TECHNICAL REPORT

ISO/IEC TR 14652

First edition
2004-09-15

Information technology — Specification method for cultural conventions

Technologies de l'information — Méthode de modélisation des conventions culturelles

Reference number
ISO/IEC TR 14652:2004(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

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

Published in Switzerland

Contents

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

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

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

In exceptional circumstances, the joint technical committee may propose the publication of a Technical Report of one of the following types:

— type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;

— type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;

— type 3, when the joint technical committee has collected data of a different kind from that which is normally published as an International Standard (“state of the art”, for example).

Technical Reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical Reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

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

ISO/IEC TR 14652, which is a Technical Report of type 1, was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces.*

This document is published as a type 1 Technical Report on the request of a number of member bodies, whose concerns are recorded in Annex D.

Introduction

This Technical Report defines a general mechanism to specify cultural conventions, and it defines formats for a number of specific cultural conventions in the areas of character classification and conversion, sorting, number formatting, monetary formatting, date formatting, message display, addressing of persons, postal address formatting, and telephone number handling.

There are a number of benefits coming from this Technical Report:

Rigid specification

Using this Technical Report, a user can rigidly specify a number of the cultural conventions that apply to the information technology environment of the user.

Cultural adaptability

If an application has been designed and built in a culturally neutral manner, the application may use the specifications as data to its APIs, and thus the same application may accommodate different users in a culturally acceptable way to each of the users, without change of the binary application.

Productivity

This Technical Report specifies those cultural conventions and how to specify data for them. With that data an application developer is relieved from getting the different information to support all the cultural environments for the expected customers of the product. The application developer is thus ensured of culturally correct behaviour as specified by the customer, and possibly more markets may be reached as customers may have the possibility to provide the data themselves for markets that were not targeted.

Uniform behaviour

When a number of applications share one cultural specification, which may be supplied from the user or provided by the application or operating system, their behaviour for cultural adaptation becomes uniform.

The specification format is independent of platforms and specific encoding, and targeted to be usable from a wide range of programming languages.

A number of cultural conventions, such as spelling, hyphenation rules and terminology, are not specifiable with this Technical Report, but it provides mechanisms to define new categories and also new keywords within existing categories. An internationalized application may take advantage of information provided with the FDCC-set (such as the language) to provide further internationalized services to the user.

This Technical Report defines a format compatible with the one used in the International string ordering standard, ISO/IEC 14651. This Technical Report is upward compatible with the ISO/IEC 9945-1:2003 POSIX, particularly its clauses 6 and 7. The major extensions from that text are listed in annex A. This Technical Report has enhanced functionality in a number of areas such as ISO/IEC 10646 support, more classification of characters, transliteration, dual (multi) currency support, enhanced date and time formatting, personal name writing, postal address formatting, telephone number handling, and management of categories. There is enhanced support for character sets including ISO/IEC 2022 handling and an enhanced method to separate the specification of cultural conventions from an actual encoding via a description of the character repertoire employed. A standard set of values for all the categories has been defined covering the repertoire of ISO/IEC 10646-1, as referenced in the normative references clause.

The Technical Report was originally scheduled for adoption as an International Standard, but a number of members of ISO and IEC found the specification problematical. It was then decided to convert the specification into a Technical Report type I. Annex D lists a number of issues that some members of ISO and IEC have with the specification.

The following clauses are thus marked as controversial:
4.3 LC_CTYPE
4.5 LC_MONETARY
4.7 LC_TIME
4.9 LC_XLITERATE
6 REPERTOIREMAP

Information technology — Specification method for cultural conventions

1 SCOPE

This Technical Report specifies a description format for the specification of cultural conventions, a description format for character sets, and a description format for binding character names to ISO/IEC 10646, plus a set of default values for some of these items.

The specification is upward compatible with POSIX locale specifications - a locale conformant to POSIX specifications will also be conformant to the specifications in this Technical Report, while the reverse condition will not hold. The descriptions are intended to be coded in text files to be used via Application Programming Interfaces, that are expected to be developed for a number of systems which comply with ISO/IEC 9945. An alignment effort has been undertaken for this specification to be aligned with the revision of the ISO/IEC 9945 standard published in 2003.

2 NORMATIVE REFERENCES

The following referenced documents are indispensible 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 639 (all parts), *Codes for the representation of names of languages*.

ISO/IEC 2022, *Information technology - Character code structure and extension techniques*.

ISO 3166 (all parts), *Codes for the representation of names of countries and their subdivisions*.

ISO 4217, *Codes for the representation of currencies and funds*.

