

Disposition of comments to FCD 14652 Specifications for Cultural Conventions

1998-10-12

Source: ISO/IEC JTC1/SC22/WG20

Status: final disposition of comments

In the following the disposition of comments is given with respect to the FCD ballot in SC22 N2638, Information technologies - Specifications for Cultural Conventions.

The ballot had the following result:

SUMMARY OF VOTING ON

Letter Ballot Reference No: SC22 N2638
 Circulated by: JTC 1/SC22
 Circulation Date: 1998-01-21
 Closing Date: 1998-06-04
 Closing Date Extended to: 1998-06-12 (at the request of the UK)

SUBJECT: FCD Approval for FCD 14652 - Information technology -
 Programming languages, their environments and system software
 interfaces - Specifications for Cultural Conventions

 The following responses have been received on the subject of approval:

"P" Members supporting approval without comments	10
"P" Members supporting approval with comments	2
"P" Members not supporting approval	3
"P" Members abstaining	2
"P" Members not voting	5
"O" Members supporting approval without comments	1
"O" Members abstaining	1
Other JTC 1 Member Bodies not supporting approval	1

Reference Document No: N2638
Circulation Date: 1998-01-21

Ballot Document No: N2638
Closing Date: 1998-06-04
Extended to 1998-06-12 at UK
request

SUMMARY OF VOTING AND COMMENTS RECEIVED

	Approve	Disapprove	Abstain	Comments	Not Voting
'P' Members					
Australia	(X)	()	()	()	()
Austria	()	()	()	()	(X)
Belgium	(X)	()	()	()	()
Brazil	()	()	()	()	(X)
Canada	(X)	()	()	(X)	()
China	()	()	()	()	(X)
Czech Republic	(X)	()	()	()	()
Denmark	(X)	()	()	(X)	()
Egypt	(X)	()	()	()	()
Finland	(X)	()	()	()	()
France	(X)	()	()	()	()
Germany	()	()	(X)	(X)	()
Ireland	(X)	()	()	()	()
Japan	()	(X)	()	(X)	()
Netherlands	()	(X)	()	(X)	()
Norway	(X)	()	()	()	()
Romania	(X)	()	()	()	()
Russian Federation	()	()	()	()	(X)
Slovenia	()	()	()	()	(X)
UK	()	()	(X)	()	()
Ukraine	(X)	()	()	()	()
USA	()	(X)	()	(X)	()

'O' Members Voting

Korea Republic	(X)	()	()	()	()
Sweden	()	()	(X)	(X)	()

Other JTC 1 Member Bodies Voting

Israel	()	(X)	()	(X)	()
--------	-----	-----	-----	-----	-----

In the following the comments from national bodies are indented by a leading string of "> " and responses are recorded after the comment.

> **Canadian vote on ISO/IEC FCD 14652. We vote YES, with
> the annexed comments.**

> _____
> General comments and suggestions:

> 1. Drafts should use "change bars" in the margins to indicate text that
> has been changed from the previous draft. This would certainly help
> in reviewing the document and would speed up the process. As it
> stands today, one has to re-read every single word again and again
> as draft revisions are created for review; this is time-consuming
> and not very productive.

>
Response: Accepted in principle. The current document processing
system that this document is prepared with, does not have this
capability, but the editor will provide an overview of the changes.

> 2. All examples, and their accompanying text, should be enclosed in a
> "box" (i.e. mark it as a figure). This makes it stand out and there
> is no confusion as to where the example text begins and ends. It
> will enhance the document's readability.

Response: accepted in principle. The editor will make examples
clearer stand out as examples.

> 3. Syntax of the keywords would be better understood if it was in plain
> text rather than in C terms with the %s and %c and \n etc. That
> only confuses a reader.

Response: The editor will look at it with help from the Canadian
member body. BNF notation will be introduced.

> 4. Keywords in text should be bold to enhance readability.

Response: accepted in principle. The text will be enhanced to improve
readability, possibly by enclosing them in double quotes.

>
> Specific comments:

> 1. Under "..benefits coming from this standard" (prior to SCOPE):

> a) Cultural Adaptability: as written this is not true; it is only
> true if the application is designed and implemented in a culturally
> neutral manner. Only then can I use the same binary to support
> different cultural conventions.

Response: accepted in principle. The concept mentioned here is that
described in the introduction as "Internationalization". It is expected
that all of the benefits listed will be realized. The text will
be carefully investigated for its promises, and most likely
be worded with less strength.

> b) Internationalization: "An application developer can remove cultural
> dependencies from an application, using the localized data given by
> the customer." This implies that for an existing application, the
> localized data will help the application developer to remove

> cultural dependencies from the application! What needs to be stated
> here is an internationalized application needs to be designed and
> implemented as culturally neutral and that, at run time, it draws on
> the cultural conventions of the user thus giving the application the
> ability to support many different cultural conventions. This standard
> specifies those cultural conventions.
>
> The rest of the this paragraph also needs to be re-worded with this
> in mind.

Response: accepted. The wordings will be used.

> c) Uniform behaviour: Disagree with the statement as written. It
> implies that the end user has control and that if the user
> codes up the cultural conventions, all applications can take
> advantages of these. This is not true. It is the applications (and
> the platform + OS) that have the primary responsibility to be
> designed and implemented to take advantage of cultural conventions
> not the user. If all applications used the same set of cultural
> conventions then the end-user would get consistent and correct
> cultural behaviour.

Response: accepted in principle. The text will be changed to
describe possible scenarios of control.

> d) Second sentence, paragraph beginning "This International..". It
> says "This Internal Standard..". Internal to whom?

Response: accepted. Changed to International.

> e) Paragraph beginning "This International..", mid-paragraph. It talks
> about handling paper, measurement system etc. Change "handling"
> to either formatting or identification because this specification
> does not handle any of this and it only identifies these elements.
> Need to change "paper" to "paper size".

response: accepted. Also reference as base for 14651 will be removed.

> 2.Under SCOPE:

>
> First paragraph: "The specification is upward...". This implies that
> this standard is for POSIX only. I thought we agreed in Egypt to
> extend it beyond POSIX so that Java etc. can also take advantage of
> these convention specifications. Re-wording to take this into
> account would help.

Response: accepted. Also see Japanese comments.

> 3.Under Terms and definitions (3.1)

> a) 3.1.5 - change "circumstances" to "conventions".

Response: accepted in principle. The clause will be reworded,
but avoiding circular definitions.

> b) 3.1.10 and 3.1.11 - these should not be included here. Else, all
> of the other keywords under LC_CTYPE should also be included here.

Response: accepted

> c) 3.1.13 - replace "the logical ordering of strings" with "logical
> ordering".

Response: accepted.

> d) 3.1.13 - "the value of the LC_COLLATE" - this does not make sense.

Response: accepted in principle. The value is the object giving
all the data for the LC_COLLATE category. The clause will be reworded.

> e) 3.1.14 - replace "letter, this is the" with "letter, as in the".

Response: accepted.

> f) 3.1.15 - replace "setting of LC_LOCALE" with "settings of
> LC_COLLATE"

Response: accepted.

> g) 3.1.16 - this restricts equivalence to primary weight only. This
> is incorrect. Also see later comments on this.

Response: rejected. The definition is taken from 9945-2. It is used
for API support in regular expressions and is only valid
for the first level.

> h) 3.1.17 and 3.1.18 - explain "yesexpr" and "noexpr".

Response: accepted.

> 4.Under 4.2.1

>

> a) General comment on the additions that have been made to LC_CTYPE:
> These are significant additions and it is not obvious as to the
> intended use of these. Supporting rationale should be included here
> to ensure a fair and sound understanding.

Response: accepted.

> b) outdigit - why is this required and how is it different from
> "digit"

"digit" classifies all digits, while "outdigit" classifies the
values used for outputting. This keyword is needed to determine
the characters used for outputting.

> c) class - the first sentence needs to be re-worded.

Accepted.

> d) left_to_right and right_to_left - is this a value or an indicator?
> Can this not be accomplished by having a default orientation
> indicated elsewhere in the locale? Why do you think is required
> in LC_CTYPE?

These keywords will be removed.

- > e) `left_to_right` and `right_to_left` - why not also have indicators for
- > `top_to_bottom` and `bottom_to_top`? This vertical orientation, in
- > addition to the above horizontal orientation, completes the set.

These will be removed due to other ballot comments.

- > f) `num_terminator` - is this a control, space, printable or
- > punctuation character? If yes, then it belongs in those classes
- > and we don't need to create another class.
- >
- > g) `num_separator` - as above.
- >
- > h) `segment_separator`, `block_separator`, `direction_control` - all of
- > these belong in the control class, it appears.
- >
- > Given that we are defining these blocks, perhaps we should also
- > look at defining: `word_break`, `line_break`, `paragraph_break` and
- > `page_break`.
- >
- > i) `sym_swap_layout`, `char_shape_selector`, `num_shape_selector` - as per
- > all the other class definitions the characters should be defined
- > here and not just referenced to another standard. As comments above,
- > do these characters not fit into a already defined class?
- >
- > j) `non_spacing_level3` - what happened to level 1 and level 2?
- >
- > k) see general comment above for the series of `*_connect*` classes.
- >
- > l) `special1`, `special2`, `special3` - what is the difference between these
- > classes? Why are they needed? I don't think that we can put in such
- > open-ended classes in this standard.
- >
- > m) `tosymmetric` - why is this class needed?

These will all be removed, due to resolving of other comments.

- > n) table 1 - why does this table not show the new classes defined?
- > As it stands, with the new classes defined in `LC_CTYPE`, this is
- > incomplete.

The new classes do not have inclusion/exclusion rules on them.

- > 5.Under 4.2.2
- >
- > Transliteration does not really belong in `LC_CTYPE`. It should in
- > a category by itself; perhaps called `LC_XLITERATE`.

Coordination with SC22/WG15 has shown that they clearly preferred the spec to be under `LC_CTYPE`, and for compatibility with POSIX this is thus retained.

- > 6.Under 4.3
- >
- > a) first paragraph "..the collation sequence definition shall..".

> The "shall" is mandating. I don't think that is intended. Perhaps
> we can say "should be used in string comparison and sorting..".

change "shall be" to "is".

> b) equivalence class definition: it should not be restricted to
> primary weight only; it can be up to any level. Perhaps we can say
> "..two or more collating elements have the same collation values
> upto a specified level..".

This is POSIX wording. it defines a class of collating elements, in a mathematical sense. On other levels there is not defined a class, for example all letters with <ACUTE>. The equivalence class term is used in regular expressions, only at the first level. We see no need for an expanded term.

> c) per script ordering rules: this is confusing with the use of
> culture rather than language and script. It needs to be re-worded.

A language or culture may need more than one script, vs. Japanese or Korean. A culture could be a mathematical culture within a country having special requirements on scripts, for example on Greek and Hebrew.

>
> d) easy ordering of scripts: as (c) above.

Response: Moot. The script concept goes away, due to changes in 14651.

> e) coll_weight_max: as stated last time, the minimum value cannot be
> 7 - in ISO 14651, this value was stated as 4.

Accepted in principle. 7 is the minimum that an implementation shall honour. The actual value in a FDCC-set may be less.

> 7.Under 4.3.1
>
> a) fifth paragraph "The ellipsis..". Because there are three ellipses
> used (.. and ... and) we should distinguish between them.
> Perhaps the word "absolute" needs to be used here for this
> definition (we say "symbolic ellipses" elsewhere).

Accepted. Use "binary"?

> b) paragraph beginning "..All characters specified.." - equivalence
> class sentence to be re-worded as per 6(b) above.

Rejected, as per responses to CA6(b) above.

> c) paragraph beginning "..The special keyword.." - same as above for
> equivalence class.

Rejected, as per responses to CA6(b) above.

> 8.Under 4.3.4
>
> The use of the word "identifier" may be better instead of symbol in
> this case because the "script-symbol" is not really a symbol but
> an identifier string.

accepted

> 9.Under 4.3.8

>
> Paragraph beginning "The directives forward and backward are
> mutually exclusive". The example following this statement shows both
> forward and backward directives! Clarify the original statement by
> adding the words "at a given level".
>
> The same exclusivity statement needs to be made about position and
> backward at any given level.

accepted.

> 10.Under 4.4

>
> a) int_curr_symbol - the definition states that it is the international
> currency symbol. This is not true. It is not the international
> currency symbol (x00a4) but is the string representing the ISO 4217
> code etc. I know POSIX mis-named it. I'd like to see it corrected
> but barring that, at least the definition should be correct.

accepted, the definition will be corrected.

> b) duo_*: these entries will mean that changes are required in
> the localedef compiler utility and programmers will have to be
> aware of the change on strfmon(). A better method to handle the
> dual currency requirement is through the use of the @modifier
> construct. That ensures that no modifications are necessary to
> the current localedef utility, the strfmon() function does not
> have to change and programmers do not have to worry about learning
> how to handle dual currency.
>
> My suggestion is to incorporate the @modifier construct and
> scrap the duo_* keywords from here. The @modifier can also be used
> to invoke different behaviour for the other LC_* categories.

Rejected.

The model of internationalization adopted by ISO/IEC in TR 11017 says that all cultural dependent data should be specified in the FDCC-set and with the same API, the application should be able to obtain culturally correct results for each culture. Shifting the FDCC-set using first the original FDCC-set and then the modified one and then shifting back to the original FDCC-set does not follow this model. The functionality proposed in WG20 addressing what strfmon() addresses, does follow this model.

Localedef needs to be changed anyway to accommodate the 14652 standard.

> c) the uno_valid_* and duo_valid_* entries do not belong in here and
> should be removed.

Rejected, see response to b) above.

> d) conversion_rate: since currency rates fluctuate by the second, this
> should be removed from here.

Rejected: The scheme proposed does not lead to cultural neutral applications. The currency rate is fixed for the cases covered, and for those not, the FDCC-set could be a reference to a very fluctuating specification, possibly available over the network. Implementing 14652 means changes to localedef anyway, even in the LC_MONETARY category.

>
> 11.Under 4.6
>
> a) add some intro text before diving into the keywords. Perhaps "The
> LC_TIME category defines the rules....."etc.

accepted.

> b) abday - the words "calendar systems" need to be removed from the
> first sentence because no other calendar systems are defined in
> this document. Once other systems are defined this sentence
> can be re-surrected.

Rejected. Calendar systems with for example 10 days a week can be specified with the week keyword.

> c) abday - replace the second sentence with "The length of the week
> is defined by the "week" keyword". See later comments as to why
> this suggestion is made.

Rejected. See response to other comments to CA-11

> d) abday - the default Sunday should be "Sun" and the default Monday
> should be "Mon" as they are supposed to be abbreviations.
>

Rejected. This is not the case. The default is not "Sun", but "1" - There is no language dependent defaults in the standard. The "Sunday" is here used to describe the weekday name in the language that the standard is written in, namely English.

> e) day - same comments as in (b) and (c) apply here as well.
>
> f) week - we should only attribute one entity to this and not try and
> overload it with many things. This one should only contain the
> number of days in the week. It should also be renamed to
> "number_of_days_in_week"; I don't think that we have to continue
> with the original limitations and directions in POSIX w.r.t
> keyword names.

