

Disposition of comments against DTR 10176**Technical Comments:****(1) Annex A: (Denmark, Japan, Netherlands, U.S.A.)**

- Add notes:

- (a) The character repertoire listed in this annex is based on the ISO/IEC 10646:1993, and subject to be changed to follow future amendments of the standard.
- (b) The character repertoire listed in this annex is a recommended repertoire for use of user defined identifier, and each programming language standard or implementation of the standard can modify the repertoire at the adaptation, considering the characteristics of the language and user requirements. For example, C language may allow LOW LINE character in addition to the character repertoire listed in the annex A, and COBOL may allow HYPHEN-MINUS as well.
- (c) Some programming language standard may allow half or full width characters in the compatibility zone. And some of them, e.g. COBOL, may recognize the characters in the manner of width insensitive.

- The following characters will be added into the list.

(a) Digits

The following digit characters will be added with the guidance that those characters should not be appeared at the head of identifiers.

0030..0039	DIGIT ZERO .. DIGIT NINE
0660..0669	ARABIC-INDIC DIGIT ZERO .. ARABIN-INDIC DIGIT NINE
06F0..06F9	EXTENDED ARABIC-INDIC DIGIT ZERO .. EXTENDED ARABIC-INDIC DIGIT NINE
0966..096F	DEVANAGARI DIGIT ZERO .. DEVANAGARI DIGIT NINE
09E6..09EF	BENGALI DIGIT ZERO .. BENGALI DIGIT NINE
0A66..0A6F	GURMUKHI DIGIT ZERO .. GURMUKHI DIGIT NINE
0AE6..0AEF	GUJARATI DIGIT ZERO .. GUJARATI DIGIT NINE
0B66..0B6F	ORIYA DIGIT ZERO .. ORIYA DIGIT NINE
0BE7..0BEF	TAMIL DIGIT ONE .. TAMIL DIGIT NINE
0C66..0C6F	TELUGU DIGIT ZERO .. TELUGU NINE
0CE6..0CEF	KANNADA DIGIT ZERO .. KANNADA DIGIT NINE
0D66..0D6F	MALAYALAM DIGIT ZERO .. MALAYALAM DIGIT NINE

0E50..0E59	THAI DIGIT ZERO .. THAI DIGIT NINE
0ED0..0ED9	LAO DIGIT ZERO .. LAO DIGIT NINE
0F20..0F29	TIBETAN DIGIT ZERO .. TIBETAN DIGIT NINE
0F2A..0F33	TIBETAN DIGIT HALF ONE .. TIBETAN DIGIT HALF NINE

(b) Letters

The following characters will be added.

0386	GREEK CAPITAL LETTER ALPHA WITH TONOS
040E	CYRILLIC CAPTITAL LETTER SHORT U
06D0	ARABIC LETTER E
06D1	ARABIC LETTER YEH WITH THREE DOTS BELOW
06D2	ARABIC LETTER YEH BARREE
06D3	ARABIC LETTER YEH BARREE WITH HAMZA ABOVE
06D5	ARABIC LETTER AE
06D6	ARABIC SMALL HIGH LIGATURE QAF WITH LAM WITH ALEF MAKSURA
0950	DEVANAGARI OM
0A74	GURMUKHI EK ONKAR
0ABD	GUJARATI SIGN AVAGRAHA
0AD0	GUJARATI OM
0CDE	KANNADA LETTER FA
0EDC	LAO HO NO (digraphs)
0CDD	LAO HO MO (digraphs)
0F00	TIBETAN SYLLABLE OM
0F40..0F47	tibetan consonants
0F49..0F69	tibetan consonants
1E9B	LATIN SMALL LETTER LONG S WITH DOT ABOVE
AC00..D7A3	hangle syllables

(c) Super and Subscript

00AA	FEMININE ORDINAL INDICATOR
00BA	MASCULINE ORDINAL INDICATOR
207F	SUPERSCRIPIT LATIN SMALL LETTER N

(d) Special characters

00B5	MICRON SIGN
00B7	MIDDLE DOT
02B0..02B8	phonetic modifiers derived from latin letters
02BB	phonetic modifiers derived from latin letters
02BD..02C1	phonetic modifiers derived from latin letters
02D0..02D1	phonetic modifiers derived from latin letters
02E0..02E4	phonetic modifiers derived from latin letters
037A	GREEK YPOGEGRAMMENI
0559	ARMENIAN MODIFIER LETTER LEFT HALF RING
093D	DEVANAGARI SIGN AVAGRAHA
0B3D	ORIYA SIGN AVAGRAHA
1FBF	GREEK PROSGEGRAMMENI
203F	UNDERTIE (general punctuation)
2040	CHARACTER TIE (general punctuation)

