ISO/IEC JTC1/SC22/WG20 N568R

Title: **Disposition of comments on ballot JTC1/SC22 N2719**
ISO/IEC FCD 14651, International String Ordering

Date: **1998-08-17**

Project: **JTC 1.22.30.02.02**

Cross References: **SC22 N N2719, SC22/WG20 N 567 (clone of the first one), SC22 N 2607**

Source: **Alain LaBonté, Project editor, on behalf of SC22/WG20**

Status: **Information required according to directives by SC22 Secretariat**

Action: **For national bodies consideration**

Summary of voting

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	5
"P" Members abstaining	3
"P" Members not voting	2
"O" Members supporting approval without comments	1
"P" Members not supporting approval	1

Secretariat Action:

WG20 is requested to prepare a Disposition of Comments Report and make a recommendation on the further processing of the FCD.

The comments accompanying the affirmative votes from Canada and Ireland, and the negative votes from Denmark, Japan, Netherlands, Sweden, United Kingdom and the USA are attached.

The comment accompanying the abstention vote from Australia was: "Not enough time to consider this proposal due to the late arrival of the document." The comment accompanying the abstention vote from Austria was:
"Lack of expert resources." The comment accompanying the abstention vote from Germany was "There is no national rapporteur."

Secretariat comment: The document has been available to all SC22 Member Bodies either electronically or in paper form since December 1997.

Disposition of comments from Canada

Canada APPROVES WITH THE FOLLOWING COMMENTS:

Canada votes YES on FCD ballot on 14651, with the following comments:

1. On the condition that normalizing characters to either precomposed sequences or decomposed sequences for characters which have double coding (called canonical equivalences according to the Unicode standard) is made mandatory, Canada is of opinion that combining marks should have a second level equivalent to the precomposed forms' second level (ex. precomposed character é [E ACUTE] has a weight for the acute accent at the second level. The character COMBINING ACUTE should have the same second level weight). This comment is to solve the problem raised by the Unicode consortium about canonical equivalences. However we insist that this scheme technically works only if normalization of character sequences is mandated, a direction to which the Unicode consortium now agreed and added/specified in consequence in its own algorithm.

CA1. SC22/WG20 experts consider that the standard should not mandate a particular manipulation of data but rather demand that the comparison behaves as if there would be normalization of character sequences. Equivalences will be dealt with in making that behaviour an obligation.

2. The syntactic mistakes of the table should be fixed and passed through an existing commercial locale parser such as the one of AIX for final checking.

CA2. Accepted.

3. Reestablish the COLL_WEIGHT_MAX=4 after the LC_COLLATE statement. We believe that this parameter should be clarified in 14652 as being a declaration of the maximum number of levels used in the table. This is very useful for not having to parse the table twice, just to determine storage allocation before processing the table. A comment to that effect will also be done for the ballot on 14652.

CA3. As syntax will be considerably simplified for the next FCD, SC22/WG20 experts decided that this was defeating the simplification exercize and this will be left to implementation. Syntax will be compatible but will not retain elements not absolutely essential to the specification itself.

4. The annex I of the previous CD should be reintroduced, however with nonrelevant parameters removed from the example as per our previous Canadian comment.

CA4. SC22/WG20 experts decided to make the compromise that a statement will be made to the effect that fundamental choices should be presented to users.

Disposition of comments from Denmark

Vote from Danish Standards Association on FCD 14651 - Sorting - SC22 N2607.

The Danish member body vote is "no" with comments.

If the technical comments can be solved satisfactorily, the Danish "no" vote can be changed to "yes", unless other significant changes be made to the standard in an unsatisfactory way.

An example of an unsatisfactory change to the standard is the technical annex in SC22 N2608 on an example user interface, which is out of scope of the standard, and does not explain the relation to the APIs and specifications in the standard, which is not trivial. The user interface assumes that a number of tailored sorting specifications be available at runtime, presumably in compiled binary form, and that selecting mechanisms be available.

DK1. Specific input to the editor is expected from the Danish expert present at the Dublin meeting during the second FCD making, so that Denmark can be satisfied as far as possible in respecting the spirit of consensus decisions taken by the experts.

Technical comments.

1. On page 120, an "order_end" and an "END LC_COLLATE" statement is missing.

DK2. Accepted at the beginning of the meeting but later on, the group decided to simplify syntax. After that decision, the *order_end* statement will be added as per this comment. However both the *LC_COLLATE* and *END LC_COLLATE* statement will be removed. That said, ISO/IEC 14652 will be able to build on this simplified syntax which is intended to be totally compatible with what was there before and with current provisions of ISO/IEC 14652. Redundancy in declarations will also be removed (declarations of symbols will be implicit as far as possible).