Rejected. The current specification is as approved by the working group and changing it does not add any functionality.

> g) week - remove all references to the first weekday in this keyword
> because this information is already carried in both "day" and
> "abday" keywords.

Rejected. the "week" keyword defines which day is the first.

> h) week - have a separate keyword ("first_week_of_year") to designate

> what constitutes the first week of the year.

Rejected. see response to f) above.

> i) abmon - replace "(January)" with "(Jan)".

same response as d)

> j) just as we added a "number_of_days_in_week" (= "week" in this
> document) keyword, we should also introduce a "number_of_months_in
> year" keyword.

Rejected. see response to f) above.

>

> k) first_weekday - perhaps we should call this
> "first_weekday_in_calendar_layout" because as it stands it could
> also apply to the first workday of month or year.

rejected. It is explained in the standard

> l) first_workday - perhaps we should call it "first_workday_of_week"
> because as it stands it could also apply to the first workday of
> month or year.

rejected. It is explained in the standard

> m) cal_direction - perhaps it is better to call this "calendar_layout".
> The definition should also be improved because "left-right from
> top" etc. is not adequate. Does this mean that the months run this
> way or that the weekday titles run this way or what?
>
> n) <std> and <dst> - the restriction of >3 and <10 characters is
> arbitrary, not culturally acceptable, and should be removed.

Rejected. This is industry standard, and the minimum 3 characters
is inherited from POSIX.

> o) <rule> - this does provide for those cases where the change to/from
> summer time is by a yearly decree and can therefore vary. We should
> make a provision for this.

accepted in principle. There is already a <year> attribute to this effect.

> p) M<m>.<n>.<d> - the statement "(0<= d<=7)" is incorrect because this
> means that one can have 8 days in the week!

Rejected. You may have more than 7 days in a week. The idea was to allow
"0" to also mean the last day of the week.

> q) M<m>.<n>.<d> - cannot designate both day 0 and day 7 to be Sunday;
> it should only be one of these.

Rejected. See response to p) above

>

> 12.Under 4.6

>

> a) Table 2: "%n - A <newline character>" does not belong here.

Rejected. It follows industry standards such as specified by X/Open.

> 13.Under 4.6

>

> There is a need to explain what is meant by "extended regular
> expressions".

Accepted.

> 14.Under 4.8

>

> This section need to talk about paper sizes in terms of what users
> are used to. Most photocopiers will take about A4 or letter or legal
> etc. size paper; same with printers. These common terms should be
> allowed here.

Rejected. Then a number of culturally dependent specifications
need to be specified here. The cultural specification format should
be culturally neutral.

> a) height - why the restriction for this to be in millimeters only?
> Why not have inches as well?

Rejected. Which inches? US or UK or Danish or Swedish... Using non-ISO
culturally dependent measures is not practical.

> b) width - same as above.

See response to a) above.

> 15.Under 4.9

>

> In terms of salutations, the set does not include profession/status
> salutations such as Doctor (Dr.) etc. Also, in some cultures both
> a full and an abbreviated salutation (for example Doctor and Dr. as
> above) are used.

There is a %p for profession. %s can also be used for "Dr."
there may be a need for yet another format effector.

> 16.Under 4.9

>

> What is CEPT-MAILCODE?

It is a standard for codes for countries, used for postal mail.
a reference will be added.

> Items from "country_ab2" to "lang_lib" does not appear to belong in
> this LC_ADDRESS section. For example, what has "country_car" got to
> do with a postal address?

The category also caters for other addresses than postal addresses.

> 17.Under 4.12

>

- > What does "other" mean in a measurement system?
- > Assume that "U.S.A measurement" means the "Imperial System".
- > Note that LC_PAPER should follow this standard and allow for
- > expression in measurement systems other than metric.

The LC_MEASUREMENT category will be removed.

- > 18.Under 4.13
- >
- > What is the rationale for including this here? This type of information
- > should not be mandatory and really belongs in header comments. For
- > example, the contact info etc. can and should only exist in header
- > comments and not as mandatory keywords. About the only thing that
- > we should discuss putting in here is the version and revision number.

Rejected. In this way the information may be obtained by the application.

- > 19.Finally, syntax should be added for the "order_start" statement of
- > LC_COLLATE to allow either conditional IGNORE of the first 3 levels
- > for special characters (as is the case now), or taking them into
- > consideration, using a toggle, to eventually allow Unicode/Java
- > ordering specs to be made compatible with 14651 (14651 would then
- > e able to be either tailored in consequence, or the template modified
- > to reflect Java tables at once).

Rejected. This is specific to some specific application.

- > _____ end of Canada comments; beginning of Denmark comments _____
- >
- > **Danish comments on FCD 14652.**
- >
- > DS votes "Yes" with comments on FCD 14652.
- >
- >
- > Technical comments:
- >
- > dk.t.1 In LC_MONETARY the uno/duo specification could be
- > expanded to handle more than one transition, like
- >
- > int_curr_symbol "BRE ";"BRR ";"BRL "
- > valid "-YYYYMMDD";"YYYYMMDD-YYYYMMDD";"YYYYMMDD-
- > conversion_rate 1/100;1/1000

accepted.

- > dk.t.2 It should be said that conversion_rate is optional.
- > The default value should be 100.

accepted.

- > dk.t.3 Doubling escape characters should be avoided in 5.1.

accepted.

- > dk.t.4 The format effectors of the date specification should
- > be checked and aligned with POSIX and Open Group specifications

accepted.

> dk.t.5 There should be examples on tosymmetric and map.

accepted.

> dk.t.6 LC_VERSIONS first parameter of "category" should be
> enclosed in double-quotes as a proper string.

accepted.

> dk.t.7 We would like to see functionality for paper margins,
> terminology, spelling and hyphenation in the standard,
>
> This could be done by:
>
> A category LC_MARGINS with keywords top bottom left and right
> with specifications in millimeters.
>
> A category LC_SPELLING with a list of words.
>
> A category LC_HYPHEN with a list of words and SOFT HYPHEN
> indicating the hyphenation possibilities. This may
> be combined with LC_SPELLING
>
> A category LC_TERMS with a list of words and relation
> to a common term reference, for example that of ISO/IEC 2382.

Rejected. The specification is immature. This could be done
in an amendment to the standard.

> dk.t.8 The scope (1.) should be extended to cover computer use of
> the specifications, as this is an information technology standard.

accepted.

>
> Editorial comments.
> dk.e.1 Symbolic ellipsis <j0148>..<>j1053> should be ... in 3.2.2
> as they are decimal.
>

accepted.

> dk.e.2 There is a typo in A.2, point 6:
>
> int_p-sep_by_space
> should be
> int_p_sep_by_space

accepted.

> dk.e.3 Strings with more than one character should be
> enclosed in double-quotes. Examples are in
> collating-symbol and transliteration examples.

accepted.

> _____ end of Denmark comments; beginning of Israel comments _____

>
> **Comments from the Israel National Body accompanying a**
> **negative vote on SC22 Letter Ballot N2638**

>
> SII comments for ISO/IEC FCD 14652:
>
> The standard cannot be approved by us unless the Bidi section undergoes
> extensive revision.
>
> Even with this revision, it may be bound by some Bidi specification from
> X/Open which I have not seen, and which may be acceptable or not.
>
> A. In section 4.2.1 "Basic keywords", definition of "class". I assume
> that the authors wish to be synchronized with the concepts of the Bidi
> algorithm in Unicode (and if not, this is IMHO a major flaw). If so, the
> explanation for "num terminator" is wrong probably due to the misleading
> term used by Unicode. In fact, the intended meaning in Unicode is rather
> prefix/suffix to numbers, like a leading or trailing sign. I suggest the
> definition: "characters which may be adjuncted before or after the digits
> of a number."

The BIDI specifications will be removed, due to immaturity of
the specifications, but may be added in an amendment.

> B. In section 4.2.1 "Basic keywords", definition of "class". I suggest
> to change the definition of "num separator" to: "number separator
> characters which can appear between digits of numbers written with any of
> the characters in the digit class." This formulation makes it clearer
> that the number separators do not segregate between numbers, but appear
> between parts of the same number.

The num_separator goes away, due to removal of BIDI support.

> C. In section 4.2.1 "Basic keywords", definition of "map", explanation
> for tosymmetric says: "for each pair also the mapping from the second
> operand to the first operand is also defined." It is not clear what the
> first "also" refers to. And it is not clear "also defined" by who? I
> suggest the following reformulation: "For each pair, the mapping from the
> second operand to the first operand is also implied."

accepted.

> D. Section 4.2.3 "il8n LC_CTYPE category", classes "right_to_left",
> "num_terminator", "num_separator", etc., which are related to Bidi: These
> classes are similar to classes defined in Unicode, but not identical.
> There are classes defined here but not in Unicode, which is perfectly
> o.k. There are classes defined in Unicode but not here, which I see as a
> problem. A big omission is the "left-to-right" class, although it is
> mentioned in section 4.2.1 of this standard. Even for those classes
> which are common in both standards, the content of the classes is much
> different.

>
> I assume that the authors wish to keep in sync with the classification
> in the Unicode standard. This is far from true in this version of 14652.
> This classification thing is a big issue. The unicode experts have spent
> much time on it and this work is still ongoing. This standard does too

> much or too little about it, with such blatant errors as classifying
> Eastern Arabic-Indic digits (U06F0 to U06F9) as right-to-left instead of
> digits. If this standard cannot just refer to the Unicode
> classification, it should "left" the classification lists from Unicode.
> Trying to do it again by itself is a waste of time and is like to give
> results much worse than what is in Unicode because not enough efforts
> will be invested. This is a question of principle, so I will not discuss
> in detail what I see as errors in the classifications of "i18n".

Bidi support will be removed.

> _____ end of Israel comments; beginning of Japan comments _____
>

> Japan:

>
> SC 22 N 2638: FCD 14652 - Specifications for Cultural Convention
>

> (X) Disapproval of the draft for reasons below
>

>
> National Body: Japan
> Date: 1998-06-02
> Signature: KATSUHIKO KAKEHI
>

> -----
>
> Japan disapproves FCD 14652 (SC22 N 2638) with following comments.
>

>
> J-01) Project objective:
> The practical value of this FCD is nothing more than POSIX. Japan suggested
> some examples for cultural conventions which are not in POSIX, and they are
> now added to the document. But these features are not designed according to
> real requirements. The new international standard should be developed when
> real market needs is confirmed. This is the main reason for Japan's
> disapproval, which does not seem to be reasonably resolved, unless such need
> is reported.

The standard goes beyond POSIX in many places, as documented in annex A, and as POSIX has stalled internationalization work, this standard is the only project enhancing internationalization specifications in JTC 1. There are a number of market-based requirements addressed in the standard, such as the support for the Euro currency, and 10646 support, and a number of issues that are response to member body requests, to address national requirements. In many cases vendors do not have detailed knowledge of specific cultures and thus they do not implement fully culturally adopted products, so the market does not solely provide an adequate means to honour national requirements. A number of the new specifications (LC_ADDRESS, LC_NAME, LC_TELEPHONE) is based on information that partly comes from the Japanese member body.

JTC 1 is giving a strong focus on internationalization by creating a new technical direction on cultural adaptability, and ensuring that all IT standards that are started will take internationalization into account, if applicable.

The standard is being investigated for implementation by a number of industry sources, including GNU gcc, Unisys and SUN, indicating that that there is a market need for it.

>
> J-02) Title:
> Title should be changed to:
>
> Specification method for Cultural Conventions
>
> reflecting the agreed content of the clause one.

Accepted.

> J-03) FOREWORD:
>
> The paragraph in FOREWORD
>
> The Standard uses text from ISO/IEC 9945-2:1993 "Information
> Technology - Portable Operating System Interface (POSIX) -
> Part 2: Shell and Utilities", primarily clauses 2.4 and 2.5.
> The major differences from this text is listed in annex A.
>
> will lead readers to think ISO/IEC 14652 is a minor modification of a
> small part of POSIX.
>
> If this FCD proves to be something more than POSIX, the paragraph should
> be changed to
>
> The Standard extends the concept of the locale specifications
> defined primarily in
> subclauses 2.4 and 2.5 of ISO/IEC 9945-2:1993 "Information
> Technology - Portable Operating System Interface (POSIX) -
> Part 2: Shell and Utilities".
> The major extensions from the locale specification are
> listed in annex A.

Accepted

>
> J-04) 1. Scope
>
> Add a note
>
> NOTE) The term "description" means that this standard defines
> a human readable format -- not a machine processable format
> used for automatic installation of systems.
>
> Rationale: Scope should state that this intentional standard specifies
> the specification method for in "paper form" clearly. Unless, there is
> very high possibility of mis-application of this standard.

accepted in principle. The standard is "paper based" but is written to also be machine applicable. This should clearly be stated in the standard. The way of using it with computers will also be specified.

>

> J-05) 2. Normative references:
>
> Add the references to
>
> ISO 639 Code for the representation of names of languages
> ISO 3166 Code for the representation of names of countries
>
> if they remains to be referred (*1) in 4.10 LC_ADDRESS etc.
>
> *1) the references will be removed by other comments dispositions.
>
Accepted.

> J-06) 3.1.5 cultural convention:
>
> The definition
>
> A data item for computer use that may vary
> dependent on language, territory, or other cultural circumstances
>
> should be changed to
>
> A data item for information technology that may vary
> depending on language, territory, or other cultural circumstances
>
> because the expression "computer use" suggests "machine processable data
> items" which are out of the scope of this FCD.

Response: accepted. However, the standard is a JTC 1 standard and is thus
for use with information technology, which is the use of computers.

> J-07) 3.1.7 charmap:
>
> The definition
> A definition of a mapping between symbolic character
> names and the encoding for a coded character set
> should be changed to
> A definition of a mapping between symbolic character names and
> character codes.

Accepted.

>
> J-08) 3.2.1 Format of syntax descriptions
>
> The first sentence of this subclause is incomplete and the second sentence
> is not understandable because the term "format" appears suddenly and there
> is no "format string enclosed in double quotes."
>
> Even if the expression
> "<format>",[<arg1>,<arg2>,...,<argn>]
> is inserted after the first sentence, the contents of this subclause
> is still incomplete, because many explanations in 2.12 of POSIX.2 are
> omitted here.
>
> The new text should be
>
> 3.2.1 notation for defining syntax

>
> In this standard, the description of an individual record in
> FDCC sets is done using the syntax notation defined in 2.12 of
> ISO/IEC 9945-2. The rest of this subclause is the short tutorial
> of the syntax notation.

>
> The syntax notation looks as follows:

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

>
> It is similar to that used by the C-language printf() function
> and the *format* string enclosed in double quotes may contain
> some conversion specifications such as

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

> and some escape sequences

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

Accepted

>
> J-09) References to the syntax notation defined in 3.2.1:

>
> There are two types of expressions in referring to the syntax notation
> after 3.2.1 as follows:

> 1) the expressions using the term "syntax" such as in

> The "translit_start" keyword may be followed by transliteration
> statements. The syntax for a transliteration statement is:

> "%s %s;%s;...;%s\n", <transliteration-source>, ...