2102	letterlike symbols
2107	letterlike symbols
210A..2113	letterlike symbols
2115	letterlike symbols
2118..211D	letterlike symbols
2124	letterlike symbols
2126	letterlike symbols
2128	letterlike symbols
212A..2131	letterlike symbols
2133..2138	letterlike symbols
2160..2182	number forms
3021..3029	hangzhou-style numerals
3005	IDEOGRAPH ITERATION MARK
3006	IDEOGRAPH CLOSING MARK
3007	IDEOGRAPH NUMBER ZERO

(e) Combining characters (Level 2 of ISO/IEC 10646)

05B0..05B9	hebrew points and punctuation
05BB..05BD	hebrew points and punctuation
05BF	hebrew points and punctuation
05C1..05C2	hebrew points and punctuation
06D7..06DC	extended arabic letters
06E8	extended arabic letters
06EA..06ED	extended arabic letters
0901..0903	devanagari various signs
093E..094C	devanagari dependent vowel signs
094D	devanagari various signs
0951..0952	devanagari various signs
0963	DEVANAGARI VOWEL SIGN VOCALIC LL
0981..0983	bengali various signs
09BE..09C4	bengali dependent vowel signs
09C7..09C8	bengali dependent vowel signs
09CB..09CC	bengali dependent vowel signs
09CD	bengali various signs
09E2..09E3	bengali generic additions
0A02	GURMUKHI SIGN BINDI
0A3E..0A42	gurmukhi dependent vowel signs
0A47..0A48	gurmukhi dependent vowel signs
0A4B..0A4D	gurmukhi dependent vowel signs
0A81..0A83	gujarati various signs
0ABE..0AC5	gujariti dependent vowel signs
0AC7..0AC9	gujariti dependent vowel signs
0ACB..0ACC	gujariti dependent vowel signs
0ACD	GUJARITI SIGN VIRAMA
0B01..0B03	oriya various signs
0B3E..0B43	oriya dependent vowel signs
0B47..0B48	oriya dependent vowel signs
0B4B..0B4C	oriya dependent vowel signs
0B4D	ORIYA SIGN VIRAMA
0B82..0B83	tamil various signs

0BBE..0BC2	tamil dependent vowel signs
0BC6..0BC8	tamil dependent vowel signs
0BCA..0BCC	tamil dependent vowel signs
0BCD	TAMIL SIGN VIRAMA
0C01..0C03	telugu various signs
0C3E..0C44	telugu dependent vowel signs
0C46..0C48	telugu dependent vowel signs
0C4A..0C4C	telugu dependent vowel signs
0C4D	TELUGU SIGN VIRAMA
0C82..0C83	kannada various signs
0CBE..0CC4	kannada dependent vowel signs
0CC6..0CC8	kannada dependent vowel signs
0CCA..0CCC	kannada dependent vowel signs
0CCD	KANNADA SIGN VIRAMA
0D02..0D03	malayalam various signs
0D3E..0D43	malayalam dependent vowel signs
0D46..0D48	
0D4A..0D4C	malayalam dependent vowel signs
0D4D	MALAYALAM SIGN VIRAMA
0E31	THAI CHARACTER MAIHAM-AKAT
0E34..0E3A	thai vowels
0E47	THAI CHARACTER MAITAIKHU
0E48..0E4B	thai tone marks
0E4C..0E4E	thai signs
0EB1	LAO VOWEL SIGN MAIKAN
0EB4..0EB9	lao vowels
0EBB	LAO VOWEL SIGN MAIKON
0EBC	LAO SEMIVOWEL SOGN LO
0EC8..0ECB	lao tone mark
0ECC..0ECD	lao signs
0F18..0F19	tibetan signs
0F35	TIBETAN MARK NGAS BZUNG NYI ZLA
0F37	TIBETAN MARK NGAS BZUNG SGOR RTAGS
0F39	TIBETAN MARK TSA-PHRU
0F3E..0F3F	tibetan mark and signs
0F71..0F7D	tibetan dependent vowel signs
0F7E..0F81	tibetan various
0F82..0F84	tibetan marks and signs
0F86..0F8B	tibetan marks and signs
0F90..0F95	tibetan subjoined consonants
0F97	tibetan subjoined consonants
0F99..0FAD	tibetan subjoined consonants
0FB1..0FB7	tibetan subjoined consonants
0FB9	tibetan subjoined consonants

