TECHNICAL REPORT

ISO/IEC
TR
30112

First edition
2014-06-15

Corrected version
2014-08-01

Information technology — Specification methods for cultural conventions

Technologies de l'information — Méthodes de spécification des conventions culturelles

iTeh STANDARD PREVIEW
(standards.iteh.ai)

COPYRIGHT PROTECTED DOCUMENT

Contents

Page

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.

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

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

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

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

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, SC 35, *User interfaces*.

This corrected version of ISO/IEC/TR 30112:2014 incorporates the following corrections: the cover page, the header on page 1 and the reference number have been corrected to reflect that this document is a Technical Report, and not an International Standard.

Introduction

This Technical Report defines general mechanisms 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 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 formats are 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 the Technical Report 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 parts of the ISO/IEC 9945 POSIX standard, especially those on POSIX locales and charmaps. 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, keyboard 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.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

Information technology — Specification methods for cultural conventions

1 Scope

This Technical Report specifies description formats and functionality for the specification of cultural conventions, description formats for character sets, and description formats 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 specifications in this Technical Report, while the reverse condition will not hold. Some of 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 ISO/IEC 9945.

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.

This document contains no normative references.

3 Terms and 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
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

Note 1 to entry: 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
member of a set of elements used for the organization, control or representation of data

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

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

3.1.2 Cultural and other major concepts

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

3.1.2.2
FDCC
Formal Definition of a Cultural Convention
cultural convention put into a formal definition scheme

3.1.2.3
FDCC-set
set of Formal Definitions of Cultural Conventions (FDCC's)
definition of the subset of a user's information technology environment that depends on language and cultural conventions

Note 1 to entry: the FDCC-set is a superset of the "locale" term in C and POSIX.

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

3.1.2.5
repertoiremap
definition of a mapping between symbolic character names and characters for the repertoire of characters used in a FDCC-set

Note 1 to entry: This is further described in clause 6.

3.1.3 FDCC categories related

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

3.1.3.2
collation
logical ordering of strings according to defined precedence rules

3.1.3.3
collating element
smallest entity used to determine logical ordering

Note 1 to entry: 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
sequence of two or more characters that collate as an entity

Note 1 to entry: 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
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
set of collating elements with the same primary collation weight

Note 1 to entry: 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 the ISO/IEC 9945 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.

3.2.2 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 with some definitions from ISO/IEC 10646.

Table 1 — Portable character set

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

Table 1 *(continued)*

Symbolic	name	Glyph	UCS Description
<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

Table 1 *(continued)*

Symbolic	name	Glyph	UCS Description
<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

Table 1 *(continued)*

Symbolic	name	Glyph	UCS Description
<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. A FDCC-set 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. The functionality is implied by the description of the categories. 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.

— LC_PAPER Paper format

— LC_MEASUREMENT Information on measurement system

— LC_KEYBOARD Format for identifying keyboards.

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. The "i18n" FDCC-set and its parts are released under the GNU Public License, version 2, as it is taken from glibc sources.

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;

(6) Multibyte characters can be represented by concatenated constants specified in byte order with the last constant specifying the least significant byte of the character. Concatenated constants can include a mix of the above character representations.

EXAMPLE   \143\xe7; "\115\xe7\d171"

Only characters existing in the character set for which the FDCC-set definition is created are specified, whether using symbolic names, the characters themselves, or octal, decimal, or hexadecimal constants. If a charmap is present, only characters defined in the charmap can be specified using octal, decimal, or hexadecimal constants. Symbolic names not present in the charmap can be specified and are ignored, as specified under item (1) above.

NOTE   The <character> symbolic character notation is recommended for use of specifying all characters in a FDCC-set, to facilitate portability of the FDCC-sets, as the coded character set of the application of the FDCC-set may be different from the coded character set of the FDCC-set source. This is also recommended for format effectors in strings, such as in LC_DATE or LC_ADDRESS, where the format effectors are allowed to be stored together with the rest of the string, in a binary string with a different encoding from that of the source FDCC-set.

4.1.2   Continuation of lines

A line in a specification can be continued by placing an escape character as the last visible graphic character on the line; this continuation character is discarded from the input. The line is continued to the next non-comment line.

4.1.3   Names for copy keyword

In most of the categories a "copy" keyword is allowed. The name specified with this copy keyword is one of:

— "i18n" which indicate the "i18n" FDCC-set defined in this specification,

— the name of a FDCC-set or POSIX locale registered by the process defined in ISO/IEC 15897,

— any other name which may be recognized in some local context - not being recommended as an international specification.

4.1.4   Pre-category statements

In a FDCC-set the following statements can precede category specifications, and they apply to all categories in the specified FDCC-set.

4.1.4.1   comment_char

The following line in a FDCC-set modifies the comment character. It has the following syntax, starting in column 1:

```
"comment_char %c\n", <comment_character>
```
The comment character defaults to the number-sign (#). All examples in this Technical Report use "%" as the <comment_character>, except where otherwise noted. Blank lines and lines containing the <comment_character> in the first position are ignored. In collating statements a <comment_character> occurring where the delimiter ";" may occur, terminates the collating statement.

4.1.4.2   escape_char

The following line in a FDCC-set modifies the escape character to be used in the text. It has the following syntax, starting in column 1:

```
"escape_char %c\n", <escape_character>
```
The escape character is used for representing characters in 4.1.1 and for continuing lines.