> 2) the expressions using the term "format" such as in

> It shall have the following format, starting in column 1:

> "charmap %s\n", <charmap>

> It is not recommended to use many expressions for one thing in one
> standard document and the latter type is wrong because 3.2.1 defines the
> syntax and not the format. The expressions of the latter type should be
> changed to the former type.

Accepted.

>
> J-10) 3.2.3 Ellipsis

>
> This subclause should be removed because

> 1) the definition of the ellipses used in collation statements in 4.3.1

> conflicts with the one defined here,
>
> 2) the usage of the "... " in the syntax notation defined in 3.2.1
> conflicts with the one defined here,
>
> 3) the definition here is too simple compared to the definitions for
> ellipses "...", ".." and "...." used in charmap (5.1),
>
> Related action) Define the usage of three kinds of ellipses in 4.2
> LC_CTYPE in the same way as in 5.1,

accepted in principle. The clauses will be reworded to be correct,
and if no generality can be achieved, it will be removed.

>
> J-11) 4. FDCC set, 2nd to last paragraph 3rd line:
>
> Current text: "LC_X_" which use is application defined.
> Change to: "LC_X_" which shall not be used for future addition of
> categories specified in this international standard.
> Those may be used for application defined categories.

Accepted.

>
> J-12) 4.1.1 Character representation:
>
> Add a new rule for UCS-notation, <Uxxxx> and <UXXXXXXXX> which looks
> like symbolic names but may not be not defined in a charmap file. The
> text in 4.1.1
>
> (1) ... Repertoiremaps have predefined symbolic names
> for UCS characters.
>
> does not cover the case where a FDCC-set does not contain repertoiremap
> statement and the first sentence of this subclause
>
> Individual characters, characters in strings, and collating
> elements shall be represented using symbolic names, UCS notation
> or characters themselves, or as octal, hexadecimal, or decimal
> constants as defined below.
>
> requires a rule for UCS notation.
>

accepted in principle. A FDCC-set may use a repertoiremap without
having it defined. Wordings on use on UCS notation will be added.

>
> J-13) 4.1.2.1 comment_char:
>
> The requirement
>
> and the remainder of a line with a <comment char> occurring
> where a syntactic semicolon may occur, shall be ignored
>
> stated here contradicts with the requirement
>
> A line in a specification can be continued by placing an escape

> character as the last visible graphic character on the line
>
> stated in 3.2.2.
>
> The comment not beginning from the first character should not be used.

Rejected. This is requested by experts of other NBs during the development of the standard.

The standard says that comment lines can not be continued with the escape character at the end of the line.

>
> J-14) 4.1.2.3 repertoiremap:
>
> The sentence
>
> The following line in a FDCC-set specifies the name of a
> repertoiremap used to define the symbolic character names
> in the FDCC-set
>
> is meaningless because there is no naming facility for the repertoiremap
> in this standard.

Rejected. Repertoiremaps may be named, even on paper.

> J-15) 4.1.2.4 charmap
>
> The sentence
>
> The following line in a FDCC-set specifies the name of a
> charmap which may be used with the FDCC-set
>
> is meaningless because there is no naming facility for the repertoiremap
> in this standard.

Rejected. (understanding the comment addresses charmaps). charmaps may be named, even on paper.

> J-16) 4.1.2.4 charmap:
>
> The sentence
>
> For the actual use of a FDCC-set, at most one charmap may be in
> use, and this may be different from any charmap specified with the
> "charmap" line.
>
> should be changed to
>
> At most one charmap shall be specified in an FDCC-set.

Rejected. A FDCC-set may be used with more than one charmap, this is one of the fundamental design principles of the FDCC-set, that it is coded character set independent, and thus is designed to be used with a number of charmaps.

>
> J-17) 4.2.1 Basic keywords:
>

> The first sentence
> The following keywords shall be defined
> should be changed to
> The following keywords shall be recognized in this standard.

Accepted.

>
> J-18) 4.2.1 Basic keywords:
>
> The expressions
>
> The keyword may be omitted
>
> and
>
> This keyword is optional
>
> in the definitions of keywords may lead readers to think the statement
> containing such a keyword e.g.
>
> class "num_terminator";<:>;<space>
>
> is replaceable with the statement not containing the keyword e.g.
>
> "num_terminator";<:>;<space>
>
> Those expressions should be changed to
>
> This keyword may not be specified
>
> which makes clear contrast to the expression
>
> The keyword shall be specified.
>

accepted in principle.

> J-19) 4.2.1 Basic keywords, "outdigit":
>
> The rationale for adding keyword "outdigit" is not understandable because
> only a short phrase "for output" is added to "digit" and adding this
> keyword is not referred in the disposition of the comments on the first
> CD.
>
> This keyword should be removed.

Rejected, the outdigit class is used for the output of digits, as explained also in response to canadian comment CA4-b).

This is needed for output of numbers such as integers or floating point. There is a need to specify which specific set of digits that is used for output, as we cannot assume that the (normal) arabic digits always will be used. For example in Arabic countries they will use their sepcific set of digits, and in various Indic regions various other sets of digits will be used. The class "digit" does not address which set of digits to be used for output, only what is recognised for input,

and "digit" can specify several sets.

> J-20) 4.2.1 Basic keywords, "class":

>
> class Define characters to be classified as characters in the
> class defined with the first operand, which is a string. The string
> shall only contain letters, digits and <hyphen-minus> and
> <underline> from the portable character set.

> The definition of "string" is incomplete because

- >
> 1) the definition of "letter" is not given in this standard,
>
> 2) the definition of "the portable character set" is not given
> at this point(*1)
>
> 3) even if the use of the portable character set becomes authorized,
> <underline>(*2) is not defined anywhere.

Accepted. reference to portable character set will be added (5.1)

Letter will be changed to alpha. underline will be added.

>
>
> *1) The sentence in the first paragraph of 4.2 LC_CTYPE
>
> Support for the portable character set is required
>
> only defines the use of the portable character set in LC_CTYPE
> and does not explain the use of the portable character set
> in the standard.
>
> The use of the portable character set should be mentioned in 3.2
> (not in 4.1.1 as was suggested in the US comments on the first CD)
> because it is a part of description of this standard and
> not a part of the FDCC-set definition.

Accepted.

> *2) If <underline> means '_', it is confusing with the expression
> with the five letters "LC_X_"
> in Clause 4 because it says '_' is a letter.

> J-21) 4.2.1 Basic keywords, "class":

>
> The defined classes "num_separator" and "num_terminator" may cause confusion
> with definitions in LC_NUMERIC. The relation should be clarified.

Accepted.

> J-22) 4.2.2 Character string transliteration

>
> This subclause should be removed because it is based on a misconception on
> the relation between FDCC-set and languages -- for example the following
> sentence
>
> Transliteration is often language dependent, and the language to be
> transliterated to is identified with the FDCC-set, which may also

> be used to identify a specific language to be transliterated from.
>
> clearly states the wrong start point.
>
> The concept of character string transformation as an element in a FDCC-set
> is not mature yet.

Rejected. The specification is stable, and implemented. It addresses a well-defined
need, within a sound model of i18n.

>
>
> J-23) 4.2.2.2 "include" keyword
>
> The name is very confusing with the other uses of "include" in information
> technology. It should be renamed, e.g. "translit-origin", even if subclause
> 4.2.2 remains.

rejected. The "include" keyword closely follows the semantics of other "include"
statements defined in other IT standards.

>
> J-24) 4.2.3 "i18n" LC_CTYPE category:
>
> This subclause should be removed because it is too early to define the
> default of character classification for all characters in UCS.

Rejected. This is a stable definition.

> J-25) 4.2.3 "i18n" LC_CTYPE category:
>
> The criterion for defining this category should be clarified. For example,
> it is not clear why some characters are declared as "alpha" and others
> are not.

Accepted

>
> J-26) 4.2.3 "i18n" LC_CTYPE category, "upper" and "lower":
>
> This part of the definition is too difficult to be checked by human readers.
>
> It should be modified by
>
> 1) introducing a notation, which is used only in these two keywords,
> such as
> <U0102>..(2)..<U010E>
> standing for
> <U0102>;<U0104>;<U0106>;<U0108>;<U010A>;<U010C>;<U010E>;
> to simplify the sequences with incremental two,
>
> 2) comment lines should be added for readability
>
> See Annex 1.

Accepted

>
> J-27) 4.2.3 "i18n" LC_CTYPE category, "upper" and "lower":
>

> The reason for omitting COPTIC CAPITAL and SMALL letters in Table 10 of
> UCS should be explained.

Accepted.

>
> J-28) 4.2.3 "i18n" LC_CTYPE category, "upper" and "lower":
>
> It should be investigated whether GEORGIAN should be treated as upper/lower
> schemes or not.

accepted

>
> J-29) 4.2.3 "i18n" LC_CTYPE category, "alpha":
>
> The description here for "alpha" is too difficult to be checked by human
> readers.
> It should be modified by
>
> - removing the characters belonging to "upper" or "lower",
>
> - adding comment lines.
>
> See Annex 2.

Accepted

>
> J-30) 4.2.3 "i18n" LC_CTYPE category, "alpha":
>
> Add the character <U3094> to alpha if the category intends to be something
> other than Annex of TR 10176.

The alpha character calls intends to be the same as annex A of 10176

>
> J-31) 4.2.3 "i18n" LC_CTYPE category, "digit":
>
> The CJK characters which may semantically be grouped as numerals
> <U4E00>;<U4E8C>;<U4E09>;<U56DB>;<U4E94>;/
> <U516D>;<U4E03>;<U516B>;<U4E5D>
> should not be handled as digits.

accepted.

> J-32) "copy" in 4.2.1, 4.3.2 etc.:
>
> The keyword "copy" should be removed from all categories or should be
> regarded as POSIX-specific one if this standard claims to be upward
> compatibility to POSIX. The keyword "copy" in POSIX assumes that a
> locale other than the implementation-supplied one may come into existence
> after the execution of the utility "localedef" and there is no
> corresponding mechanism for FDCC-sets.
>
> NOTE 1 Related Action) The sentence in 4.1 FDCC-set Definition
> A category source definition shall contain either
> the definition of a category or a copy directive.
> should be changed to
> A category source definition shall contain either
> the definition of a category.

>
> NOTE 2) If there are strong needs to define a FDCC-set inheriting
> the definitions from some other FDCC-sets, a new keyword, say
> "see_attachment" may be introduced with a syntax
> "see_attachment %s\n", <name_in_referred_FDCC-set>
> or
> "see_attachment %d\n", <attachment_number_of_referred_FDCC-set>
> which refers the corresponding category definition from the
> specified FDCC-set attached to the current FDCC-set.
>
Rejected. The standard is intended for use with an utility
like localedef.

> J-33) 4.3.1 Collation statements:

>
> The following lines
>
> The "order_start" and "replace-after" keyword shall be followed
> by collating statements. The syntax for the collating statements
> is

> "%s %s;%s;...;%s\n",<collating-element>,<weight>,<weight>,...

> Each collating-element shall consist of either a character ...

> should be changed to

>
> The "order_start" and "replace-after" keyword shall be followed
> by collating statements. The syntax for the collating statements
> is

> "%s %s;%s;...;%s\n",<collating-identifier>,<weight>,<weight>,...

> Each <collating-identifier> shall consist of either a character ...

Accepted. See also US editorial comment 14

> J-34) 4.3.8 "order_start" keywords:

>
> Give a definition of "substring" and add a sentence
> The direction of scanning substrings is towards the logical end of
> the string.
> to the explanation of the directives "forward" and "backward".

Accepted

> J-35) 4.3.14.6 "else" keyword

> The sentence

> If the preceding block of statements were not used, the statements
> are used, otherwise they are ignored

> should be changed to

> If no preceding "ifdef", "ifndef" or "elif" statement has been used,

> the statements are used, otherwise they are ignored.

Accepted.

>
> J-36) 4.4 LC_MONETARY:
>
> Change
>
> uno_valid_from an integer representing a Gregorian date
> in the form YYYYMMDD,
>
> to
>
> uno_valid_from a digit string representing a Gregorian date
> in the form YYYYMMDD,

Rejected. There is no definition of a "digit string", and the integer represent the value adequately..

> J-37) 4.4 LC_MONETARY
>
> The "i18n" FDCC-set is for the LC_MONETARY category should be removed
> because there is no internationally accepted value for the keyword
> "mon_decimal_point" which shall be specified.

Rejected. The "mon_decimal_point" follows ISO rules for specifying numbers. There is a need for default values for all categories.

> J-38) 4.5 LC_NUMERIC
>
> The "i18n" FDCC-set is for the LC_NUMERIC category should be removed
> because there is no internationally accepted value for the keyword
> "decimal_point" which shall be specified.

Rejected. The "decimal_point" follows ISO rules for specifying numbers. There is a need for default values for all categories.

> J-39) 4.6 LC_TIME
>
> Add the following paragraph at the beginning of this subclause:
>
> The LC_TIME category defines the rules and symbols that shall be
> used to format date and time information based on ISO 8601 or its
> variant with a different starting year e.g. the Era system in
> Japan (JIS X 0301). The exceptions are the descriptors %c, %x and
> %X.
>
> NOTE: The support for date and time information greatly
> apart from ISO 8601 is an emergent matter and it is
> expected to amend this standard as soon as possible
> if such date and time systems are authorized from the
> view point of information technology.
>
> RATIONALE: It is of no use to do unsystematic adaptation, such as allowing
> 13 Hebrew months without its algorithm being explicit.

Rejected. This 13 hebrew months are as requested from other member bodies.

> J-40) 4.6 LC_TIME

>

> The concept of "timezone" and "summer time" should be separated.

Accepted in principle. The concepts are reasonably separated, and the specification is based on industry practise.

> J-41) 4.6.1 Date Field Descriptors:

>

> 1) The function of the escape sequence %f is the same as that of %u in POSIX which is missing in this table. It should be renamed.

>

> 2) The escape sequence %V, %Ou, and %OV in POSIX are missing. It should be defined here.

>

> 3) Change %Z to %z as in POSIX.

>

> 4) Change %u to other some other value to keep compatibility with POSIX.

accepted.

>

> J-42) 4.6.2 Modified Field Descriptors:

>

> The value

> d_t_fmt "<%><a><SP><%><F><SP><%><T>" -- 2 1997-10-07 10:00:01

> should be changed to

> d_t_fmt "<%><F><(><%><a><)><SP><%><T>" -- 1997-10-07(2) 10:00:01

> because to write an abbreviated weekday name just after the day number is logical and recommended as an international default compared to some local existing practice of weekday first.

Accepted in principle. The day number will be removed..

> J-43) 4.8 LC_PAPER

>

> 1) Change the sentence

>

> The LC_PAPER category defines the paper size.

>

> to

>

> The LC_PAPER category defines the default size of paper used for documents.

>

> 2) Change from

>

> height Shall be used to specify the height of the paper. ...

>

> to

>

> height Shall be used to specify the vertical dimension of the paper ...

>

> 3) Change from

>
> width Shall be used to specify the width of the paper
>
> to
>
> width Shall be used to specify the horizontal dimension of the
> paper ...
>
> 4) Add a note
>
> NOTE) if the height is greater than the width, it is called
> to be in portrait position, else it is called to be in landscape
> position.

accepted.