- The following characters will be removed from the list

(a) Special characters

0384	GREEK TONOS
05F3	HEBREW PUNCTUATION GERESH

05F4	HEBREW PUNCTUATION GERSHAYIM
0EAF	LAO ELLIPSIS
309D	HIRAGANA ITERATION MARK
309E	HIRAGANA VOICED ITERATION MARK
30FD	KATAKANA ITERATION MARK
30FE	HIRAGANA VOICED UTERATION MARK

(b) Japanese letters

3094	HIRAGANA LETTER VU
30F7	KATAKANA LETTER VA
30F8	KATAKANA LETTER VI
30F9	KATAKANA LETTER VE
30FA	KATAKANA LETTER VO

(c) Vacant position

040D
FB42

(d) Compatibility zone

F900..FA2D	cjk compatibiity Ideographs
FB1F..FB36	alphabetic presentation forms
FB38..Fb3C	alphabetic presentation forms
FB3E	alphabetic presentation forms
FB40..FB41	alphabetic presentation forms
FB43..FB44	alphabetic presentation forms
FB46..FB4F	alphabetic presentation forms
FB50..FBB1	arabic presentation forms-a
FBD3..FD3F	arabic presentation forms-a
FD50..FD8F	arabic presentation forms-a
FD92..FDC7	arabic presentation forms-a
FDF0..FDFB	arabic presentation forms-a
FE70..FE72	arabic presentation forms-b
FE74	arabic presentation forms-b
FE76..FEFC	arabic presentation forms-b
FF21..FF3A	full width latin capital letters
FF41..FF5A	full width latin small letters
FF66..FFBE	half width katakana letters
FFC2..FFC7	half width hangul letters
FFCA..FFCF	half width hangul letters
FFD2..FFD7	half width hangul letters
FFDA..FFDC	half width hangul letters

- The following character will be removed since they are categorized as Level3

(d) Hangul combining alphabet (Level3)

1100..1159	hangul jamo
1161..11a2	hangul jamo
11a8..11f9	hangul jamo

- The following code points were typo, then will be corrected as follows.

0E0D -> 0E8D Lao
5E76 -> FE76 CJK Unified

Attachment 1 Denmark

Due to the change in ISO/IEC 10646 of the encoding of Hangul characters, we propose to change the allowable characters defined in the appendix on extended identifiers as follows.

Remove the range: U3400..U4DFF
Insert the range: UAC00..UD7AF

Disposition: Accepted.(See above, to be discussed in WG20)
The Hangul characters in the area from AC00 through D7AF were added into the list. No action was taken for the area from 3400 through 4DFF, since the area had not been defined in the Annex A of DTR 10176.

Attachment 2 Japan

Japan's Comments on ISO/IEC DTR 10176, Title: Information technology -- Guidelines for the preparation of Programming language standards

The National Body of Japan approves ISO/IEC DTR 10176 with the following comments.

1. Category (Editorial) at the note 2 of 3.6.5
Proposed modification: replace "in not a" with "is not a".

Disposition: Accepted.

2. Category (Editorial) in the subclause 3.6
Problem: the third level clause number 3.6.7 and 3.6.8 are duplicated.
Proposed modification: renumber after the first occurrence of 3.6.8.

Disposition: Accepted.

3. Category (Editorial) at the note 1 and 2 of 3.6.11
Proposed modification: replace "of character" with "of a character".

Disposition: Accepted.

4. Category (Editorial) at the note 4 of 4.1.1

Problem: unnecessary line break exists.
Proposed modification: reformat.

Disposition: Accepted. Final text will not have the line break.
(No action, since unexpected format error)

5. Category (Editorial) at the 4.1.3
Proposed modification: move ", e.g. ISO/IEC 10646-1" at immediate
after of "multi-octet character set",
and add ", e.g. ISO/IEC 8859-1" at the end of the sentence.