2. A number of scripts should have a specified ordering.

DK3. Accepted.

3. It needs to be explained that it is not necessary to convert into binary sorting strings to do the comparison. The comparison can be done character by character on the fly, and this may often result in a difference within the first few characters, thus avoiding costly complete conversion of both the strings for comparison. This should at least be explained for COMPCAR.

DK4. The API specification will disappear in the second FCD.

4. It needs to be explained that the binary strings are locale dependent and therefore not adequate for storing, in databases and the like.

A note on this should be available for CHARABIN.

DK5. The API specification will disappear in the second FCD. However a warning should be added at an appropriate place that differently tailored tables will produce incompatible keys.

5. As a number of national specifications have been developed using
specific naming of weights different from the names in the standard
it is necessary to support this existing practise to add these weights
via the "symbol-equivalence" statement of 14652. The weights in question include:

```
collating-symbol <NONE>
collating-symbol <ACUTE>
collating-symbol <ACUTE+DOT>
collating-symbol <GRAVE>
collating-symbol <DOUBLE-GRAVE>
collating-symbol <BREVE>
collating-symbol <BREVE+ACUTE>
collating-symbol <BREVE+GRAVE>
collating-symbol <BREVE+MACRON>
collating-symbol <BREVE+HOOK>
collating-symbol <BREVE+TILDE>
collating-symbol <BREVE+DOT-BELOW>
collating-symbol <BREVE-BELOW>
collating-symbol <INVERTED-BREVE>
collating-symbol <CIRCUMFLEX>
collating-symbol <CIRCUMFLEX+ACUTE>
collating-symbol <CIRCUMFLEX+GRAVE>
collating-symbol <CIRCUMFLEX+HOOK>
collating-symbol <CIRCUMFLEX+TILDE>
collating-symbol <CIRCUMFLEX+DOT-BELOW>
collating-symbol <CARON>
collating-symbol <CARON+DIAERESIS>
collating-symbol <CARON+DOT>
collating-symbol <RING>
collating-symbol <RING+ACUTE>
collating-symbol <RING-BELOW>
collating-symbol <DIAERESIS>
collating-symbol <DIAERESIS+MACRON>
collating-symbol <DIAERESIS+ACUTE>
collating-symbol <DIAERESIS+GRAVE>
collating-symbol <DIAERESIS+CARON>
collating-symbol <DOUBLE-ACUTE>
collating-symbol <HOOK>
collating-symbol <TILDE>
collating-symbol <TILDE+ACUTE>
collating-symbol <TILDE+DIAERESIS>
collating-symbol <TILDE-BELOW>
collating-symbol <DOT>
collating-symbol <DOT-BELOW>
collating-symbol <DOT+DOT-BELOW>
collating-symbol <STROKE>
collating-symbol <STROKE+ACUTE>
collating-symbol <CEDILLA>
collating-symbol <CEDILLA+ACUTE>
collating-symbol <CEDILLA+GRAVE>
collating-symbol <CEDILLA+BREVE>
collating-symbol <OGONEK>
collating-symbol <OGONEK+MACRON>
collating-symbol <MACRON>
collating-symbol <MACRON+ACUTE>
collating-symbol <MACRON+GRAVE>
collating-symbol <MACRON+DIAERESIS>
collating-symbol <MACRON+DIAERESIS-BELOW>
collating-symbol <MACRON+DOT>
collating-symbol <MACRON+DOT-BELOW>
collating-symbol <MACRON+CIRCUMFLEX>
collating-symbol <LINE-BELOW>
collating-symbol <HORN>
collating-symbol <HORN+ACUTE>
collating-symbol <HORN+GRAVE>
```

```
collating-symbol <HORN+HOOK>
collating-symbol <HORN+TILDE>
collating-symbol <HORN+DOT-BELOW>
collating-symbol <PRECEDED-BY-APOSTROPHE>
collating-symbol <GREEK>
collating-symbol <PSILI>
collating-symbol <PSILI+VARIA>
collating-symbol <PSILI+VARIA+PROSGEGRAMENI>
collating-symbol <PSILI+VARIA+YPOGEGRAMMENI>
collating-symbol <PSILI+OXIA>
collating-symbol <PSILI+OXIA+PROSGEGRAMENI>
collating-symbol <PSILI+OXIA+YPOGEGRAMMENI>
collating-symbol <PSILI+PERISPOMENI>
collating-symbol <PSILI+PERISPOMENI+PROSGEGRAMENI>
collating-symbol <PSILI+PERISPOMENI+YPOGEGRAMMENI>
collating-symbol <PSILI+PROSGEGRAMENI>
collating-symbol <PSILI+YPOGEGRAMMENI>
collating-symbol <DASIA>
collating-symbol <DASIA+VARIA>
collating-symbol <DASIA+VARIA+PROSGEGRAMENI>
collating-symbol <DASIA+VARIA+YPOGEGRAMMENI>
collating-symbol <DASIA+OXIA>
collating-symbol <DASIA+OXIA+PROSGEGRAMENI>
collating-symbol <DASIA+OXIA+YPOGEGRAMMENI>
collating-symbol <DASIA+PERISPOMENI>
collating-symbol <DASIA+PERISPOMENI+PROSGEGRAMENI>
collating-symbol <DASIA+PERISPOMENI+YPOGEGRAMMENI>
collating-symbol <DASIA+PROSGEGRAMENI>
collating-symbol <DASIA+YPOGEGRAMMENI>
collating-symbol <VRACHY>
collating-symbol <GREEK-MACRON>
collating-symbol <VARIA>
collating-symbol <VARIA+PROSGEGRAMENI>
collating-symbol <VARIA+YPOGEGRAMMENI>
collating-symbol <OXIA>
collating-symbol <OXIA+PROSGEGRAMENI>
collating-symbol <OXIA+YPOGEGRAMMENI>
collating-symbol <PERISPOMENI>
collating-symbol <PERISPOMENI+PROSGEGRAMENI>
collating-symbol <PERISPOMENI+YPOGEGRAMMENI>
collating-symbol <TONOS>
collating-symbol <PROSGEGRAMENI>
collating-symbol <YPOGEGRAMMENI>
collating-symbol <DIALYTICA>
collating-symbol <DIALYTICA>
collating-symbol <DIALYTICA+VARIA>
collating-symbol <DIALYTICA+OXIA>
collating-symbol <DIALYTICA+PERISPOMENI>
collating-symbol <DIALYTICA+TONOS>
collating-symbol <CYRILLIC>
collating-symbol <HIRAGANA>
collating-symbol <KATAKANA>
collating-symbol <SPECIAL>
collating-symbol <LAST>

collating-symbol <CAPITAL>
collating-symbol <CAPITAL-SMALL>
collating-symbol <SMALL-CAPITAL>
collating-symbol <SMALL>
collating-symbol <SUPERSCRIPT>
collating-symbol <SUBSCRIPT>
collating-symbol <OTHER>
```