>
> J-44) 4.9 LC_NAME
>
> Add a note after the first sentence of this subclause as follows:
>
> NOTE: There are a number of variations for addressing a person
> among the cultures. Middle names are not used in many countries
> and even the family names are not used in some countries.
> The specification below should be regarded as a start point for
> this problem.

Accepted

>
> J-45) 4.9 LC_NAME, "name_gen"
>
> 1) Change the sentence for name_gen from
> The operand is a string defining a salutation valid for all
> persons,
>
> example: the Japanese "-san" salutation
> to
> The operand is a string defining a salutation valid for all
> persons,
>
> example: the Japanese "-sama" salutation in a letter
>
> 2) Reorder the keyword "name_ms" before "name_mrs" in a general-salutation-
> first convention.

Accepted.

>
> J-46) 4.10 LC_ADDRESS
>
> It is questionable to define this category because addressing schemes
> differ from country to country and the current draft, which looks
> street-oriented way, is not applicable to other systems -- e.g.
> block-oriented addressing in Japan.
>
> This subclause should be removed

Partly accepted . - Changes will be made to clarify that the street
address can also be used with Japanese block numbers.

> J-47) 4.10 LC_ADDRESS
>
> The first sentence of this subclause should be changed from
>
> The LC_ADDRESS category defines formats to be used in
> addressing a person, e.g. in a postal address or in a letter, and
> other items of geographic nature
>
> to
>
> The LC_ADDRESS category defines formats to be used in
> specifying location of a person's living or office used
> in a postal address or in a letter.

Accepted.

>
> J-48) 4.10 LC_ADDRESS
>
> It is questionable to define this category because addressing schemes
> differ from country to country and the current draft, which looks
> street-oriented way, is not applicable to other systems -- e.g.
> block-oriented addressing in Japan.
>
> This subclause should be removed

This seems to be the same comment as J-46.

> J-49) 4.10 LC_ADDRESS
>
> Add a note after the first sentence of this subclause as follows:
>
> NOTE: There are a number of variations for specifying location
> of a person's living or office.
> among the cultures. Middle names are not used in many countries
> and even the family names are not used in some countries.
> The specification below should be regarded as a start point for
> this problem.

Accepted in principle.

>
> J-50) 4.11 LC_TELEPHONE
>
> Add an escape sequence
> %c alternative carrier service code used for dialing abroad

Accepted.

>
> J-51) 4.12 LC_MEASUREMENT
>
> 1) This subclause should be removed because it is useless to declare a
> measurement system generally and the unit of measurement varies greatly
> even in one culture in contrast to MONETARY or DATE representation.

Accepted.

>
> J-52) 4.12 LC_MEASUREMENT

> existing FDCC-set to be used as the source for the
> definition of this category.
> </OLD>

Rejected. The version can be included in the name, if wanted.

> 4) The keyword "language" should be removed or changed to
>
> language Natural languages used as comments in this
> FDCC-set

Rejected. The natural language is a significant part for the identification of the FDCC-set, as also described in TR 11017 and ISO/IEC 15897.

> 5) The keyword "territory" should be removed or changed to
>
> territory The geographic extent where this FDCC-set serves
> (need not be a national extent)

Accepted in principle as changed, with "applies" substituting "serves".

>
> J-54) 5.1 Character Set Description Text
>
> The declarations <escseq>, <addset> and <include> should be removed.
>
> RATIONALE:
>
> 1) The FDCC-set is a human readable document and needs no consideration
> for encoding,
>
> 2) The charmap, which maps symbolic names to specific code values,
> should be regarded as a old tools for keeping upward compatibility for
> POSIX locales and should not be augmented.
>
> The linkage of symbolic character names to a code system based on ISO
> 2022 environment is a local and/or implementation matter outside of the
> cultural convention.

Rejected. The encoding of characters are a cultural element. For example in Denmark it is the cultural convention to employ a specific set of characters, and the encoding, possibly using 2022 techniques is also a specific cultural convention. The charmaps are necessary for making the FDCC-sets function in an IT environment.

>
> J-55) 6. REPERTOIREMAP:
>
> To define the symbolic character names by using the ISO/IEC 10646 code
> position as stated in the paragraph
>
> The repertoire mapping is defined by specifying the symbolic
> character name and the ISO/IEC 10646 code position in

> hexadecimal form (with a preceding 'U') and optionally the
> long ISO/IEC 10646 character name in the following format:
>
> "%s %s %s\n",<symbolic-name>,<10646-codepoint>,<comments>
>
> makes FDCC-sets unstable because the meaning assigned to the ISO/IEC
> 10646 code position depends the version of the standard.
>
> Instead of the definition by code position, the identifiers provided by
> SC2, which look like code positions but guaranteed for their independence
> from version-up, should be used.
>
> The whole text in this clause needs review by SC2 experts.

Accepted in principle. 10646-codepoint changed to 10646 short identifier in the form XXXX or XXXXXXXX.

>
>
> J-56) Clause 6. Repertoiremap:
>
> Do not use specific mnemonics to specify "i18n" repertoiremap.
> Whatever wording is used, this description may give an user of this
> standard
>
> an impression of "this mnemonics is normative".
> The mnemonics project proposal was rejected at SC22 WG20 long time ago,
> so, to sneak in the rejected proposal into JTC1 standard should not be
> done.
>
> As was pointed out in the previous US comments. this list is arbitrarily
> chosen, and the principles for characters in it are unstated. If the
> repertoire file is not going to correspond to one of the named and
> numbered subsets of ISO/IEC 10646 (and Subset 300, the BMP, would be the
> obvious choice), then the choice of characters in the repertoire file
> *must* be justified in 14652.
>
> If the intention is, rather, to just define a bunch of short mnemonics,
> then most of this entire listing is useless and should be omitted.
> Introducing mnemonics such as <c*> for GREEK SMALL LETTER XI and <z%>
> for CYRILLIC SMALL LETTER ZHE and <K%> for HEBREW LETTER FINAL KAF is
> completely confusing. A very small percentage of these mnemonics has
> seen widespread use in plaintext reference to accented characters. The
> rest should be completely abandoned in CD 14652 in favor of use of the
> hexadecimal value as the unique symbolic identifier for a 10646
> characters (e.g. <U0436>).

Rejected. The list of mnemonics builds on existing practise, including POSIX and Internet use.

>
>
> J-57) Clause 7.Conformance:
>
> 1) 7.1 FDCC-set: Change "A FDCC-set" to "A FDCC-set description"
>
> 2) 7.2 FDCC-set category: Change "a category" in the first line to
> " a category description"
>

> 3) 7.2 FDCC-set category: Change "conformance ... can be claimed ...
> against each of the clauses ... " to "conformance ... can be claimed ...
> according to each of the clauses ... "
>
> 4) 7.3 Charmap: Change "A charmap" to "a charmap description"
>
> 5) 7.4 Repertoiremap Change "Repertoiremap" to "Repertoiremap description"
> and add a note:
> note: only description (on paper form in principle) can conform
> this standard directly, and no system, platform, application can
> conform this standard directly.

Accepted in principle, but the scope and conformance will be clearly
described to be paper based, so there is no need to change the
wording here.

>
> J-58) BIBLIOGRAPHY:
>
> Remove the references to
>
> ISO/IEC 8824, "Information technology - Open Systems Interconnection
>
> - Specification of Abstract Syntax Notation One (ASN.1)"
>
> and
>
> ISO/IEC 8825, "Information technology - Open System
> Interconnection - Specification of Basic Encoding Rules for
> Abstract Syntax Notation One (ASN.1)"
>
> because these specifications are not relevant to this standard in any
> sense.

Accepted.

>
>
> J-59) B.1.2 LC_COLLATE Rationale
>
> The paragraph
>
> The Far East (particularly Japanese/Chinese) collations are often
> based on contextual information and pronunciation rules (the same
>
> Such collation, in general, falls outside the desired goal of the
> standard. There are, however, several other collation rules
> (stroke/radical, or "most common pronunciation") which can be
> supported with the mechanism described here. Previous drafts
> contained a substitute statement, which performed a regular
> expression style replacement before string compares. It has been
> withdrawn based on balloter objections that it was not required
> for the types of ordering this standard is aimed at.
>
> should be removed or changed to
>
> In Japan, collations of strings containing CJK characters
> (ideograms) are often done considering some related information

> such as pronunciation which needs a bulk dictionary (and some
> common sense).
> Such collation, in general, falls outside the desired goal of the
> standard. The standard can support only a restricted part of
> collation used in Japan.

Accepted in principle. The text will be modified to reflect to the Japanese conventions, while retaining the information on other East-asian cultures, such as the redical/stroke scheme.

```
>
> -----
> Annex 1 -- Replacement text for "upper" category "
>
> upper /
> % TABLE 1 BASIC LATIN
> <U0041>..<U005A>;
> % TABLE 2 BASIC LATIN
> <U00C0>..<U00D6>;<U00D8>..<U00DE>;/
> % TABLE 3 LATIN EXTENDED-A
> <U0100>..(2)..<U0136>;/
> <U0139>..(2)..<U0147>;/
> <U014A>..(2)..<U0178>;/
> <U0179>..(2)..<U017D>;/
> % TABLE 4 LATIN EXTENDED-B
> <U0181>;<U0182>..(2)..<U0186>;<U0187>;
> <U0189>..<U018B>;<U018E>..<U0191>;<U0193>;<U0194>;/
> <U0196>..<U0198>;<U019C>;<U019D>;<U019F>;/
> <U01A0>..<U01A4>;/
> <U01A7>;<U01A9>;<U01AC>;<U01AE>;<U01AF>;<U01B1>..<U01B3>;/
> <U01B5>;<U01B7>;<U01B8>;<U01BC>;<U01C4>;<U01C5>;<U01C7>;<U01C8>;/
> <U01CA>;<U01CB>;/
> <U01CD>..(2)..<U01DB>;/
> <U01DE>..(2)..<U01EE>;/
> <U01F1>;<U01F2>;<U01F4>;<U01FA>..(2)..<U01FE>
> % TABLE 5 LATIN EXTENDED-B
> <U0200>..(2)..<U0216>;/
> % TABLE 6 IPA EXTENSIONS
> <U0262>;<U026A>;<U0274>;<U0276>;/
> <U0280>;<U0281>;<U028F>;<U0299>;<U029B>;<U029C>;<U029F>;
> % TABLE 9 BASIC GREEK
> <U0386>;<U0388>..<U038A>;<U038C>;<U038E>;<U038F>;<U0391>..<U03A1>;
> <U03A3>..<U03AB>;/
> % TABLE 11 CYRILLIC
> <U0401>..<U040C>;<U040E>..<U042F>;<U0460>..(2)..<U047E>;
> % TABLE 12 CYRILLIC
> <U0480>;<U0490>..(2)..<U04BE>;<U04C1>;<U04C3>;<U04C7>;<U04CB>;/
> <U04D0>..(2)..<U04EA>;<U04EE>..(2)..<U04F4>;<U04F8>;/
> % TABLE 13 ARMENIAN
> <U0531>..<U0556>;
> % TABLE 31 LATIN EXTENDED ADDITIONAL
> <U1E00>..(2)..<U1E7E>;/
> % TABLE 32 LATIN EXTENDED ADDITIONAL
> <U1E80>..(2)..<U1E94>;/
> <U1EA0>..(2)..<U1EF8>;
> % TABLE 33 GREEK EXTENDED
```

```

> <U1F08>..<U1F0F>;<U1F18>..<U1F1D>;<U1F28>..<U1F2F>;<U1F38>..<U1F3F>;/
> <U1F48>..<U1F4D>;<U1F59>..<U1F5F>;<U1F68>..<U1F6F>;/
> % TABLE 34 GREEK EXTENDED
> <U1F88>..<U1F8F>;<U1F98>..<U1F9F>;<U1FA8>..<U1FAF>;<U1FB8>..<U1FBC>;/
> <U1FC8>..<U1FCC>;<U1FD8>..<U1FDB>;<U1FE8>..<U1FEC>;<U1FF8>..<U1FFC>;
> % TABLE 122 HALFWIDTH AND FULLWIDTH FORMS
> <UFF21>..<UFF3A>
>
>
> -----
> Annex 2 -- Replacement text for "alpha" category "
>
> alpha /
> % TABLE 2 BASIC LATIN
> <U00AA>;<U00BA>;<U00D7>;/
> % TABLE 6 IPA EXTENSIONS
> <U0294>..<U0298>;<U02A1>;<U02A2>;/
> % TABLE 10 GREEK SYMBOLS AND COPTICS
> <U03D0>..<U03D6>;<U03DA>;<U03DC>;<U03DE>;<U03E0>;<U03E2>..<U03F3>;/
> % TABLE 10 GREEK SYMBOLS AND COPTICS
> % TABLE 34 GREEK EXTENDED
> <U1FC2>..<U1FC4>;/
> % TABLE 14 HEBREW
> <U05B0>..<U05B9>;<U05BB>..<U05BD>;<U05BF>;<U05C1>..<U05C2>;/
> <U05D0>..<U05EA>;<U05F0>..<U05F2>;/
> % TABLE 15 ARBIC
> <U0621>..<U063A>;<U0640>..<U0652>;<U0670>..<U06B7>;<U06BA>..<U06BE>;/
> <U06C0>..<U06CE>;<U06D0>..<U06DC>;<U06E5>..<U06E8>;<U06EA>..<U06ED>;/
> % TABLE 17 DEVANAGARI
> <U0901>..<U0903>;<U0905>..<U0939>;<U093E>..<U094D>;<U0950>..<U0952>;/
> <U0958>..<U0963>;/
> % TABLE 18 BENGALI
> <U0981>..<U0983>;<U0985>..<U098C>;<U098F>..<U0990>;/
> <U0993>..<U09A8>;<U09AA>..<U09B0>;<U09B2>;<U09B6>..<U09B9>;/
> <U09BE>..<U09C4>;<U09C7>..<U09C8>;<U09CB>..<U09CD>;<U09DC>..<U09DD>;/
> <U09DF>..<U09E3>;<U09F0>..<U09F1>;/
> % TABLE 19
> <U0A02>;<U0A05>..<U0A0A>;<U0A0F>..<U0A10>;<U0A13>..<U0A28>;/
> <U0A2A>..<U0A30>;<U0A32>..<U0A33>;<U0A35>..<U0A36>;<U0A38>..<U0A39>;/
> <U0A3E>..<U0A42>;<U0A47>..<U0A48>;<U0A4B>..<U0A4D>;<U0A59>..<U0A5C>;/
> <U0A5E>;<U0A74>;/
> % TABLE 20
> <U0A81>..<U0A83>;<U0A85>..<U0A8B>;<U0A8D>;<U0A8F>..<U0A91>;/
> <U0A93>..<U0AA8>;<U0AAA>..<U0AB0>;<U0AB2>..<U0AB3>;<U0AB5>..<U0AB9>;/
> <U0ABD>..<U0AC5>;<U0AC7>..<U0AC9>;<U0ACB>..<U0ACD>;<U0AD0>;<U0AE0>;/
> % TABLE 21
> <U0B01>..<U0B03>;<U0B05>..<U0B0C>;<U0B0F>..<U0B10>;<U0B13>..<U0B28>;/
> <U0B2A>..<U0B30>;<U0B32>..<U0B33>;<U0B36>..<U0B39>;<U0B3E>..<U0B43>;/
> <U0B47>..<U0B48>;<U0B4B>..<U0B4D>;<U0B5C>..<U0B5D>;<U0B5F>..<U0B61>;/
> % TABLE 22
> <U0B82>..<U0B83>;<U0B85>..<U0B8A>;<U0B8E>..<U0B90>;<U0B92>..<U0B95>;/
> <U0B99>..<U0B9A>;<U0B9C>;<U0B9E>..<U0B9F>;<U0BA3>..<U0BA4>;/
> <U0BA8>..<U0BAA>;<U0BAE>..<U0BB5>;<U0BB7>..<U0BB9>;<U0BBE>..<U0BC2>;/
> <U0BC6>..<U0BC8>;<U0BCA>..<U0BCD>;/
> % TABLE 23
> <U0C01>..<U0C03>;<U0C05>..<U0C0C>;<U0C0E>..<U0C10>;<U0C12>..<U0C28>;/
> <U0C2A>..<U0C33>;<U0C35>..<U0C39>;<U0C3E>..<U0C44>;<U0C46>..<U0C48>;/