ISO 8601, *Data elements and interchange formats - Information interchange - Representation of dates and times*.

ISO/IEC 9945:2003 *Information technology - Portable Operating System Interface (POSIX)*.

ISO/IEC 10646:2003, *Information technology - Universal Multiple-Octet Coded Character Set (UCS)*. Only the fixed collection 301 plus the characters U20AC EURO SIGN and the UFFFC OBJECT REPLACEMENT CHARACTER (corresponding to UNICODE 2.1) and the control characters U0000..U001F are used in this document.

ISO/IEC 14651:2001, *Information technology - International string ordering and comparison - Method for comparing character strings and description of the common template tailorable ordering*.

ISO/IEC 15897:1999, *Information technology - Procedures for registration of cultural elements*.

3 TERMS, DEFINITIONS AND NOTATIONS

3.1 Terms and definitions

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

3.1.1 Bytes and characters

3.1.1.1
byte:
An individually addressable unit of data storage that is equal to or larger than an octet, used to store a character or a portion of a character.

A byte is composed of a contiguous sequence of bits, the number of which is implementation defined. The least significant bit is called the low-order bit; the most significant bit is called the high-order bit.

3.1.1.2
character:
A member of a set of elements used for the organization, control or representation of data.

3.1.1.3
coded character:
A sequence of one or more bytes representing a single character.

3.1.1.4
text file:
A file that contains characters organized into one or more lines.

3.1.2 cultural and other major concepts

3.1.2.1
cultural convention:
A data item for information technology that may vary dependent on language, territory, or other cultural habits.

3.1.2.2
FDCC:
A Formal Definition of a Cultural Convention, that is a cultural convention put into a formal definition scheme.

3.1.2.3
FDCC-set:
A Set of Formal Definitions of Cultural Conventions (FDCC’s). The definition of the subset of a user’s information technology environment that depends on language and cultural conventions.

Note: the FDCC-set is a superset of the "locale" term in C and POSIX.

3.1.2.4
charmap:
A definition of a mapping between symbolic character names and character codes, plus related information.

3.1.2.5
repertoiremap:
A definition of a mapping between symbolic character names and characters for the repertoire of characters used in a FDCC-set, further described in clause 6.

3.1.3 FDCC categories related

3.1.3.1
character class:
A named set of characters sharing an attribute associated with the name of the class.

3.1.3.2
collation:
The logical ordering of strings according to defined precedence rules.

3.1.3.3
collating element:
The smallest entity used to determine logical ordering.

See collating sequence. A collating element consists of either a single character, or two or more characters collating as a single entity. The LC_COLLATE category in the associated FDCC-set determines the set of collating elements.

3.1.3.4
multicharacter collating element:
A sequence of two or more characters that collate as an entity.

For example, in some languages two characters are sorted as one letter, as in the case for Danish and Norwegian "aa".

3.1.3.5
collating sequence:
The relative order of collating elements as determined by the setting of the LC_COLLATE category in the applied FDCC-set.

3.1.3.6
equivalence class:
A set of collating elements with the same primary collation weight.

Elements in an equivalence class are typically elements that naturally group together, such as all accented letters based on the same letter.

The collation order of elements within an equivalence class is determined by the weights assigned on any subsequent levels after the primary weight.

3.2 Notations

The following notations and common conventions for specifications apply to this Technical Report:

3.2.1 Notation for defining syntax

In this Technical Report, the description of an individual record in a FDCC-set is done using the syntax notation given in the following.

The syntax notation looks as follows:

"<format>",[<arg1>,<arg2>,...,<argn>]

The <format> is given in a format string enclosed in double quotes, followed by a number of parameters, separated by commas. It is similar to the format specification defined in clause 5 in the ISO/IEC 9945-1:2003 standard and the format specification used in C language printf() function. The format of each parameter is given by an escape sequence as follows:

%s specifies a string
%d specifies a decimal integer
%c specifies a character
%o specifies an octal integer
%x specifies a hexadecimal integer

A " " (an empty character position) in the syntax string represents one or more <blank> characters.

All other characters in the format string represent themselves, except:

%% specifies a single %
\n specifies an end-of-line

The notation "..." is used to specify that repetition of the previous specification is optional, and this is done in both the format string and in the parameter list.

3.2.3 Portable character set

A set of symbolic names for characters in Table 1, which is called the portable character set, is used in character description text of this specification. The first eight entries in Table 1 are defined in ISO/IEC 6429 and the rest is defined in ISO/IEC 9945-1:2003 with some definitions from ISO/IEC 10646-1.