Disposition: Accepted.

6. Category (Editorial) at the note 1 of 4.1.3.1.3
Proposed modification: replace "is by not English" with "is not
English".

Disposition: Accepted.

7. Category (Editorial) at the note 1 of 4.1.3.1.4
Proposed modification: remove the last sentence.
Reason: The annex A does not discuss about possible solution.

Disposition: Accepted.

8. Category (Editorial) at the first paragraph and note 1 of 4.1.3.3
Proposed modification: replace "every repertoire" with "entire
repertoire".

Disposition: Accepted.

9. Category (Editorial) at the note 1 of 4.1.3.3
Proposed modification: Replace "In the case if repertoire list which
enumerate allowable repertoire of characters for the character
datatype is not specified explicitly," with "In the case if the
value space of a character datatype is not specified explicitly, by
using the repertoire list that enumerate allowable repertoire of
characters for the datatype,"

Disposition: Accepted

10. Category (Editorial) at the last sentence of 4.1.3.3.3
Proposed modification: makes the last sentence as note and reword it
as "Assignment from a character datatype whose value space is ISO/IEC
646 IRV to another character datatype whose value space is
ISO/IEC 10646-1 is an example of inter character datatype
assignment." .

Disposition: Accepted

11. Category (Editorial) at the second sentence of 4.1.3.4.2
Proposed modification: replace "couture" with "culture".

Disposition: Accepted

12. Category (Editorial) at the note 5 of 4.1.3.5

Proposed modification: replace "being standardized as CD 14651" with "being standardized towards ISO/IEC 14651".

Disposition: Accepted

13. Category (Editorial) at note of 4.7.2

Proposed modification: replace "TR 11017" with "ISO/IEC TR 11017".

Disposition: Accepted

14. Category (Technical) in Annex A

Proposed modification: Add notes and clarify:

- (1) The character repertoire listed in this annex is based on the ISO/IEC 10646-1:1993, and subject to be changed if ISO/IEC 10646 is amended.
- (2) The character repertoire listed in this annex is a recommended repertoire for use of user defined identifier, and each programming language standard or implementation of the standard can modify the repertoire, considering the characteristics of the language and requirements, at the standardization or implementation of the language.

Disposition: (See above)

15. Category (Technical) in Annex A

Proposed modification: Remove the following characters from the list: 309b-309e, 30fd, 30fe, 3094, 30f7-30fa, and characters in the compatibility zone (f900-ffdc).

Disposition: (See above)

Attachment 3 Netherlands

COMMENTS TO THE NEGATIVE VOTE

Removal of Annex A is required to turn our NO vote into YES. This Annex contains rules for characters from scripts to be permitted in identifiers, without any indication that these are the right choice. Only the NBs of countries where these scripts are in use can state that, and these were not consulted. In particular, the rules for Indian scripts allow only for consonants, not vowels, in identifiers, which will cause great merriment in India, to the expense of the reputation of SC22/WG20 as a body of experts, and of SC22 as a serious standards developing group.

Disposition: Rejected. Because U.S. and Danish national bodies strongly objected against the removal. WG20 can not resolve both opinions. In stead,

the repertoire of the annex is modified (see above), and combining characters including Indian vowels become allowable.

A number of our comments in N 2163 appear to be proposed for rejection in N 2411 without any justification. Our vote will remain NO as long as no clarification is given.

Disposition: WG20 sorry for that the Netherlands comment on 4.1.3.1.3, regarding Indian script has not well addressed in the DTR. The comment is now resolved by the modification of Annex A (see above).

Editorial comments

It is a pity that in a DTR still sentences occur not checked for correct English. Omission of the Definite or Indefinite Article is not allowed in the English language, a well known stumble-block to Japanese writers. Some are even not understandable. We mark these with This Sentence Is Incomprehensible (TSII).

3.6.3 Change:

Each element of a combining sequence --

Each element of a composite sequence

Add after last sentence:

(as it is in ISO/IEC 10646-1.)

Disposition: Accepted

3.6.5 Change:

(Note 2)

A composite sequence in not --

A composite sequence is not --

Disposition: Accepted

3.6.12 Note 2:

Insert "the" and "a".