```

```

> <U0C4A>..<U0C4D>;<U0C60>..<U0C61>;/
> % TABLE 24
> <U0C82>..<U0C83>;<U0C85>..<U0C8C>;<U0C8E>..<U0C90>;<U0C92>..<U0CA8>;/
> <U0CAA>..<U0CB3>;<U0CB5>..<U0CB9>;<U0CBE>..<U0CC4>;<U0CC6>..<U0CC8>;/
> <U0CCA>..<U0CCD>;<U0CDE>;<U0CE0>..<U0CE1>;/
> % TABLE 25
> <U0D02>..<U0D03>;<U0D05>..<U0D0C>;<U0D0E>..<U0D10>;<U0D12>..<U0D28>;/
> <U0D2A>..<U0D39>;<U0D3E>..<U0D43>;<U0D46>..<U0D48>;<U0D4A>..<U0D4D>;/
> <U0D60>..<U0D61>;/
> % TABLE 26
> <U0E01>..<U0E3A>;<U0E40>..<U0E5B>;/
> % TABLE 27
> <U0E81>..<U0E82>;<U0E84>;<U0E87>..<U0E88>;<U0E8A>;<U0E8D>;/
> <U0E94>..<U0E97>;<U0E99>..<U0E9F>;<U0EA1>..<U0EA3>;<U0EA5>;<U0EA7>;/
> <U0EAA>..<U0EAB>;<U0EAD>..<U0EAE>;<U0EB0>..<U0EB9>;<U0EBB>..<U0EBD>;/
> <U0EC0>..<U0EC4>;<U0EC6>;<U0EC8>..<U0ECD>;<U0EDC>..<U0EDD>;/
> % TABLE ??
> <U0F00>;<U0F18>..<U0F19>;<U0F35>;<U0F37>;<U0F39>;<U0F3E>..<U0F47>;/
> <U0F49>..<U0F69>;/
> <U0F71>..<U0F84>;<U0F86>..<U0F8B>;<U0F90>..<U0F95>;<U0F97>;/
> <U0F99>..<U0FAD>;<U0FB1>..<U0FB7>;<U0FB9>;/
> % TABLE 28
> <U10A0>..<U10C5>;<U10D0>..<U10F6>;/
> % TABLE 50 .. HIRAGANA see J-30
> <U3041>..<U3094>;<U309B>..<U309C>;/
> <U30A1>..<U30F6>;<U30FB>..<U30FC>;/
> % TABLE 51
> <U3105>..<U312C>;/
> % CJK see J-31
> <U4E01>..<U9FA5>;/
> %
> <UAC00>..<UD7A3>;/
> % Misc.
> <U00B5>;<U00B7>;<U02B0>..<U02B8>;<U02BB>;<U02BD>..<U02C1>;/
> <U02D0>..<U02D1>;<U02E0>..<U02E4>;<U037A>;<U0559>;<U093D>;<U0B3D>;/
> <U1FBE>;<U203F>..<U2040>;<U2102>;<U2107>;<U210A>..<U2113>;<U2115>;/
> <U2118>..<U211D>;<U2124>;<U2126>;<U2128>;<U212A>..<U2131>;/
> <U2133>..<U2138>;<U2160>..<U2182>;<U3005>..<U3006>;<U3021>..<U3029>
>
> _____ end of Japan comments; beginning of Netherlands comments _____
>
>

```

> Comments with the NNI no vote on FCD 14652

> GENERAL

> The text has certainly been improved. Nevertheless the whole is far
> too much oriented on POSIX conventions. This implies in practice that it
> will be difficult to get the necessary information about cultural
> conventions from knowledgable people who do not understand at all the
> frames in which this information should be placed. We are afraid that
> the result may suggest a false security to software writers, that the
> data taken will reflect the true conventions, while it does not.

Response:

Ballot comments on the first CD requested that a formal description language be employed, and formal specifications normally make specifications harder to make. But it also makes the specifications readily usable with an application. For free-form cultural specifications we recommend using ISO/IEC 15897, which allows narrative specification of most cultural conventions.

> Technical comments

>

> We support the US comments on the two letter mnemonics. These things have been criticised repeatedly, because they are not mnemonic at all.
> If short identifiers are wanted the Uxxxx forms will, and there is no need to multiply the ways characters may be identified. Unless they are removed our NO vote cannot be turned into YES.

Rejected. The use reflects standard and industry practice. It is permitted to use the UXXXX identifiers also.

> The tables for toupper and tolower contain bugs according to the US NB.
> The D of C does not answer convincingly why the US should be wrong.
> Until further argument is supplied we cannot approve this disposition.
> The alpha specification is said to be different from that in Java.
> Anyway at a first inspection it is unacceptable to classify the MICRO SIGN and the FEMININE and MASCULINE ORDINAL INDICATORS as alpha. They were classified in ISO 6937/1:1983 as specials, and that is what they are. (No SC2 standard specifies a classification of characters anymore.)

The US comments on the CD has already been responded to. Feminine and masculine ordinator are part of words. The characters mentioned are all in the TR 10176 annex A. A note will be added to say that this is also meant for what is allowed in words and recommended for identifiers as per TR 10176 annex A.

> In LC_TYPE the list contains under class the term "non_spacing". This is to be changed into "combining", which is the term used in ISO/IEC 10646-1. No SC2 standard at present specifies non-spacing characters.
> The non-spacing diacritical marks in ISO/IEC 6937:1994 are not characters and are not included in the character repertoire of 6937.
> The disposition on p. 38 of N 2637 is a misrepresentation of the wording of 6937 and is totally wrong. ISO/IEC 6937 does not specify any combination of characters. It just specifies a coding for each of the characters of its repertoire, some with one octet, some with two. That is all. It is a mixed coding system, like UTF8 with 10646.

Accepted, non_spacing will be changed to "diacritical mark"

> We found checking of tables for LC_TYPE very time-consuming, and without access to ISO/IEC 10646-1:1993 and all its amendments almost impossible. Nevertheless, we took a few samples, and had to conclude that these tables as given are just unreliable.

>

> In the toupper list there are duplicates of:

> (<U01C6>,<U01C4>)

> (<U01C9>,<U01C7>)

> (<U01CC>,<U01CA>)

> (<U01F3>,<U01F1>)

Accepted.

> As for the tolower list, the assignment of upper equivalents to IPA
> (International Phonetical Alphabet) letters is highly artificial. IPA
> characters are essentially classless, and inventing capitals for them is
> merely a display of academic pedantry. We support the comments of Japan
> on this topic (J-23), and do not accept the disposition.

Rejected. This is harmonized with Unicode specs, as represented by
the US member body.

> Furthermore, we found ambiguities:
> (<U01C6>,<U01C4>)
> (<U01C6>,<U01C5>)
> etc.

This is related to the following:

> Expressed in visible letters we find (the Z WITH CARON is here written
> as a H):

>
> short id letter class tolower toupper
> U01C4 DH up dh
> U01C5 Dh up low DH
> U01C6 dh low DH,Dh DH
> U01C4 LJ up lj
> U01C5 Lj up low LJ
> U01C6 lj low LJ,Lj LJ
> U01C4 NJ up nj
> U01C5 Nj up low NJ
> U01C6 nj low NJ,Nj NJ
> U01C4 DZ up dz
> U01C5 Dz up low DZ
> U01C6 dz low DZ,Dz DZ

>
> This is quite ridiculous. We wonder what we would have found, had we
> inspected more.

This is a special case where there are both upper and lower case letters
in a character, and it only affects these characters. The tables
will be checked and corrected.

> _____ end of Netherlands comments; beginning of USA comments _____
>

> USA comments

>
> The US National Body votes to Disapprove FCD 14652 - Information
> technology - Programming languages, their environments and system
> software interfaces - Specifications for Cultural Conventions. See
> comments below.

> U.S. comments accompanying the NO vote on FCD 14652.

>
>

> General Comments

>

> 1. The U.S. considers it inappropriate to extend the ISO 9945 POSIX
> framework to provide ISO/IEC 10646 support without sufficient
> attention to the implication of the shift of focus from locale
> definition for multiple character sets to FDCC-set definition
> based on the *universal* character set. The draft, throughout,
> shows evidence of piecemeal extensions to the existing framework,
> instead of holistic consideration of the UCS. As a result, it
> is riddled with inconsistencies of coverage that will lead to
> problems of implementation.

>

> The U.S. urges that either:

>

> 1. The support for 10646 in 14652 be systematically circumscribed
> to a well-defined subset, with no pretensions to universality,
> so that what is presented can at least be checked for
> internal consistency, or

>

> 2. The support for 10646 in 14652 be corrected to properly
> treat the UCS as a *universal* character set, with attendant
> changes to such constructs as LC_CTYPE to ensure that
> universal properties associated with the UCS itself are
> not mixed with cultural conventions associated with the
> FDCC-set definitions.

Rejected. It is important to define properties for the whole of UCS.

In that way most character sets can be covered.

14652 needs to be able to cater for all character sets, and that
FDCC-sets can be written with a repertoire different from that of
UCS in its evolving versions.

> 2. In line with the implications of comment #1, the U.S. considers
> it inappropriate to define *any* character properties for the
> UCS in a standard devoted to the specification of cultural
> conventions. The one exception to this is the case mapping of
> characters, which have a few, well-known language-specific
> exceptions from the general, default mappings.

>

> The U.S. is well aware that since 14652 is explicitly an extension
> of the ISO/IEC 9945 framework, with a goal of backwards compatibility to
> ISO/IEC 9945, the outright omission of existing locale-related
> constructs would not be a viable option. However, just as clause
> 4.1.1 on Character representation formally deprecates the 9945
> practice of representing characters in terms of numeric constants,
> in favor of symbolic names throughout, so 14652 could and should
> deprecate the use of LC_CTYPE as part of FDCC-set definitions.
> At the very least it should not compound the error by extending
> the number of character classes defined and enumerated in the
> wrong standard for this purpose.

>

> The U.S. categorically rejects the disposition of its comments
> on the CD 14652 regarding this topic by the editor of 14652.
> The editor claimed in the Disposition of comments, that
> "In general the properties of a character is thus culturally
> dependent." The U.S. states that this is technically incorrect
> and, if taken seriously promotes bad software engineering and

> interoperability problems in international contexts. The U.S.
> restates its earlier comment:
>
> "Character properties are *not* subject to local cultural
> conventions. It is *not* acceptable to redefine GREEK SMALL
> LETTER TAU to be uppercase, or to define CIRCLED DIGIT SIX
> to be punctuation, for example. Such definitions do not belong
> in specifications for *cultural conventions*, or if
> character properties must be defined there, they should at
> least be clearly earmarked as different from all other
> categories of an FDCC-set."

Rejected.

Character properties may be culturally dependent, such as which characters to be considered letters, as some cultures could consider the letters of other cultures as special characters, for example Hebrew characters in a culture using the Latin script. Also for example digits and hexadecimal digits may be culturally dependent, and some special character may be used for specific purposes such as quotation or spacing. It is agreed that a specific set of properties is advisable and thus the i18n FDCC-set has been specified.

> 3. The U.S. considers the extension of locales to FDCC-sets not
> to be the best mechanism for the international specification of
> cultural conventions. The specification of FDCC-sets in 14652
> extends an already faulty mechanism that has largely been
> abandoned outside the UNIX community as a means of specifying
> cultural conventions. By promoting the definition of FDCC-sets
> with even more information crammed together in single constructs,
> regardless of the appropriate scope of the different kinds of
> cultural data involved, 14652 has the potential to further
> fragment and Balkanize the implementation of cultural adaptability,
> instead of promoting commonalities and comprehensible
> interoperability. 14652 should provide a mechanism for describing
> cultural conventions, without enforcing the concept of such
> descriptions constituting a monolithic FDCC-set definition.

The cultural conventions used in this standard are proven to be useful, and the specification technique is adequate.

The POSIX notation has fostered that the information on cultural issues have been published, and that this information can be used on a number of platforms, thus leading to a more uniform and consistent handling of these issues.

> 4. Furthermore, the draft for 14652, despite the formal claim
> to be "independent of platforms" (page 4) shows a distinct
> UNIX bias, as well as its orientation to particular UNIX
> implementations that presuppose association of a locale (read
> FDCC-set) with a process. This orientation runs very deep in
> the draft, down to the definitions of terms themselves. For
> example, a recurrent phrase in the definition of terms in Clause
> 3 is "in the current FDCC-set". This expression is a direct
> calque, derived from the phrase "in the current locale" in
> corresponding definitions in the Glossary for the X/Open
> XSH and XCU specifications. Such definitions are all embedded
> in a context that presupposes UNIX-oriented API's such as
> setlocale(). It is one thing to extend the concept of locale
> within an explicitly acknowledged UNIX framework where it

> makes sense; it is another thing, entirely, to push it into an
> *international* and supposedly platform-independent standard,
> where some of the basic definitions themselves are lacking
> an agreed-upon context. At the very minimum the concept of
> "the current FDCC-set" must be either defined in a meaningful
> platform-independent way in 14652, or it must be dropped from
> 14652.

There are some specifications where a global FDCC-set model have been assumed. This is in line with TR 11017. The global locale model comes from programming language C (not POSIX, which does not have locale oriented APIs besides `setlocale()`). The global locale model should be changed into a more object oriented approach. It will be investigated to change "current FDCC-set" into "associated FDCC-set" indicating that this is not a global FDCC-set.

> Another example of this kind of thing can be seen on page 49,
> for the definition of `LC_MESSAGES`. The definition of "yesexpr"
> and "noexpr", which make use of the concept of an "extended
> regular expression." "Extended regular expression" is not
> defined in FCD 14652; the text of FCD 14652 just refers out
> to ISO/IEC 9945-2, clause 2.8.4. In the original form, the
> the X/Open specifications, "extended regular expressions" are,
> of course, defined right there in the specification, where
> they are referred to. (And they are quite complex, in and
> of themselves.) But FCD 14652 is just assuming this UNIX-oriented
> background, derivative from the X/Open specifications, instead
> of standing as a self-contained, platform-independent standard.