Table 1: Portable character set

Symbolic name	Glyph	UCS	Description
<NUL>		<U0000>	NULL (NUL)
<alert>		<U0007>	BELL (BEL)
<backspace>		<U0008>	BACKSPACE (BS)
<tab>		<U0009>	CHARACTER TABULATION (HT)

```
<carriage-return>                      <U000D>   CARRIAGE RETURN (CR)
<newline>                              <U000A>   LINE FEED (LF)
<vertical-tab>                         <U000B>   LINE TABULATION (VT)
<form-feed>                            <U000C>   FORM FEED (FF)
<space>                                <U0020>   SPACE
<exclamation-mark>       !             <U0021>   EXCLAMATION MARK
<quotation-mark>         "             <U0022>   QUOTATION MARK
<number-sign>            #             <U0023>   NUMBER SIGN
<dollar-sign>            $             <U0024>   DOLLAR SIGN
<percent-sign>           %             <U0025>   PERCENT SIGN
<ampersand>              &             <U0026>   AMPERSAND
<apostrophe>             '             <U0027>   APOSTROPHE
<left-parenthesis>       (             <U0028>   LEFT PARENTHESIS
<right-parenthesis>      )             <U0029>   RIGHT PARENTHESIS
<asterisk>               *             <U002A>   ASTERISK
<plus-sign>              +             <U002B>   PLUS SIGN
<comma>                  ,             <U002C>   COMMA
<hyphen-minus>           -             <U002D>   HYPHEN-MINUS
<hyphen>                 -             <U002D>   HYPHEN-MINUS
<full-stop>              .             <U002E>   FULL STOP
<period>                 .             <U002E>   FULL STOP
<slash>                  /             <U002F>   SOLIDUS
<solidus>                /             <U002F>   SOLIDUS
<zero>                   0             <U0030>   DIGIT ZERO
<one>                    1             <U0031>   DIGIT ONE
<two>                    2             <U0032>   DIGIT TWO
<three>                  3             <U0033>   DIGIT THREE
<four>                   4             <U0034>   DIGIT FOUR
<five>                   5             <U0035>   DIGIT FIVE
<six>                    6             <U0036>   DIGIT SIX
<seven>                  7             <U0037>   DIGIT SEVEN
<eight>                  8             <U0038>   DIGIT EIGHT
<nine>                   9             <U0039>   DIGIT NINE
<colon>                  :             <U003A>   COLON
<semicolon>              ;             <U003B>   SEMICOLON
<less-than-sign>         <             <U003C>   LESS-THAN SIGN
<equals-sign>            =             <U003D>   EQUALS SIGN
<greater-than-sign>      >             <U003E>   GREATER-THAN SIGN
<question-mark>          ?             <U003F>   QUESTION MARK
<commercial-at>          @             <U0040>   COMMERCIAL AT
<A>                      A             <U0041>   LATIN CAPITAL LETTER A
<B>                      B             <U0042>   LATIN CAPITAL LETTER B
<C>                      C             <U0043>   LATIN CAPITAL LETTER C
<D>                      D             <U0044>   LATIN CAPITAL LETTER D
<E>                      E             <U0045>   LATIN CAPITAL LETTER E
<F>                      F             <U0046>   LATIN CAPITAL LETTER F
<G>                      G             <U0047>   LATIN CAPITAL LETTER G
<H>                      H             <U0048>   LATIN CAPITAL LETTER H
<I>                      I             <U0049>   LATIN CAPITAL LETTER I
<J>                      J             <U004A>   LATIN CAPITAL LETTER J
<K>                      K             <U004B>   LATIN CAPITAL LETTER K
<L>                      L             <U004C>   LATIN CAPITAL LETTER L
<M>                      M             <U004D>   LATIN CAPITAL LETTER M
<N>                      N             <U004E>   LATIN CAPITAL LETTER N
<O>                      O             <U004F>   LATIN CAPITAL LETTER O
<P>                      P             <U0050>   LATIN CAPITAL LETTER P
<Q>                      Q             <U0051>   LATIN CAPITAL LETTER Q
<R>                      R             <U0052>   LATIN CAPITAL LETTER R
<S>                      S             <U0053>   LATIN CAPITAL LETTER S
<T>                      T             <U0054>   LATIN CAPITAL LETTER T
<U>                      U             <U0055>   LATIN CAPITAL LETTER U
<V>                      V             <U0056>   LATIN CAPITAL LETTER V
<W>                      W             <U0057>   LATIN CAPITAL LETTER W
<X>                      X             <U0058>   LATIN CAPITAL LETTER X
<Y>                      Y             <U0059>   LATIN CAPITAL LETTER Y
<Z>                      Z             <U005A>   LATIN CAPITAL LETTER Z
<left-square-bracket>    [             <U005B>   LEFT SQUARE BRACKET
<backslash>              \             <U005C>   REVERSE SOLIDUS
<reverse-solidus>        \             <U005C>   REVERSE SOLIDUS
<right-square-bracket>   ]             <U005D>   RIGHT SQUARE BRACKET
```