DK6. The second FCD will be decoupled from ISO/IEC 14652 although the simplified syntax will remain compatible. It will be possible, for those eventually following ISO/IEC 14652, to define symbol equivalences. Symbols will be redefined in the second FCD so that automatic maintenance of the table in future editions of the standard (according to development of the UCS) be done automatically as far as possible. This means that most symbols will become numeric in the second FCD, and will be tightly coupled to UCS IDS for character definitions.

6. It is wrong to define weights two times. Weights <a8> etc.
on page 44-51 needs to be deleted.

DK7. Accepted.

Editorial comments:

1. The standard should use established weight names such as
the ones in ISO/IEC 9945-2 annex G and many POSIX locales, and
not reinvent the wheel.

DK8. Not accepted for the same reason as in DK6.

2. The annexes should use normal ISO numbering, and normative annexes
should be using numbers "A" and "B" etc.

DK9. Accepted.

3. a reference should be made to ISO/IEC 9899 programming language C,
possibly in the normative references.

This concludes the DS ballot text on SC22 N2607.

DK10. API specification will disappear so this comment will not be applicable any more.

Disposition of comments from Ireland

****** Ireland approves with the following comments

_____ general

****** Ireland believes that this standards is important and timely. However, we see a need for harmonization with work done in this area by the Unicode Consortium where possible and practicable, in order to increase the likelihood that standard will actually be used.

IE1. Accepted.

_____ technical

****** Normalization of characters to either recomposed sequences or decomposed sequences should be made mandatory.

IE2. SC22/WG20 experts consider that the standard should not mandate a particular manipulation of data but rather demand that the comparison behaves as if there would be normalization of character sequences.

****** Errors in the specification table (for instance, in Cyrillic and elsewhere) should be corrected.

IE3. Accepted.