Accepted in principle.

The extended regular expression problem will be clarified.

> =====
>
> Technical Comments
>
> 1. The main thrust of 14652 is the formal definition of the
> syntax for an FDCC-set. However, the standard lacks a formal
> syntactic definition as generally understood. This makes it
> more difficult A) to determine whether the formal definition
> is complete and consistent, and B) for an implementer to
> determine if his implementation is complete and conformant.
>
> Therefore, 14652 should include a formal BNF definition of the syntax for
> the FDCC-set.
>
> Note that the X/Open specifications for locale syntax from
> which FCD 14652 is descendant *do* provide a formal BNF syntax
> for locale definition. Furthermore, as it correctly should
> do, the text there states, "The grammar takes precedence over
> the text." Since the BNF grammar is logically and formally complete,
> any mistake or incompleteness in the text of the specification,
> which may have been missed during review, is dealt with by
> openly declaring that the formal grammar is the correct
> specification where there is any question.
>

> It is a serious defect of 14652, betokening a lack of rigor
> and thoroughness, that no similar effort has been made to
> provide the corresponding formal definition for the FDCC-set.

Accepted in principle.

The syntax definition is a formal definition of the syntax of FDCC-sets.
According to SC22 recommendations a more formal approach would not
take precedence over the less formal approach.
However a BNF description will be added.

> 1a. Re Section 3.1 "terms and definitions",
>
> The "portable character set" should be defined, with a reference to the
> full list in Table 3.

accepted.

> 1b. Re Section 3.1.15 "collating sequence",
>
> The mention of the LC_LOCALE category should be "LC_COLLATE" category.

accepted.

> 2. Re 3.2.3 Ellipses
>
> The FCD 14652 improves the description of the ellipses
> conventions, but still leaves the basic U.S. objection
> to the introduction of 3 styles of ellipses unaddressed.
>
> The U.S. restates its basic objection:
>
> "The introduction of distinctions between two-dot, three-dot,
> and four-dot ellipses is overly complex and subject to error
> in use."
>
> That this use of different numbers of dots is likely to
> provoke errors is embarrassingly demonstrated by the text of
> FCD 14652 in the very clause in question, where the decimal
> symbolic ellipsis is exemplified as "<j0148>..<j0153>", when
> the decimal symbolic ellipsis has been defined as "....",
> so that the example should read "<j0148>....<j0153>".
>
> The U.S. restates its preference:
>
> "It is generally better practice to simply have a single
> range notation for a formal syntax, while maintaining clear
> syntactic differentiation of the elements which can form the
> items at each end of a range. So if the FDCC-set syntax must
> distinguish a range of symbols, a range of decimal values,
> a range of octal values, a range of hexadecimal values, and
> so on, the notation for "symbol", "decimal value", "octal
> value", "hexadecimal value", and so on should be unique and
> mutually exclusive, so that interpretation of the type of
> range does not depend on the number of dots."

Rejected. The ellipses should not be dependent on the
names of the characters involved. The example will be clarified.

section 5.1 will explain that the absolute ellipses are deprecated. A rationale for the distinction between decimal and hexadecimal ellipses will be given.

> 3. Re 4 FDCC-set

>
> There are nowhere naming guidelines for FDCC-set files. The U.S. understands that this standard wishes to keep away from the idiosyncrasies of file-naming conventions in different operating systems. However, recommendations should be given, or alternatively it should be specified that there are no rules, to make things clear for those of us who remember naming conventions for locales.

Accepted in principle.

Naming rules is the subject of ISO/IEC 15897.

A note with a reference to this standard will be given.

> The FDCC-set is declared to be "the definition of the subset of a user's information technology environment that depends on language and cultural conventions." This reflects one of the fundamental problems with the FDCC-set concept--it presumes that there is a well-defined set of such information appropriate to a particular user's "environment". This completely skates by the problem of multilingual and multicultural environments that are increasingly common in today's IT settings. By defining everything together as a FDCC-set, the standard precludes more promising approaches that distribute cultural conventions to the objects where they are appropriate.

>
> At the very least, 14652 should acknowledge this limitation to the FDCC-set.

there is conceptually not much difference between haveing a number of scattered cultural conventions and then collecting these into one hat. The difference is that if they are collected, they can also be managed as a whole, so the user can specify a collection, say "Italian". If not collected, the user needs to set an unknown number of different variables to his/her liking. The possibility of individually setting preferences is also available in the "one hat" model. It can be explained that applications may take advantage of cultural information in the FDCC-set to provide even further cultural adaptability.

> The statement "This standard also defines an FDCC-set named 'i18n' with values for each of the above categories" (page 5) is not technically correct, since the "i18n" LC_COLLATE category is not defined in *this* standard but in ISO/IEC 14651. Definition by reference to other standards is o.k.--in fact it is preferable where appropriate. But the statement on page 5 should be qualified to point this out.

The standard defines the values of the LC_COLLATE category in the i18n FDCC-set to be that of 14651. The reference to 14651 will be clarified.

> 4. Re 4.1.1 Character representation (1)

>
> The description of character representation by symbolic name
> includes in the example the symbol "<c-cedilla>", which does
> not in fact occur in the i18n repertoiremap defined in the
> standard. While "<c-cedilla>" is a valid symbol, it is not
> self-consistent for the standard to promote an elaborate
> repertoiremap of symbols and then use different, undefined
> symbols in the examples in the text. Either such examples should
> be corrected to strictly use symbols from the repertoiremap,
> or a statement should be added to 3.2 Notations, allowing that
> symbols not in the repertoiremap will be used in examples, to
> illustrate the range of symbols allowed by the syntax.

The examples are just examples, and should be allowed to use whatever the standard prescribes as valid. A note will be added on this in the notations section.

> The U.S. sees no reasonable need for allowing the right angle
> bracket as part of symbolic names, thus requiring escaping. That
> is occasioned only by the choice to include ">" as a
> shorthand for circumflex in the repertoiremap list of symbols.
> It would be better to omit this requirement altogether.

The inclusion of the right angle bracket is done for generality, and also for backwards compatibility with the POSIX standard. The right angle bracket is used in existing implementations and naming schemes. It is also used in 14651.

> 5. Re 4.1.2.4 charmap
>
> FCD 14652 states "For the actual use of a FDCC-set, at most
> one charmap may be in use,..." This is fundamentally at odds
> with applications and application architectures that handle
> multiple character encodings simultaneously. It fundamentally
> limits the usefulness of the charmap concept. The text of 14652
> should clarify how an application is to specify the ability
> to support multiple character encodings, while making use of
> one or more sets of cultural conventions.

You can have more than one FDCC-set in use at a given time. The statement only indicate that the run-time FDCC-set is character encoding dependent. The character encoding may span multiple character sets.

> 6. Re 4.2.1 Basic keywords: alpha
>
> FCD 14652 defines the "alpha" category as "letters or other
> characters used in words of natural languages such as syllabic
> or ideographic characters". But the actual definition of
> the alpha class under the i18n LC_CTYPE on pp. 16-17, while
> much improved from the CD 14652 listing, still has defects
> in it. It includes some punctuation, such as U+203F and U+2040
> that cannot reasonably be considered alphabetic, while also
> omitting whole classes of characters, such as combining marks,
> that can be "used in words of natural languages". This problem
> stems partly from the inconsistency between the attempt to
> make the "alpha" category mean "alphabetic (broadly construed
> to include syllabic and ideographic characters)" versus the

> use of "alpha" through a POSIX-style API isalpha() to assist
> in the lexing of identifiers. This inherent inconsistency,
> which can be glossed over for small character sets or Japanese,
> is glaringly obvious when applied to all of 10646. If 14652
> is going to (erroneously, in our opinion) insist on extending
> the alpha class in this standard (or get its values from
> TR 10176 annex A, which are also wrong), then it should take
> an explicit stand on whether "alpha" is to mean "alphabetic"
> or is to be used to define identifier boundaries. The implications
> are different for which characters are included or excluded.

The FCD regards identifiers to be what can be recognized as words
of a language. The set of characters allowed for identifiers and
for words of natural languages are thus the same.

>
> The text of 14652 should show some sign of having taken into
> account the detailed specification of identifier syntax and
> of the alphabetic property provided by the Unicode Consortium.

The standard reflects the recommendations of TR 10176, in the
development of which consideration of Unicode specifications
were given.

> 7. Re 4.2.1 Basic keywords: space
>
> In the disposition of earlier U.S. comments, the editor stated
> that "The NO-BREAK exclusion will be explained, classes <blank>
> and <space> are meant for finding possible break points." While the
> revised text does state that the "space" class is "to find
> syntactical boundaries" it does not explicitly explain the
> NO-BREAK exclusion. The enumeration of the "space" class on page
> 17 does correctly omit the NO-BREAK spaces, U+00A0, U+2007,
> and the ZERO-WIDTH NO-BREAK SPACE U+FEFF, but the definition
> on page 9 is not explicit about this omission.

The expalantion about NO-BREAK SPACE will be added.

> 7a. Re 4.2.1 Basic keywords: graph and print
>
> The definitions for "graph" and "print" should be moved after the
> definition of "xdigit" since they refer to it.

accepted.

> 7b. Re 4.2.1 Basic keywords: blank
>
> The definition for "blank" should be moved before the definition for
> "space", which refers to it.

accepted.

> 8. Re 4.2.1 Basic keywords: class
>
> On pp. 10-11, the text for FCD 14652 lists among others, six
> classes relevant to bidirectional layout:
>
> left_to_right

> right_to_left
 > num_terminator
 > num_separator
 > segment_separator
 > block_separator
 >
 > These 6 classes should **not** be defined here. They are merely a
 > subset of the complete set of bidirectional properties, which
 > are **normatively** defined in the Unicode Standard. The listing
 > and defining of any of these (especially incorrectly, and
 > incompletely) in FCD 14652 can only lead to interoperability
 > problems with applications that implement the Unicode bidi algorithm.
 > These classes and their incomplete definitions on page 23 **must**
 > be removed from FCD 14652. If they are not, the following keywords
 > definition must be phrased as following:
 >
 > "num_terminator:
 > characters which may be adjuncted before or after
 > the digits of a number", which is in keeping with the intended meaning
 > of this class in the Unicode bidirectional algorithm.
 >
 > "number separator:
 > characters which can appear between digits of numbers written with
 > any of the characters in the digit class". This formulation makes
 > it clearer that the number separators do not segregate between
 > numbers, but appear between parts of the same number.

These keywords will be removed, due to immaturity of the specification. Functionality like this may be added in a future amendment.

> 8a. Re 4.2.1 "Basic keywords", definition of "map",
 > explanation for "tosymmetric" says: "For each pair also the mapping
 > from the second operand to the first operand is also defined".
 > It is not clear what the first "also" refers to. And it is not
 > clear "also defined" by who? While the U.S. prefers that the
 > entire "tosymmetric" class be removed, because of the errors in
 > the listing, a clearer reformulation of this explanation would be:
 > "For each pair, the mapping from the second operand to the first
 > operand is also implied".

Change of wording accepted in principle. "Tosymmetric" will be removed.

> 8b. Re 4.2.2.1 "Transliteration statements", the paragraph
 > starting with "The order the <transliteration-strings> is defined
 > in" is confusing. "...having characters that are all in
 > the coded character that is transformed into" is not "for example"
 > but should be made an essential constraint. It is not clear either
 > on what the "desired string length" is based.
 > A better phrasing is needed here, if this section is to be
 > retained at all in the standard.

accepted. The following wording will be used:

"If a transliteration statement contains more than one
 <transliteration-string>, the order that each <transliteration-string>
 occurs in the transliteration statement defines the precedence order

for choosing a particular <transliteration-string> to substitute for the <transliteration-source>. When a process makes use of a transliteration statement to transliterate text, and that transliteration statement contains more than one <transliteration-string>, that process shall choose the first <transliteration-string>, in the defined precedence order, that satisfies the requirements of the transliteration. Note: the exact definition of the concept of satisfying the requirements of the transliteration is outside the context of this standard. If, for example, a transliteration involves a change in the coded character set of a string, a <transliteration-string> must be chosen, all of whose elements are members of that coded character set. In order to determine this, it would be expected that a repertoire describing which characters are to be present in the resulting transformed string be available to the transliteration API. Also, a transliteration may involve requirements such as that string length not change under transliteration. Such requirements may also affect the choice among alternative <transliteration-string> values."

The second parameter in the transliteration statement definition will be removed.

> 8c. re 4.2.2.1 "transliteration statements", paragraph starting
> with "If more than one transliteration statement". The condition of
> having more than one transliteration statement for a given
> <transliteration-source> should simply be an error. Allowing for
> assumption that the "last transliteration statement" is applied
> creates technical complications in implementation.
> a) This is not in style with the precedence of transliteration
> strings in the same statement, where the first satisfying one is
> chosen.
> b) This complicates the building of the internal tables, because the
> program (equivalent of localedef) cannot be sure that a
> specification is definitive until the end of all specifications.
> The U.S. prefers that the entire section 4.2.2. be omitted until
> the mechanism is worked out better, but if retained, then section
> 4.2.2.1 should simply state that duplicate transliteration statements
> are ignored (with a warning).

Accepted, in the retain version. The allowance for more than one statement for one transliteration-source was to ease tailoring. This can be done instead with a "redefine" statement.

> 9. Re 4.2.3 "i18n" LC_CTYPE category
>
> Concerning the classes "right_to_left",
> "num_terminator", "num_separator" etc... which are related to Bidi:
> These classes are similar to classes defined in Unicode, but not
> identical. Even for those classes which are common in both standards, the
> content of the classes is much different.
>
> Our assumption is that the authors wish to keep in sync with the
> classification in the Unicode standard. This is far from true in
> this version of 14652.

The specifications will be removed.

> This classification thing is a big issue. The Unicode experts have
> spent much time on it, and have not got a perfect result (yet?).
> This standard does too much or too little about it, with such
> blatant errors as classifying Eastern Arabic-Indic digits (U06F0 to
> U06F9) as right-to-left instead of digits. If this standard cannot
> just refer to the Unicode classification, it should "lift" the
> classification lists from Unicode. Trying to do it again by itself
> is a waste of time and is likely to give results much worse than
> what is in Unicode, because not enough efforts will be invested.

There are other sources than Unicode to determine these specifications.
WG20 needs to carefully inspect all submissions. There has not been
a submission from Unicode on this to WG20, on acceptable terms to WG20.

> The following text identifies a number of errors in the class definitions
> given in the text of FCD 14651, including, but not necessarily
> limited to:

>
> 9a.
>
> punct (page 17) defines the range <U00A0>..<<U00BF>, which is
> inconsistent with the (correct) specification of <U00AA> and
> <U00BA> as alpha on page 16.

Accepted.