<circumflex-accent>	^	<U005E>	CIRCUMFLEX ACCENT
<circumflex>	^	<U005E>	CIRCUMFLEX ACCENT
<low-line>	_	<U005F>	LOW LINE
<underscore>	_	<U005F>	LOW LINE
<grave-accent>	`	<U0060>	GRAVE ACCENT
<a>	a	<U0061>	LATIN SMALL LETTER A
<b>	b	<U0062>	LATIN SMALL LETTER B
<c>	c	<U0063>	LATIN SMALL LETTER C
<d>	d	<U0064>	LATIN SMALL LETTER D
<e>	e	<U0065>	LATIN SMALL LETTER E
<f>	f	<U0066>	LATIN SMALL LETTER F
<g>	g	<U0067>	LATIN SMALL LETTER G
<h>	h	<U0068>	LATIN SMALL LETTER H
<I>	I	<U0069>	LATIN SMALL LETTER I
<j>	j	<U006A>	LATIN SMALL LETTER J
<k>	k	<U006B>	LATIN SMALL LETTER K
<l>	l	<U006C>	LATIN SMALL LETTER L
<m>	m	<U006D>	LATIN SMALL LETTER M
<n>	n	<U006E>	LATIN SMALL LETTER N
<o>	o	<U006F>	LATIN SMALL LETTER O
<p>	p	<U0070>	LATIN SMALL LETTER P
<q>	q	<U0071>	LATIN SMALL LETTER Q
<r>	r	<U0072>	LATIN SMALL LETTER R
<s>	s	<U0073>	LATIN SMALL LETTER S
<t>	t	<U0074>	LATIN SMALL LETTER T
<u>	u	<U0075>	LATIN SMALL LETTER U
<v>	v	<U0076>	LATIN SMALL LETTER V
<w>	w	<U0077>	LATIN SMALL LETTER W
<x>	x	<U0078>	LATIN SMALL LETTER X
<y>	y	<U0079>	LATIN SMALL LETTER Y
<z>	z	<U007A>	LATIN SMALL LETTER Z
<left-brace>	{	<U007B>	LEFT CURLY BRACKET
<left-curly-bracket>	{	<U007B>	LEFT CURLY BRACKET
<vertical-line>	|	<U007C>	VERTICAL LINE
<right-brace>	}	<U007D>	RIGHT CURLY BRACKET
<right-curly-bracket>	}	<U007D>	RIGHT CURLY BRACKET
<tilde>	~	<U007E>	TILDE

This Technical Report may use other symbolic character names than the above in examples, to illustrate the use of the range of symbols allowed by the syntax specified in 4.1.1.

4 FDCC-set

A FDCC-set is the definition of the subset of a user's information technology environment that depends on language and cultural conventions. It is made up from one or more categories. Each category is identified by its name and controls specific aspects of the behaviour of components of the system. This Technical Report defines the following categories:

LC_IDENTIFICATION	Versions and status of categories
LC_CTYPE	Character classification, case conversion and code transformation.
LC_COLLATE	Collation order.
LC_TIME	Date and time formats.
LC_NUMERIC	Numeric, non-monetary formatting.
LC_MONETARY	Monetary formatting.
LC_MESSAGES	Formats of informative and diagnostic messages and interactive responses.
LC_XLITERATE	Character transliteration.
LC_NAME	Format of writing personal names.
LC_ADDRESS	Format of postal addresses.
LC_TELEPHONE	Format for telephone numbers, and other telephone information.

Note: In future editions of this Technical Report further categories may be added.

Other category names beginning with the 3 characters "LC_" are reserved for future standardization, except for category names beginning with the five characters "LC_X_" which is not used for future addition of categories specified in this Technical Report. An application may thus use category names beginning with the five characters "LC_X_" for application defined categories to avoid clashes with future standardized categories.

