPLE 2 A message box appears, displaying the short meaningful message, “The network has become unavailable.” The user can choose the “OK” button to dismiss the message, or choose the “Details” button to see the more detailed message, “Error #527: thread 0xA725 has failed to yield mutex during maximum timeout period”, which might or might not be comprehensible to the user.

8.4.12 Facilitate navigation to the location of errors

When software detects that users have entered invalid data, it should notify them in a way that allows the users to identify and navigate easily to the location of the error.