> 9b.
>
> digit (page 17) includes the ideographic zero (U+3007) and the Han
> characters for 1 to 9. This is incorrect, since the Han characters
> do not normally form decimal radix numbers, and should not be
> characterized as digits. (The ideographic zero is a debatable
> exception.) It is also inconsistent, since it omits Hangzhou
> and alternative, fraud-proof Han characters for the same values.
> The correct solution is to omit ideographs altogether from the
> "digit" class.

accepted. Hangzhou numerals will be added.

> 9c.
>
> The toupper and tolower case mapping tables on pp. 19..21 contain
> several errors that were identified in the U.S. comments to the
> CD draft, errors that were summarily dismissed by the editor in
> the disposition of comments. The U.S. categorically rejects that
> disposition and reiterates its statement of the errors:
>
> "In the toupper table, the entry (<U0258,<U018E>) is incorrect and
> should be removed."
>
> "In the toupper table, (<U0275>,<U019F>) should be added."
>
> The editor stated in response: "This is not obvious, and needs
> further documentation."
>

> The correct case mappings are: U+01DD <--> U+018E
> U+0275 <--> U+019F
>
> as documented in the Unicode Character Database:
>
> 018E;LATIN CAPITAL LETTER REVERSED E;Lu;0;L;;;;N;LATIN CAPITAL LETTER
> TURNED E;;;01DD;
> 01DD;LATIN SMALL LETTER TURNED E;Ll;0;L;;;;N;;;018E;;018E
>
> 019F;LATIN CAPITAL LETTER O WITH MIDDLE TILDE;Lu;0;L;;;;N;LATIN CAPITAL
> LETTER BARRED O;;;0275;
> 0275;LATIN SMALL LETTER BARRED O;Ll;0;L;;;;N;;;019F;
>
> The incorrect case mapping and the omitted case mapping shown
> in FCD 14651 have as their origins the incomplete and inconsistent
> set of name changes required by WG2 during the merger of the
> Unicode 1.0 repertoire and the DIS 2 10646 repertoire in 1991. These
> name changes are also shown in the Unicode Character Database,
> where you can see the original Unicode 1.0 name for these characters,
> which reflected the normal naming conventions for case pairs. The
> fact that WG2 requirements disturbed the symmetry between the
> names of the case pairs does not invalidate the case mappings
> themselves.
>
> Is that enough?

accepted, due to further information given.

> "In the toupper table, (<U1E9B>,<U1E60>) should be added."
>
> The editor stated in response: "The characters will be considered
> when they both are fully included in 10646."
>
> The toupper table already includes the entry (<U017F>,<U0053>),
> so there can be no question that the intent is to specify the
> uppercase of the long s to be a (normal) capital S. So there is
> also no question that the uppercase of the long s with underdot
> should be a (normal) capital S with underdot. The character
> U+1E9B LATIN SMALL LETTER LONG S WITH DOT ABOVE was added to
> ISO/IEC 10646 by Amendment 7. The normative references for
> FCD 14652, on page 1, include:
>
> ISO/IEC 10646:1997, "Information technology - Universal Multiple-
> Octet Coded Character Set (UCS), including Cor. 1 and AMD 1-9"
>
> Therefore there is no question of the propriety of including
> U+1E9B, and that disposition of comments has no valid grounds
> to stand.

Accepted.

> 9d.
>
> The "tosymmetric" table on page 24 is derived from
> the informative Annex C, "Mirrored characters in Arabic
> bi-directional context", from 10646. There are two problems with
> this. First of all, it is dubious for one ISO character-related

> standard to define a *normative* list in its text derived from
> an *informative* list in the original standard. Changes to
> the 10646 informative list (which have happened, just recently),
> can cause a disconnect with the putatively normative list
> defined in the other standard.
>
> Second, and more disturbing, the "tosymmetric" mappings on
> page 24 contain gross errors, mapping for example,
> (<U2201>,<U2202>) and (<U22A8>,<U22A9>), which pairs are even
> casually evident not to be symmetric pairs. The U.S. can only
> conclude that not even the slightest care was taken in producing
> this listing, and the entire class should be omitted from
> FCD 14652.

Accepted. "tosymmetric" will be removed.

> 10. Re 4.2.2 Character string transliteration
>
> The U.S. considers this proposed mechanism for specifying
> transliteration to be of dubious value. It is not clear
> that it is either a complete nor particularly elucidative
> way of specifying transliterations. Nor is it apparent that
> the already cluttered mechanism of FDCC-set specifications
> should be further weighed down and fragmented by also
> specifying transliteration schemes in them.
>
> The entire mechanism of specification of transliteration
> should be removed from FCD 14652.

Rejected.

The transliteration description has been worked on for some time and is stable. It will be specified that the longest string that matches will be the one that is transliterated. A note that this specification only caters for simple transliteration and that more advanced transliteration is either cumbersome or not addressed will be added.

> 11. Re 4.3 LC_COLLATE
>
> The U.S. restates its basic objection to the syntax proposed
> here:
>
> "The syntax introduced for tailoring a collation sequence
> definition for cultural conventions is overly complex. It
> is very tightly coupled to the specific way in which
> a collation is defined in CD 14651, which itself is in
> question. A much simpler syntax has been promulgated by the
> Java developers to accomplish the same task, and it would
> be desirable to examine the alternatives before standardizing
> an LC_COLLATE syntax of unnecessary complexity. Unlike most
> of the rest of the categories involved in an FDCC-set
> definition, which merely specify lists of things, the
> LC_COLLATE syntax introduces notions of scope, reordering,
> and a macro control language. Granted that reordering
> rules are needed for defining collations, but it is
> unclear that all of the rest of the syntax is."
>

> The editor commented in the Disposition of Comments that
> "The mechanism used are one-line statements and then
> directives using prior art and tools like the C preprocessor."
>
> The thrust of the original U.S. comment was not to claim that
> 14652 was inventing things that no one had ever heard of --
> but that such mechanisms had not formerly been a part of the
> LC_COLLATE syntax for locale definitions. Introduction of
> such mechanisms distinctly complicates the processing of
> FDCC-set definitions. It is also specious to claim that these
> are "using prior art", since the "prior art" was not something
> applied prior to the constructs in question. One could, on
> that basis, recast the entire locale-related syntax in terms
> of a category grammar and require its processing through
> yacc and lex and claim it was "using prior art", for that
> matter. The U.S. still considers it of dubious value to
> introduce these complications into the parsing of FDCC-set
> definitions when the exact mechanisms for correctly specifying
> international string ordering are still under debate.

The LC_COLLATE specification builds on prior art as in POSIX-2 and the syntax for tailoring is needed - for tailoring. It is the same as specified in 14651.

> 12. Re 4.3 LC_COLLATE (cont.)
>
> FCD 14652 on page 24 states, in normative language, that
> "The collation sequence definition shall be used by regular
> expressions, pattern matching, and sorting." It is not clear
> yet that anyone has actually figured out exactly how to make
> use of a full 10646 collation sequence definition consistently
> in regular expression syntax. Until the problem of the
> extension of regular expression syntax to take 10646 into account
> can be resolved, it is not advisable for 14652 to make a
> normative requirement on collation that cannot obviously
> be followed.

accepted. "shall" changed to "may".

> 13. Re. 4.3.1 Collation statements
>
> The use of 3 different styles of ellipses in the syntax for
> collation statements is as objectionable as it is in the
> syntax for charmaps. It should be replaced by a specification
> for a single indication of range.

Rejected. The different semantics should be reflected in the syntax.

> 14. Re. 4.3.1 Collation statements
>
> On page 28, FCD 14652 advocates the use of the "absolute"
> ellipsis in an LC_COLLATE definition to stand for "the
> value of each character defined by the ellipsis". This can
> only be meaningful for a particular coded character set, since
> a symbolic representation of a character set does not have
> an inherent order. Cf. page 4: "The absolute ellipsis

> specification is only valid within a single encoded character
> set." Subclause 3.2.3 in fact deprecates this use of the
> absolute ellipsis. Therefore, the specification of collation
> statements in subclause 4.3.2 should also indicate that this
> is deprecated for collation statements and should state
> the limitation implied. FCD 14651 in fact makes no use of
> the "absolute" ellipsis in defining the common tailorable
> template.

accepted.

> 15. Re 4.3.3 "col_weight_max" keyword
>
> The minimum value of 7 is an unreasonable and unjustified
> value. Cf. the normative text on page 27, "If the two strings
> compare equal, the process shall be repeated for the next weight
> level, up to the limit "COLL_WEIGHTS_MAX". Yet FCD 14651 defines
> a tailorable template for a major subset of 10646 using just
> 4 levels, and no plausible account has been brought forward
> requiring more levels for culturally correct international string
> ordering. Arbitrarily requiring an artificially high minimum
> value is an implementation penalty that should not be imposed
> by a standard.
>
> By the way, the specification that the minimum value is 7
> seems at odds with the Disposition of Comments for the
> Canadian comment 10, which also objected to the minimum value
> of 7 for this value in the CD 14652. The Disposition of
> that comment stated:
>
> "accepted in principle. The default will be removed."
>
> But it appears that the default has not in fact been removed
> in the FCD 14652.

7 is not a default, it is a minimum requirement. It is compatible
with POSIX. To be able to process all POSIX conforming locales, it
is necessary to retain this value.
The rationale will be added in the rationale section. The requirement
will be moved to the conformance section, allowing smaller values
to be specified.

> 15a. Re 4.3.4 "script keyword"
>
> It is not clear how characters are allocated to specific scripts. This
> should be clarified.

accepted.

> 16. Re 4.3.5 "collating-element" keyword
>
> This piece of LC_COLLATE syntax appears to be intended to deal
> both with the issue of defining "multicharacter collating elements"
> of the normal sort (e.g. "ch" or "ll" in Spanish, "aa" in
> Danish, etc.) and apparently also as a mechanism for dealing with
> combining characters. The example "with ISO/IEC 6937" includes
>

> collating-element <e-acute> from <acute><e>
>
> This mechanism might make sense for a limited character set
> using combining characters exclusively, but does not specify
> how to deal with the *equivalence* of a preexisting, encoded
> form, and the collating-element so defined. This problem should
> be squarely addressed in the syntax provided.

The problem of combining characters is addressed in the charmap specification. A note can also be added for the LC_COLLATE, saying that there can be an equivalence up to the 3rd level, as is being specified in 14651.

> Furthermore, the example shows the dangers of trying to mix
> a syntax appropriate for the UCS with a syntax appropriate for
> arbitrary (non-universal) character sets. The "<acute>" cited
> above is the prepositive combining character from 6937 (which
> interestingly, in the il8n repertoire map is cited as "<'>",
> not "<acute>"). This can only make sense for a LC_COLLATE definition
> particular to that encoded character set, since it conflicts
> with the UCS' conventions for combining characters. Once again,
> 14652 is vacillating between encoding-specific representations
> and encoding-independent symbolic representations, when what it
> *should* be doing is making use of the *universal* character
> set representations.

The example is only an example, and it should illustrate the point

> 17. Re 4.3.8 "order_start" keyword
>
> The directives "forward" and "backward" are defined so that they
> "Specif[y] that the direction of scanning a substring in this
> script at a given point in a string is done towards the logical
> end/beginning of the string for this weight level." The problem
> with this definition is the interaction with the concept of
> being "in this script", the "script" keyword, and the "reorder_after"
> keyword. The "reorder_after" keyword can arbitrarily reorder a
> collating element from any one script "area" in a collation to
> any other. This raises an open issue of what the script identity
> of that character then is -- its inherent script as defined by
> the UCS, or the script defined by some scope for the "script
> keyword in the LC_COLLATE definition. This makes the scope of
> the qualification "in this script" unclear for the "forward"
> and "backward" directives.
>
> This is not just a theoretical concern. There is some real
> difference of opinion regarding the overlap and identity of
> some characters in the Latin and Cyrillic scripts, for example.
> Furthermore, correct collation of mixed-script, mixed-language
> data may require processing of accents in both directions,
> depending on the particular accents and the script of the
> base characters involved. It is not clear that the implications
> of interaction of these mechanisms is well-defined in the text
> of FCD 14652 as it currently stands. They should be clearly and
> completely stated.

The text will be aligned with that of 14651.

> 17a. Re 4.3.10.1 "Example of reorder-after",
>
> The symbols "<y8>" and "<z8>" are not defined in this standard,
> but appear only in the common tailorable template of FCD 14651.
> If they are going to be introduced in an example here in this
> standard, they need to be explained and clarified.

accepted.

> The usage of parentheses within the sequences in bullet 4 of
> the explanations is unclear. This usage should be clarified.

accepted.

> 18. Re 4.8 LC_PAPER, 4.9 LC_NAME, 4.10 LC_ADDRESS, 4.11 LC_TELEPHONE,
> and 4.12 LC_MEASUREMENT

>
> These categories were added in response to the Japanese comments
> on CD 14652. The U.S. does not think that the particular
> categories and their definitions for these classes of cultural
> conventions, as specified in this section, have had enough
> exposure, discussion, and justification, to be suddenly added
> and approved at the last minute. Unlike the other categories,
> which at least have a long history of implementation by UNIX
> vendors, these new categories have been created de novo, without
> much apparent input or review.

>
> For example, while it may be logically complete to specify paper
> size in terms of width and height measured in millimeters, it is
> not clear whether that maps well to the actual categories of
> relevance to printer control, for example. Does the millimeter
> measurement (rounded up?, rounded down?) correspond to 8-1/2 by
> 11 (inches), to A4? Did anybody bother to examine categories widely
> implemented in "Page Setup" dialogues in common software?

Page setup dialogues were examined in the design.

> The LC_NAME category introduces another complex syntax of escape
> sequences for specifying name syntax. It is at least plausibly
> complete for most European conventions and for Japanese names,
> but has anybody done the research to see if it handles name
> conventions elsewhere in the world (or even Latin America,
> for that matter), or if it reasonably matches anybody's existing
> implementations of a name formatting abstraction?

>
> The rationales provided for all these new categories in Annex
> B are particularly thin and hardly convincing.

>
> The U.S. is not opposed to the specification of cultural conventions
> in this area -- and in fact believes that they do reasonably lie
> within the scope of 14652. However, the U.S. *is* opposed to
> the addition of detailed syntax specifications for particular
> LC_XXX categories without evidence of due diligence in research,
> analysis, or review of these categories.

There could have been more research on these issues, but Northern European

and Eastern Asian sources have been considered.

> 19. Re 4.13 LC_VERSIONS

>
> The mandatory inclusion of the "language" keyword, which is required
> to be a value for a "natural language, as specified in ISO 639"
> cripples the concept of FDCC-set as a useful construct for
> multilingual applications or other hybrids that may want to mix
> languages or specify behavior at a dialect level, etc., in ways
> not recognized by ISO 639.
>
> It is insufficient to state, as on page 55, that "if required
> information is not present in ISO 639 or ISO 3166, the
> relevant Maintenance Authority should be approached to get
> the needed item registered." That, of course, presupposes that
> the kinds of categories that are acceptable for registration in
> those standards match the user requirements for cultural conventions--
> and that is exactly *not* the case for dialectal, bilingual, or
> multilingual conventions.
>
> At the minimum, the specification of LC_VERSIONS should point out
> this limitation to FDCC-set definition, since it does not, in
> principle, appear to be fixable given the structure of FDCC-set's.