This Technical Report also defines an FDCC-set named "i18n" with values for some of the above categories in order to simplify FDCC-set descriptions for a number of cultures. The contents of "i18n" categories should not necessarily be considered as the most commonly accepted values, while in many cases it could be the recommended values. The complete "i18n" FDCC-set is defined as the sum of the "i18n" categories specified in the clauses below.

4.1 FDCC-set description

FDCC-sets are described with the syntax presented in this subclause. For the purposes of this Technical Report, the text is referred to as the FDCC-set definition text or FDCC-set source text.

The **FDCC-set definition text** contains one or more FDCC-set category source definitions, and does not contain more than one definition for the same FDCC-set category. If the text contains source definitions for more than one category, application-defined categories, if present, appears after the categories defined by this clause. A category source definition contains either the definition of a category or a copy directive. In the event that some of the information for a FDCC-set category, as specified in this Technical Report, is missing from the FDCC-set source definition, the behaviour of that category, if it is referenced, is unspecified. A FDCC-set category is the normal way of specifying a single FDCC.

There are no **naming conventions** for FDCC-sets specified in this Technical Report, but clause 6.8 in ISO/IEC 15897:1999 specifies naming rules for POSIX locales, charmaps and repertoiremaps, that may also be applied to FDCC-sets, charmaps and repertoiremaps specified according to this Technical Report.

A **category source definition** consists of a category header, a category body, and a category trailer. A category header consists of the character string naming of the category, beginning with the characters "LC_". The category trailer consists of the string "END", followed by one or more "blank"s and the string used in the corresponding category header.

The **category body** consists of one or more lines of text. Each line is one of the following:

- a line containing an identifier, optionally followed by one or more operands. Identifiers are either keywords, identifying a particular FDCC, or collating elements, or section symbols,
- one of transliteration statements defined in 4.3.

In addition to the keywords defined in this Technical Report, the source can contain application-defined keywords. Each **keyword** within a category has a unique name (i.e., two categories can have a commonly-named keyword); no keyword starts with the characters "LC_". Identifiers are separated from the operands by one or more "blank"s.

Operands are characters, collating elements, section symbols, or strings of characters. Strings are enclosed in double-quotes. Literal double-quotes within strings are preceded by the <escape character>, described below. When a keyword is followed by more than one operand, the operands are separated by semicolons; "blank"s are allowed before and/or after a semicolon.

4.1.1 Character representation

Individual characters, characters in strings, and collating elements are represented using symbolic names, UCS notation or characters themselves, or as octal, hexadecimal, or decimal constants as defined below. When constant notation is used, the resultant FDCC-set definitions need not be portable between systems.

(0) The left angle bracket (<) is a reserved symbol, denoting the start of a symbolic name; when used to represent itself outside a symbolic name it is preceded by the escape character.

(1) A character can be represented via a **symbolic name**, enclosed within angle brackets (< and >). The symbolic name, including the angle brackets, exactly matches a symbolic name defined in a charmap or a repertoiremap to be used, and is replaced by a character value determined from the value associated with the symbolic name in the charmap

or a value associated via a repertoiremap. Repertoiremaps have predefined symbolic names for UCS characters, see clause 6. A FDCC-set may also use the UCS notation of clause 6 to represent characters, without a repertoiremap being defined for the FDCC-set. Use of the escape character or a right angle bracket within a symbolic name is invalid unless the character is preceded by the escape character.

Example: <c>;<c-cedilla> "<M><a><y>"

The items (2), (3), (4) and (5) are deprecated and are retained for compatibility with the POSIX standard. FDCC-sets should be specified in a coded character set independent way, using symbolic names. To make actual use of the FDCC-set, it is used together with charmaps and/or repertoiremaps, so that the symbolic character names can be resolved into the actual character encoding used.

(2) A character can be represented by the character itself, in which case the value of the character is application-defined. Within a string, the double-quote character, the escape character, and the right angle bracket character are escaped (preceded by the escape character) to be interpreted as the character itself. Outside strings, the characters

, ; < > escape_char

are escaped by the escape character to be interpreted as the character itself.

Example: c ä "May"

(3) A character can be represented as an octal constant. An octal constant is specified as the escape character followed by two or more octal digits. Each constant represents a byte value.

Example: \143; \347; "\115"

(4) A character can be represented as a hexadecimal constant. A hexadecimal constant is specified as the escape character followed by an x followed by two or more hexadecimal digits. Each constant represents a byte value.

Example: \x63;\xe7;

(5) A character can be represented as a decimal constant. A decimal constant is specified as the escape character followed by a d followed by two or more decimal digits. Each constant represents a byte value.

Example: \d99; \d231;