Disposition: Accepted

3.6.16 Note 2 Change:

the same with --

the same as

Disposition: Accepted

4.1.3.1.2 Note 4 Change:

for coding --

for character coding

(SC29 is developing standards for audio-visual coding, not meant here.)

Disposition: Accepted

4.1.3.1.3 Note 3:

The SC2 intends

Remove this sentence. A TR is about facts, not intentions.

Disposition: Accepted. Rewords the sentences and removed the word "intends", since the character short identifier has already been standardized by ISO/IEC JTC1/SC2.

4.1.3.1.4 Note 3:

Remove this note. A TR is about facts, not intentions.

Disposition: Accepted. Rewords the sentences, since the character short identifier has already been standardized by ISO/IEC JTC1/SC2.

4.1.3.2 Note 1 Change:

variant --

version

Disposition: Accepted

4.1.3.3.1 Notes 1, 2 " "

STII (see above)

Disposition: Rejected. WG20 suppose the words "repertoire-list" may cause the confusion. The "repertoire-list" comes from ISO/IEC 11404 Language-independent datatype. According the the standard, the character datatype shall be specified as "character" ["(" repertoire-list ")"], where the repertoire-list indicates allowable character repertoire for the character datatype.

4.1.3.4.2 Notes 1, 2, 3

STII (see above)

"couture" ???

Disposition: Accepted. The "couture" is a typo of "culture".

4.1.3.6 Change:

whose values space --

whose value space

(NOTE):

should not to require --

should not require --

Disposition: Accepted

4.1.3.1.4, 4.7.2 Change:

Programming language committee should consider --

The Programming language committee should consider --

Disposition: Accepted

Attachment 4 Sweden

Editorial comment: The document has been typeset for US Letter format, which is evident from the uneven margins on even/odd pages when printed in A4 paper. Overall, the margins are too narrow for easy use, at least when printed on A4.

Disposition: Accepted. WG20 sorry that the distributed document was formatted by US letter format. The final text for the TR will be ISO A4 format.

Attachment 5 UK

UK vote on JTC1 N4579 - DTR 10176: Document SC22/WG20 N477

The UK votes NO. The vote will become YES if Issues 1-3 and 12 are resolved satisfactorily.

A number of issues which were identified as major technical issues in the PDTR ballot were not resolved satisfactorily, or at all, in the Disposition of Comments SC22 N2163:

Issue 1:

Clause 4.7 provided unclear and minimal guidance for the handling of non-character set related issues for internationalization. A recommendation for WG20 was made.

Disposition:

None provided, but a reference to TR 11017 Framework for internationalization exists.

Action:

4.7.2 is especially unclear as to the meaning and needs to be clarified for subsequent processing.

Disposition: Rejected. WG20 believes that guidelines for cultural convention related function should be minimal, since the support requirements of the function may vary from a programming language to another. In stead of having the guidelines in this TR, WG20 has a project that establish internationalization API standard. The internationalization functions that can be utilized from every programming language will be specified by the standard.

Issue 2:

Clause 4.1.3.1.1 provided no guidelines for ISO 10646 handling.

Disposition:

Refers to CHARACTER datatype.

Action:

Datatypes are not relevant to character sets used for program text. Hence the original problem is still unresolved.

Disposition: Rejected . The TR provides ISO/IEC 10646 handling

in program text in 4.1.3.1.2, 4.1.3.1.3, and 4.1.3.1.4.
WG20 will address further guidelines for ISO/IEC 10646 and the further guidelines will be added at the future revision of the TR 10176, when becomes ready.

Issue 3:

Clause 4.1.3.4.2 makes no recommendations about classes of characters which should be provided for internationalized applications.

Disposition:

None

Action:

Recommendations need to be included.

Disposition: Rejected. The requirements for the character transliteration may vary from a programming language to another, and the classes of characters may vary from culture to another, therefore it is difficult to include a common recommended class of characters across programming languages and human cultures.

This issue will be addressed in the development of IS 14652 that WG20 is now working.

In addition a number of other issues need to be addressed:

Issue 4:

Clause 3.6.8 refers to a family when it is not clear that a family is being referred to. (Note the use of data type as two words or datatype as one word is inconsistent throughout.)