****** A version of informative Annex I of the previous CD should be put back into the standard.

IE4. SC22/WG20 experts decided to make the compromise that a statement will be made to the effect that fundamental choices should be presented to users.

_____ editorial

****** The entire English text needs to be revised for language.

IE5. Accepted. SC222/WG20 has nominated Michael Everson coeditor of the standard for the English version. Michael Everson will correct the English language second FCD draft provided by the editor.

Disposition of comments from Japan

SC 22 N 2607: FCD 14651
Method for Comparing Character Strings and the Description of
Common Template Tailorable Ordering.

(X) Disapproved

National Body: Japan
Date: 1998-02-17
Signature: KATSUHIKO KAKEHI

The National Body of Japan disapproves FCD 14651 for the reasons below.

If the comments are satisfactorily resolved, Japan will change its vote to approval.

J.1) Considering the high dependency of this FCD on ISO/IEC 14652, Japan requests to wait and synchronize the review of FCD 14651 until FCD 14652 is approved.

JP1. Dependency on ISO/IEC 14652 will be completely eliminated.

J.2) The concept of the common template is not consistent in this draft as follows:

a) on one hand, the draft defines that the common template is not a default ordering, in the following text:

a.1) the text in 7.1 Data specification --

> Although the data in table 1 defines a specific ordering, it should not be considered as an internationally acceptable default ordering but only as a starting point for the tailoring process prescribed in the next paragraph.

a.2) the content of the common template in Annex 1 is designed so as to invoke an intentional mistake if it is parsed without tailoring -- defining some toggle switches.

b) on the other hand, the draft describes as if the common template were a default ordering, in the following places:

b.1) the text at the beginning of Clause 5 Requirements --

> Hence the first conformance level (conformance level A) is limited to a comparison API using the fixed common template specified in annex 1 (table 1) with the result precision fixed;

b.2) the descriptions of the parameter "table" in the procedures COMPCAR and CARABIN --

> - table: the name of the ordering table to be used to make the comparison. This is only required for conformance level C. If this parameter is not provided, the common template will be assumed. ...

Japan's proposals for the problem:

c.1) keep the a.1 and a.2 as it is (except for some detail),

JP2AB. Accepted in principle (except for details due to the major modifications of the standard for syntax simplification [the intentional mistakes will go away]). The tailorability requirement will become even more obvious in the second FCD. Examples of significant (not theoretical) deltas (tailoring) will be given in informative annexes which will apply even to major communities of users.

c.2) change b.1 as to delete the first conformance level

c.3) change b.2 as to invoke the implementation dependent table
if the parameter "table" is not provided.

JP2C. The API specification will be completely removed in the second FCD.

J.3) The relation to ISO/IEC 14652 is not well described.

The last sentence in the first bullet of the Clause 1 Scope should be changed from

> This interface uses transformation tables derived from either the common template provided or from its tailoring.

to the two sentences

> This interface uses transformation tables which conforms to the specifications for LC_COLLATE in ISO/IEC 14652.
> The table shall be derived from either the common template provided or from its tailoring as defined in this standard.

JP3. See disposition of comment JP1.

J.4) The term "prehandling" should be applied to a process handling character strings to be compared as defined in Clause 3.

Japan's proposal:

1) Change the title of Subclause 5.1 from "prehandling phase" to "preprocessing phase" or "preparation phase"

2) Change the title of Subclause 5.1.1 from "prehandling of the symbolic table data" to "preparation of the symbolic table data".

3) Change the first sentence of the fourth paragraph of 5.1.1 from

> As prehandling can be done on many different tables in a given application environment, each one shall be identified by a name in this environment

to

> As preparation may be done to generate more than one table in a given application environment, each one shall be identified by a name in this environment

JP4. Text will be moved to an informative annex.

J.5) In splitting the functionality of COMPCAR into COMPBIN and CARABIN,
the concept of "precision" is introduced in COMBIN in this draft.

But adding the parameter "precision" also in CARABIN will produce a
comparable bit string shorter than the one for full precision and
those shorter strings are preferable in many situations.

JP5. See disposition of comment JP2C.

J.6) The definition

> collating symbol a symbol used to specify weights assigned
> to a character in a symbolic fashion rather than absolutely

is not understandable. What word does the word "absolutely" correspond?

JP6. The definitions will be revised. This one will probably go away, as syntax will be considerably simplified. The term "absolutely" will otherwise be changed to "numerically".