Accepted. some care could also be given to world-wide FDCC-set.

> 20. Re 5 CHARMAP

>
> On page 58, FCD 14652 states "The encoded values associated with each
> member of the portable character set shall be invariant across all
> FDCC-sets supported by the application."
>
> This would seem to disallow applications which support both ASCII
> and EBCDIC encodings. A note should be added to the text to
> either explicitly state so or to state that that is not true and
> why.

accepted. It makes implementations both catering for ascii and ebcidic unspecified.

> Note that the statement in FCD 14652 is descendant from the rather looser
> statement in the X/Open specification:

>
> "If the encoded values associated with each member of the
> portable character set are not invariant across all locales
> supported by the implementation, the results achieved by
> an application accessing those locales are unspecified."
>
> The X/Open wording seems more correct, in that it does not
> prohibit implementations to make use of ASCII and EBCDIC, but it
> also does not specify that implementations must be able to do
> so, nor that they have specified behavior if they access locales
> so defined.

The 14652 could be made more aligned with the X/Open spec.

> 21. Re 5.1 Character Set Description Text

>
> On page 59, the declarations for <escseq>, <addset>, and <include>
> were added specifically and explicitly in order to support
> ISO 2022 shifting in a character set description. These are easily
> the most complex part of the character set description syntax,
> yet no exemplification is given, nor is their any justification
> given for why ISO 2022 profiling must be describable in the
> FDCC-set. At a minimum, a full exemplification of the use of
> these declarations for one or more real examples such as
> 2022jp **must** be provided in this section of 14652.

accepted.

> 22. Re 6 REPERTOIREMAP

>
> The U.S. comments to the CD 14652 stated:

>
> "This list is arbitrarily chosen, and the principles for
> characters in it are unstated. If the repertoire file is
> not going to correspond to one of the named and numbered
> subsets of ISO/IEC 10646 (and Subset 300, the BMP, would
> be the obvious choice), then the choice of characters
> in the repertoire file **must** be justified in 14652."

>
> The Canadian comments also pointed out that the repertoiremap
> was incomplete.

>
> The disposition of comments stated: "partly accepted. The list of
> characters corresponds to prior art on the works of POSIX
> locales, and it is included to facilitate reuse of locale
> data already in use. There will be an explanation to this
> effect in the rationale."

>
> The revised text states, in toto:

>
> "The 'il8nrep' repertoiremap is defined to accomodate prior art."

>
> This is a classic example of a non-explanation explanation.

>
> The list is still arbitrarily chosen. There is still no clear
> justification why anybody should be making use of a repertoiremap
> so chosen, nor why the particular collection of duplications in
> symbols is justified in an international standard. Is this
> all just to ensure that some existing LINUX implementation
> has its repertoiremap grandfathered into the standard without
> review of how that was developed in the first place?

>
> The U.S. objects to the particular collection of useless
> and arbitrary symbol names coined helter-skelter and with
> no real mnemonic value. The international standard 14652
> should make use of either the 10646 names of characters or
> the 10646 short character names (Amd 9). If other short,
> symbolic names are required, beyond those which may already
> be in widespread UNIX locale usage for the portable character
> set, then some other widely adopted and useful set of symbolic
> identifiers such as SGML/HTML entity names should be used,
> instead of a completely arbitrary new set which is confusing

> and anti-mnemonic to boot.

There can be added more rationale. The list is for use with existing locales and reflects use in national bodies, X/open and POSIX-2.

> Even as a reference list for the bad mnemonics, the
> repertoiremap doesn't work, since it is listed in UCS
> order. There is no reasonable way to find an arbitrary
> symbol in the table like "<y+;>" or "<)I>" or "<dh>" unless
> you already *know* where to look. <:->

You can locate the wanted name by an editor.

> And the U.S. has particular questions about the repertoiremap
> definition:
>
> Why is <(JU)> U+321C PARENTHESES HANGUL CIEUC U included, but
> no other parenthesized Hangul or Katakana character??
>
> Why are <am> U+33C2 SQUARE AM and <pm> U+33D8 SQUARE PM included
> but no other compatibility square alphabetic characters?
>
> Why are the Old Church Slavonic Cyrillic characters included,
> but not other Cyrillic extensions?
>
> Why are Hebrew points omitted, when Arabic points are not?
>
> Why are Arabic compatibility positional variants included,
> when Japanese halfwidth and fullwidth forms are omitted?
>
> Why are Japanese hiragana and katakana included, but no
> kanji from the Unified Repertoire and Ordering of 10646?
> (And this despite the fact that the LC_CTYPE definition
> refers to them all??)

The kanji is included via Uxxxx names

> Why are ISO 6937 combining characters included (and assigned to
> *private use* code values in 10646 short form, in a normative
> standard!), when 10646 combining characters are systematically
> omitted?

Combining characters will be changed to outside private use.
UCS combining characters did not occur in any other coded character sets
than UCS at the time of the specification of the mnemonics..

> The U.S. reiterates its comment that this kind of arbitrary
> choice makes no sense in 14652, and that the "i18n" repertoiremap
> should logically consist of Subset 300, the BMP, of 10646,
> with only those additional character symbols defined as are
> truly in widespread use already.

The list of mnemonics reflects widespread use. They are in use
in millions of computers today.

> Incidentally, the actual list provided in Clause 6 for the
> "i18n" repertoiremap seems directly at odds with the statement

> made on page 104:
>
> "This standard defines a FDCC-set defined on the character repertoire
> of ISO/IEC 10646 standard, in a character set independent way."
>
> The repertoiremap should be corrected to make it accord with
> this statement in fact.

The specification of the mnemonics does not conflict with the standard being character set independent.

> 23. Re REPERTOIREMAP (C1 characters)
>
> The U.S. also objects to the inclusion of C1 characters in the
> definition of the "i18n" repertoiremap. These presuppose mapping
> in a particular set of control functions, when unlike the C0
> control functions, there is no widespread and universal agreement
> about what these should be.
>
> The disposition of comments on the earlier U.S. comment on this
> issue stated:
>
> "10646 does contain the ISO 6429 control characters per the
> normative inclusion of this standard."
>
> The U.S. objects to that resolution of comments. ISO 10646,
> Clause 8, per Amd 3 states that "Code positions 0080 to
> 009F are reserved for control characters." 10646 does *not*
> specify what those control characters are. 10646 *does* state
> that when used in the context of ISO/IEC 2022, how escape
> sequences are to be used to identify C1 sets of ISO/IEC 6429.
> But no such set is implied by default or explicitly by 10646.
>
> It is fundamentally wrong for ISO 14652 to normatively declare
> a particular C1 set in a repertoiremap, when no such set is
> implied by common usage nor normatively by 10646 itself.

10646 incorporates normatively 6429 thus defining C0 and C1.
6429 can be considered the default for C1.

> 24. Re B.1.2.2 awk script for "reorder-after" construct
>
> The rationale for this awk script is not provided. It
> claims to "implement" the "reorder-after" construct.
>
> What it looks like it does is read the source file for
> a FDCC-set definition, perform a physical reordering of
> the lines in the LC_COLLATE section based on the reorder-after
> commands, and produce a new source file with the lines
> reordered (including any required inclusion of an LC_COLLATE
> section from a copy command). Is this kind of cut and paste
> what it means to "implement" the "reorder-after" construct?
>
> If so, at the very least, that should be explained in this
> informative section, and the code should be commented.
> It is inexcusable to publish uncommented code as part of
> a standard, especially awk script code making use of

> non-obvious identifiers.

Accepted.

> =====

>

> Editorial Comments

>

> 1. page 5, line 2. "defines following categories" -->

> "defines the following categories"

accepted.

> 2. page 6, In section 4.1.1 "Character Representation", in the paragraph

> numbered (2), the sentence starting with "Outside strings" should be

> terminated with a period, after words "the character itself".

accepted.

> 3. page 8, In section 4.1.2.1 "comment_char", the words "All examples this

> standard" should be "All examples in this standard".

accepted.

> 4. page 9, lower, line 4. "my be omitted" --> "may be omitted"

accepted.

> 5. page 9, space, line 1. "for to find" --> "to find"

accepted.

> 6. page 11, In section 4.2.1 "Basic keywords", definition of "class",

> explanation for "segment_separator": "delimits" should be "delimit"

> (plural form of verb).

accepted.

> 7. page 11, In section 4.2.1 "Basic keywords", definition of "class",

> explanation for "block_separator": "delimits" should be "delimit"

> (plural form of verb).

accepted.

> 8. page 11, In section 4.2.1 "Basic keywords", definition of "map",

> explanation for "tosymmetric": "eachother" should be "each other".

accepted.

> 9. page 11, In section 4.2.1 "Basic keywords", definition of "map",

> explanation for "tosymmetric": "mapping form" should be "mapping

> from".

accepted.

> 10. page 13, In section 4.2.2.1 "Transliteration statements", paragraph

> starting with "The order the <transliteration-strings> is defined":

> "is defined" should be "are defined".

accepted.

> 11. page 24, Section 4.2.3 "i18n LC_CTYPE category", map "tosymmetric":
> There should be escape characters (/) at the end of each line
> except the last one.

accepted.

> 12. page 26, Section 4.3 "LC_COLLATE", about "Toggling keywords": there are
> tabulation problems in the lines for "else" and "elif".

accepted.

> 13. page 26, 27, Section 4.3.1 "Collation statements". In the paragraph
> starting with "The ellipsis symbol ("...") specifies", in the last
> sentence, there are 2 occurrences of "ellipses". It is not clear if
> it should be "ellipsis" or "ellipses".

The first is plural, the second is single. (ellipses, ellipsis)

> 14. page 27, Section 4.3.1 "Collation statements". 2nd paragraph, line 1
> The sentence: "The symbolic ellipses (".." or "....") specifies that a
> sequence collating statements." is meaningless. Fix it!!

<collating-element> changed to <collating-identifier> describing the first
operand.

> 15. page 27, Section 4.3.1 "Collation statements". In the paragraph
> starting with "The symbolic ellipses (".." or "....")": replace
> "higher then" by "higher than".

accepted.

> 16. page 30, Section 4.3.5, 2nd paragraph, line 5.
> "with the LC_COLLATE category" --> "within the LC_COLLATE
> category" ??

accepted.

> 17. page 44, In section 4.6 "LC_TIME", explanation for "day": The
> field descriptor should be "%A" and not "%a".

accepted.

> 18. page 47, Section 4.6.1 "Date Field Descriptors": there are tabulation
> problems on the lines for %f, %j, %A.

accepted.

> 19. page 51, Section 4.9 "LC_NAME": there are tabulation problems
> on the lines for %f, %l, %t.
> Not all items are terminated with a period.

Accepted.

- > 20. page 52, Section 4.10 "LC_ADDRESS": there are tabulation problems
- > on the lines for %f, %t.
- > Many items are not terminated with a period.

Accepted.

- > 21. page 53, Section 4.10 "LC_ADDRESS", in the "i18n" listing:
- > there appears to be a superfluous <%> at the end of the first line for
- > "postal_fmt", just before the slash.

accepted

- > 22. page 53, Section 4.11 "LC_TELEPHONE", in the explanation of %a
- > and %A, "are" should be "area".
- > There is a tabulation problem in the line for %l.

accepted

- > 23. page 54, Section 4.13 "LC_VERSIONS", in the first sentence: "defines
- > which specifications methods that have been used" should be "defines
- > which specifications methods have been used".
- > There is a tabulation problem in the line for "tel".

accepted in principle, see also the Japanese comment.

- > 24. page 58, Section 5.1 "Character Set Description Text", in the
- > explanation for <code_set_name>, "taken form" should be "taken from".

accepted

- > 25. page 59, Section 5.1 "Character Set Description Text", in the
- > explanation for <repertoiremap>, "taken form" should be "taken from".

accepted

- > 26. page 59, Section 5.1 "Character Set Description Text", in the
- > explanation for <escseq>, replace "what range of characters in the
- > charmap that is affected" by "what range of characters in the charmap
- > is affected".

accepted

- > 27. page 59, Section 5.1 "Character Set Description Text", in the
- > explanation for <include>, replace "what range of characters in the
- > referenced charmap" by "a range of characters in the referenced
- > charmap".

accepted.

- > 28. page 98, In section "Annex A", first paragraph, "comformant" should
- > be "conformant".

accepted.

- > 29. page 99, Section A.2 "Enhancements", paragraph 12 starting with
- > "The <Uxxx> and <Uxxxxxxxx>", the clause "together with a number
- > symbolic character names derived from POSIX" is not comprehensible

> (and also seems to be grammatically incorrect). It should be corrected.

Add "notations" after first word.

> 30. page 99, Section A.2 "Enhancements", paragraph 10.
> "elipsises" --> "ellipses".

accepted.

> 31. page 99, Section A.2 "Enhancements", paragraph 14 starting with "New
> categories": "has been" should be "have been".

accepted.

> 32. page 99, Section A.2 "Enhancements", paragraph 16 starting with "The
> digit keyword": "support" should be "supports".

accepted.

> 33. page 99, Section A.2 "Enhancements", paragraph 18 starting with "The
> LC_TIME has got": "calendar" should be "calendar".

accepted.

> 34. page 100, Section B.1 "FDCC-set Rationale": the last paragraph mentions
> a "grandfather clause". This metaphor is not in general international
> English usage. Is it possible to substitute a
> more direct expression?

Standards and industry practice can be referred to.

> 35. page 101, Section B.1.1 "LC_CTYPE Rationale", last paragraph: replace
> "The definition of character class digit allows that alternate digits
> (e.g., Hindi or Ideographic) can be specified here." by "The
> definition of character class digit allows alternate digits
> (e.g., Hindi or Ideographic) to be specified here."

accepted.

> 36. page 103, Section B.1.2 "LC_COLLATE Rationale", next to last paragraph
> starting with "The character": replace "elements defines" by
> "elements define".

accepted.

> 37. page 106, Section B.1.2.3 "Sample FDCC-set specification for Danish":
> the line after "reorder-after <CAPITAL>" says "<CAPITAL>". This seems
> strange, like removing <CAPITAL> then reinserting it exactly
> at the same place. Should this line be removed?

no. This provides a complete self-contained specification.

> 38. page 111, Section B.1.5 "LC_TIME Rationale", third paragraph starting
> with "The field descriptors": there is an unwanted line break between
> "the traditional" and "field descriptor".

accepted

>
> 39. page 113, Section B.2 "Character Set Rationale", fifth paragraph
> starting with "The charmap was introduced": replace "an application or
> an application" by "an application".

accepted.

>
> 40. page 114, Section B.2 "Character Set Rationale", next to last paragraph
> starting with "The charmap allows": replace "for example as a fully
> composed character and as a base character" by "for example a fully
> composed character and a base character".

Accepted. The text will be replaced with the following text:

"...This allows for encodings that can encode items in more than one way. For example, an item can be encoded once as a fully composed character and again as a base character plus combining character. This would allow either representation to be recognized. As only the first occurrence of the character may be output, this technique could be used to normalize a character stream."

End of disposition of comments.