Action:

Change definition to A character datatype is a datatype whose value space is a character set. Also replace wide with large in the Note.

Disposition: Reject. This definition comes from ISO/IEC 11404 Language-independent datatypes..

Issue 5:

Clause 3.6.9 muddles codes and values.

Action:

Replace definition with An octet datatype is the datatype whose values are single octets (often used for character sets and private encoding.) Also replace wide with large twice in the Note.

Disposition: Rejected. This definition comes from ISO/IEC 11404.

Issue 6:

Clause 3.6.10 is confused.

Action:

Replace definition with An octet string datatype is a datatype of variable-length whose elements are of an octet datatype. Also replace of extended character sets with an extended character set .

Disposition: Rejected This definition comes from ISO/IEC 11404.

Issue 7:

Clause 3.6.12 could refer to non-integer multiples of octets

Action:

Replace that size is equal to or larger than two octets with whose values are multiple octets

Disposition: Accepted.

Issue 8:

Clause 4.1.3.1.3 Note 2 last sentence refers to an Annex A which no longer exists.

Action:

Delete

Disposition: Accepted.

Issue 9:

Clause 4.1.3.1.3 Note 3 last sentence refers to an SC2 intention.

Action:

Either delete or fully explain.

Disposition: Accepted. The character short identifier has already been standardized by ISO/IEC JTC1/SC2. The words "intends" will be removed.

Issue 10:

In the DTR Clause 4.1.3.2 was identified as limiting to sequence of octets.

Disposition:

WG20 intended to review, but no changes have been made, so the problem is still outstanding.

Action:

WG20 should review before further progress.

Disposition: Rejected. WG20 believes that it is too tough for implementations of programming language standards to support all encoding schemes of coded character sets in the world, and some of character data handled by programming languages do not have an identification of the coded character set that the character data is encoded. Therefore, it is too difficult for programming language to handle character data as "character" regardless of its encoding. For the time being, the removal of assumption on a specific encoding is the best can do effort for the programming language.

Issue 11:

Clause 4.1.3.3.2 Notes 2 does not make any sense to this (English) reader.

Is it trying to say that portability can be maintained if octet datatypes are used? This may be true for a limited subset of portability issues. If so then it should say which classes of portability would be maintained.

Action:
Needs to be re-written

Disposition: Accepted. Clarified it is for existing programs that assumes that the size of character datatype is an octet, and shares a memory area between the character datatype and another datatypes.

Issue 12:
Clause 4.1.3.6 Note was recommended to be replaced with a guidelines. This was accepted by WG20, but no change has been made to the document.

Action:
Change needs to be made in response to original problem statement before further progression of the document.

Disposition: Accepted.

Issue 13:
Clause 4.1.3.7 a and b) mentions octet-string when header refers to multi-byte

Action:
In a) delete stored in an octet string datatype
In b) delete in an octet string datatype

Disposition: Rejected. The multi-byte representation of characters are only stored in either octet or octet string datatype.

Issue 14:
Many English problems

Action:
Issue :
Clause 3.6.11
In Note 2 replace character bound with the character boundary

4.1.3.1.5
second sentence insert a after permit

4.1.3.3.1
Replace the character with a character and is by includes
In Note 1 replace if by that a and emamerate by enumerate

4.1.3.3.2
Replace use by provide
In Note 1 replace wide by large and all repertoire by all repertoires

In Note 3 delete to
4.1.3.4.2
In second sentence replace couture by culture

In Note 2 replace will be used by by should be usable by a
4.1.3.5

In the second paragraph replace one of by a
4.1.3.6

Replace the character by a character
In the Note replace should not to by need not , should be stored in by could be stored in a

In the final paragraph insert single value of a before datatype , the before provision , and replace distinct datatype from character datatype by datatype distinct from other character datatypes

4.1.3.7

Replace in by using

In b) replace bound by boundary

Disposition: Accepted except 4.1.3.3.2, since the octet and octet string may be provided by other purposes, e.g. to store an integer value or bool value.

Attachment 6 USA

The US National Body votes to Disapprove with comments ISO/IEC DTR 10176 - Guidelines for the Preparation of Programming Language Standards.

See Comments listed below.

Comments:

General comments

In general, we are finding TR 10176 rather uninformed about object-oriented language design and mostly irrelevant to the major new language development that it might be attempting to address, namely Java. We also finding that the document is anchored in the past in its usage of terminology and its application of coded character sets. These points are developed in the technical comments section.