J.7) The definitions for "first order token", "second order token",
"third order token", and "fourth order token" should be merged to

> n-th order token: an absolute number used as a comparison element
> corresponding to the n-th column of a line in the table.
>
> In the typical case of Latin characters, there are
> four levels of tokens and the first three levels of token
> describes a character (note that some
> characters, such as ligatures, may lead to more than one token
> for a character at one given level) while the fourth order token
> describes a special character, or specifying the position of the
> special character represented in the original string;
> tokens of the fourth order level are always in pairs, the first
> token being a position,
> the second one being a weight for the character represented.

JP7. Accepted.

J.8) The symbols and abbreviations introduced in Clause 4 seems to appear
only in Annex 1 and the second, fourth and fifth paragraphs are not
appropriate for "Symbols and abbreviations". It is recommended to put
the material into subclause 5.4 or Annex 1.

JP8. Accepted.

J.9) The sentence in Clause 4

> If a character outside of the standard repertoire of ISO/IEC 10646
> is to be used in tailored ordering tables, it is recommended that
> the code-independent symbol identifying this character use the form
> <U8XXXXXXX> for documentary purposes indicating its nonstandard
> nature

is not understandable for many readers and apt to be upset by future
ISO/IEC 10646 revisions. We propose to use the form <!UXXXXXXXX>
where '!U' means "not 10646".

JP9. The group finally agreed to use form PXXXXXXXX for private-use characters.

J.10) Subclause 5.1 is not in the scope of the standard. It should be moved to a new informative annex.

JP10. Prehandling considerations will remain but most of it will be moved to an informative annex already treating that subject. See disposition of comments CA1 and IE2.

J.11) The first sentence of the second paragraph of subclause 5.2.1:

"consists in" -> "consists of"

JP11. English will be completely revised by coeditor of the English version Michael Everson.

J.12) 5.2.1.1, Input parameters:

The description

- string1: referenced character string that is to be compared to the reference string (string2)

- string2: reference character string that is to be compared to the referenced string (string1)

should be changed to

- string1: referenced character string that is to be compared to the reference string (string2)

- string2: reference character string -- string serves as a base reference for comparison

J.13) 5.2.1.1, Output Parameter:

The name of the parameter "return" should be changed to "result".

J.14) 5.2.1.2:

The changes similar to J.12 and J.13 should be done.

J.15) 6. Conformance:

The definition of the conformance level here is different from the one in Clause 5. It should be aligned.

JP12-15. See disposition of comment JP2C.

J.16) 7.1, Data Specification, 2nd para.

The sentence

> Although the data in table 1 defines a specific ordering, it should not be considered as an internationally acceptable default ordering but only as a starting point for the tailoring process prescribed in the next paragraph.

should be changed to

> The data in table 1 does not define a specific ordering. It should not be considered as an internationally acceptable default ordering but only as a starting point for the tailoring process prescribed in the next paragraph.

JP16. Accepted in principle.

J.17) Annex 2

It should be changed to informative with changing the first sentence from

> The benchmark shall be tested in using the default table of annex 1, adapted in defining the following toggles: PRIMNBSP, ACCENTSONLATIN, PRIMIN

to

> The benchmark should be tested when the default table of annex 1 is tailored by defining the following toggles: PRIMNBSP, ACCENTSONLATIN, PRIMIN.

JP17. Bennchmark will be informative and toggles will go away due to the specification of the syntax. The intent of the specific toggles of 1st-FCD-14651 is respected even in simplified syntax altough these exceptions will be removed in the simplification.

J.18) (not on FCD 14651) Japan supports the decision at Quebec meeting to eliminate the informative annex I in the first CD draft because the annex misleads readers to think tailoring is a very complex process that almost always needs a visual interface.

JP18. SC22/WG20 experts decided to make the compromise that a statement will be made to the effect that fundamental choices should be presented to users.

Disposition of comments from the Netherlands

The NNI votes NO on this FCD 14651 for the reasons detailed hereafter:

-1- The NNI is of the opinion that this document shows insufficient
maturity to warrant an _F_CD.

NL1. Accepted.

-2- The structure, contents and wording of this document is considered to
be below standards. This is considered unacceptable.

NL2. Although the text is the result of years of compromise and is the result of a collective effort, it will be considerably simplified and will continue to be structured according to the spirit of the latest JTC1 directives.

-3- The editor has ignored the comments on the CD from the NNI.
This is equally unacceptable.

NL3. Unaccepted. The comments of the NNI had been considered by the working group and the editor had faithfully reflected these considerations. The disposition of comments had been approved collectively and duely to avoid such comments.