Furthermore, the document requires a lot of editorial work, there are many typos and many parts of the document are difficult to understand text sections. These issues are explained in the following editorial comments.

Overall, the U.S. position is that the document should be withdrawn, unless it is completely rewritten to take into account the current language technology and submitted to a comprehensive editorial phase.

Technical comments

a) Byte terminology

Ref: 3.6.1, 3.6.11 and 3.6.12

The usage of the byte terminology should be completely avoided. This document is redefining the byte in 3.6.2 in a manner slightly different from well known standards (for example ISO/IEC 2022:1994 defines it as 'a bit string that is operated upon as a unit'). Despite

the fact that these definitions refer to the byte as an entity with a variable bit size, it is a well-established practice that the byte is assimilated to an octet. To avoid the issue, increasingly standards

are referring only to the octet that has a very precise association with 8-bit encoding. Using the concept of multi-byte and two-octet bytes in the same sentence is more confusing than clarifying.

Disposition: Rejected. Since the term of "byte" was used in the approved previous edition of the TR, the use of the term can not be avoided. Although most of implementation of programming languages implement "byte" as octet, it is not necessary from the view point of programming language standards, for example C language has a keyword CHARBIT that specify bit size of "byte". To follow the guideline for the provision of character datatype that value space is entire repertoire of ISO/IEC 10646 in C language, CHARBIT=16 is an option for implementations of C language.

b) Guideline: Character sets used for program text
Ref 4.1.3.1.1

New languages should not be restricted to the usage of invariant part of ISO/IEC 646. This limitation is not realistic anymore and is de facto ignored by new languages. This would exclude characters like '[]{}' that are commonly used in today languages. A vast majority of programmers is using environments based on (or related to) ISO/IEC 8859 or even ISO/IEC 10646, not national variants of ISO/IEC 646. This guideline is anchored in the past, not the present situation.

Disposition: Rejected. National versions of ISO/IEC 646 are still widely used in the world. Therefore, WG20 believes that the guideline is still valid. Note that the guideline does not prohibit use of the outside repertoire of the ISO/IEC 646 invariant set, but recommends to provide an alternative representation of the characters, e.g. trigraphs of C language.

c) Guideline: Guideline: Character datatype
Ref 4.1.3.3.1

"The character datatype should be independent from any coded character set."

The most recent developments in language technology like Java are being done, with good reasons, in contradiction with this guideline. The correct way to implement 10646 in a computer language is to *identify* the character datatype with a fixed-width encoding form of the universal character set. The point of a universal character set for a programming language is to *use* the universal character set directly, not simply to treat it as a reference by which to define all the single-byte, multi-byte anarchy that is currently implemented.

Disposition: Partially accepted. The most of programming language standards are developed for keeping source code portability, therefore encoding of character is outscope of the standard. In case of Java, it also addresses to maintain object code level portability, i.e. Java Bytecode level, portability, thus encoding of character need to be specified in the standard. Add a sentence that clarifies that the programming languages which

address to object code level portability is not the case. And recommend ISO/IEC 10646 encoding for such programming languages.

d) Guideline: Character transliteration
ref 4.1.3.4.2

This section distorts the standard meaning of "transliteration". What is meant here are classes of character transformation, explicitly case-transformation, and width-transformation (for Japanese hankaku/zenkaku characters). In addition, we presume that 'couture' was supposed to be 'culture'.

Disposition: Accepted.

e) Guideline: Cultural convention set switching mechanism
ref 4.7.1

This guideline for provision of a mechanism such as `setlocale()` on a per-thread basis is too limited and inappropriate for object oriented languages like Java. Java provides I18N functionality through a set of classes which reflect an entirely different architecture.

Disposition: Accepted. Add locale object as an alternative.

In general the guidelines in TR 10176 reflect a view of programming languages which generally seems to be completely uninformed by object-oriented programming language design. This is just one example.

Disposition: Rejected.

As pointed in the U.S. comment, the guidelines provided by this TR may not well fit to object-oriented programming languages. However, WG20 believes that the guidelines provided by this TR are applicable to modern object-oriented programming languages, such as Java.

f) Recommended extended repertoire for user-defined identifier
Ref Annex A

The annex needs a complete reworking.

This annex has errors scattered throughout. (e.g. U+0384 where U+0386 is clearly intended, in the Greek set) It is not up-to-date against Unicode 2.0 (or 10646 plus Amendments). For example, it arbitrarily omits the Hangul syllables (U+AC00 to U+D7AF3), and the CJK Unified Ideographs is a mixed bag containing characters that have nothing to do with Ideographs (Arabic presentation forms, Halfwidth and Fullwidth forms of Latin, Kana and Hangul,.). It also arbitrarily legislates against modifier letters or IPA values for identifiers.

However, the worst error is in claiming that combining marks do not

belong in identifiers. The fallacy of this can be seen by looking at the Devanagari list, which is utterly nonsensical. The recommended values for identifiers are U+0905-U+0939, U+0958-U+0962. In other words, this annex is recommending that *only consonants or initial vowels* are o.k. in Devanagari identifiers, but other vowels and virama should be omitted.

See the Unicode Standard, pp. 5-25 to 5-27, plus corrections posted on the Unicode website, for a meaningful recommendation regarding how to extend identifier syntax to the 10646 repertoire. (The Java implementation of identifiers in Unicode is very close to this recommendation.)

Disposition: Partially accepted. (See above)

The level 2 set of combining characters are added in the recommended list of annex A. Also, it is clarified that each programming language standard, such as Java, can modify the recommended repertoire and apply it in the standard specification.

Editorial comments

a) Guidelines on the use of character sets Ref 4.1.3

The following text "including multi-octet character sets and non-English single octet character sets, e.g. ISO/IEC 10646-1." is well intentioned, but badly worded, since it implies that 10646-1 is a single octet character set! This could be improved by swapping the two elements of the sentence.

Disposition: Accepted.

b) Guideline: Character sets used in character literals Ref 4.1.3.1.4

"Any conforming processor should be required to accept method c) to reeprresent a character literal outside of "minimal set" defined in 4.1.3.1.1, any "non-printing character", or any special-purpose character, in a way that is independent from code value of the character of the character in any coded character set."

We do not understand the meaning of the paragraph. The representation of a literal by its 10646 value cannot be independent of its code value *in* 10646, which is itself a coded character set.

Disposition: Accepted. Clarify the coded character set referred to is source code coded character set, not the coded character set of the literal itself.

c) Guideline: Character sets used in comments
Ref 4.1.3.1.5 (Note)

Change

"... Since comments are intended for human reading and hence escape mechanisms are unnecessary, there is no disadvantage in printing characters simply representing themselves (apart of course from any characters or sequences of characters marking the end of the comment), and in limiting non-printing characters to those (like carriage return and line feed) necessary for layout purposes."

By

"Program comments are intended for human reading. Except for the provision of unambiguous characters or sequences of characters to delimit the comments, the specification of a computer language should not restrict characters which can occur in comments. No escape mechanism should be necessary for inclusion of any character in comments."

Disposition: Rejected. This sentences are inherited from approved previous edition of this TR.

d) Guideline: Character datatype

"The programming language standard should provide the character datatype whose value space is every repertoire of the extended character set in an execution environment."

We don't understand that sentence. We presume this intends to say "the entire repertoire". Furthermore, the intent of the term "extended character set in an execution environment" is unclear.

Note 1: "In the case if repertoire list which emamerate allowable repertoire of characters for the character datatype..."

That note is completely incomprehensible. Besides the obvious typos (emamerate instead of enumerate, lack of articles, etc.), we cannot make any sense of the note. This defeats the purpose of this document that is aiming at being a set of 'guidelines'.

Disposition: Partially accepted. The "every" is replaced with "entire" as suggested. "execution environment" is defined in 3.6.16. The "repertoire list" is specified in ISO/IEC 11404 Language-independent datatype, therefore WG20 believes that the term is understandable for programming language committes.

e) Guideline: Octet and octet string datatype
ref 4.1.3.3.2

Note 1 "The value space of the octet datatype is wide enough to represent every repertoire of the basic character set, but not all repertoire in the extended character set."

Again, we don't understand this, we presume the author meant "the entire" for "every" and "all" here. The sentence needs to be completely rewritten (no suggestion as we don't understand it).

Disposition: Accepted.