The NNI will change its vote into a yes vote only when all three issues mentioned above and detailed hereafter will have been resolved satisfactorily.

NL4. SC22/WG20 experts take good note.

re 1:
Given the number of comments on the previous version of this document, the NNI considers the stability of content and form of this document insufficient to warrant its circulation as a FINAL CD.
Given the comments detailed hereafter, the NNI requests that at least one other CD be circulated before another FCD can be attempted.

NL5. Accepted as in NL1 above.

re 2:
The algorithm described has been partially detailed in the definition clauses and in _in_formative annexes.
The normative body of the document does not contain a sufficient description of the algorithm. The annexes have been formulated as examples only.
Although it will be not easy to construct an appropriate abstract definition of the algorithm, the document is considered incomplete without this.

NL6. Accepted in principle.

The document contains English text that clearly has been translated from French. Many remains from the French text are still visible.

These remains make the document difficult to understand.

NL7. English will be revised by coeditor of the English version, Michael Everson.

re 3:
In our comments on the CD the NNI stated that:
"The text of this standard leaves much to be desired
regarding precision of definition, clarity of presentation ...
The NNI cannot give detailed comments here, nor offer replacement text
as doing so would require rewriting more than half of the
document for which we have no resources available.

NL8. There will be a second FCD considerably simplified, and the API specification, which causes most of the problems raised by NNI, will be completely removed so that consensus be achieved, and NNI satisfied as far as possible.

The NNI already gave some directions with its vote on the CD-registration,
but found alas almost no improvement in this CD"
The Disposition of Comments sc22n2466 completely ignored our comment
by stating " This is purely ITTF matter ...."
However, sharpness of definition and clarity of presentation is not
an ITTF matter at all. It is the main task of a responsible editor.

NL9. Not accepted. It is the task of national bodies to provide replacement text (in particular after many iterations) and to participate when there is interest in a project of standard. Otherwise it is a vicious circle, interpretations of the NB comments have to be made remotely without any possibility to ask for explanations, and these interpretations might not reflect what a face to face presence could have allowed to solve much more efficiently.

The NNI feels very disappointed that apparently the editor of this document is unable to write clearly (see below), is unfamiliar with programming language terminology and is unfamiliar with the mathematics behind programming languages but is even more so unable to read and understand the comments (not only ours, as can be seen clearly from the DOC).

NL10. SC22/WG20 experts take good note.

As we felt these shortcomings in definitions and presentation to be clearly visible to the intelligent reader, we refrained from enumerating them.
This seems however not to have worked, so this time we will give a _few_ examples from the clauses 3 and 5 of the document.
Please note that there is no use in making local repairs to the document.
The document should be totally checked and revised.

Comments will be given in-line with the text using question marks or on separate lines preceded by a hash mark and a number.

Before starting with our comments on clauses 3 and 5 it should be noted that 7 out of the 28 notions defined in clause 3 have not been used in the document at all.

NL11. SC22/WG20 experts take good note.

3 Definitions

API

Application Program Interface defined as a
standardized application process describing the
specifications of different procedures ...

#001# How can an API be defined as a STANDARDIZED APPLICATION PROCESS?
It seems the editor is unfamiliar with the meaning of API and/or
APPLICATION.

#002# What is a process here; what is the difference between process
and procedure?

NL12. This definition will go away as all references to APIs will disappear in the second FCD.

character string

a series of characters concatenated in
logical sequence

#003# It is unclear what makes a logical sequence; does the logical sequence
of characters in a string differ from their physical sequence?

#004# This definition uses concatenation as if it was defined
with the following signature:
Concatenation : Char x Char x Char x Char ... -> CharString
Thus is VERY unusual.

NL13. If this definition is ever preserved, it will be changed to "a sequence of characters considered as a single object".

collating symbol

a symbol used to specify weights assigned to a
character in a symbolic fashion rather than absolutely

#005# Absolute contrasts relative; symbolic contrasts real.

#006# In SC22 terminology "to assign" means that variables will be involved;
this however, seems not to be the case.

#007# What does it mean to "assign something in a symbolic fashion"?

#008# What does it mean to "assign something absolutely"?

NL14. If this definition is ever preserved, it will be changed to "a symbol used to specify weights attributed to a character"

collating element

the smallest entity used to determine the logical
ordering of strings.

#009# What is a logical ordering of strings? how does it differ from their
physical ordering?

It normally consists of either a single character
or two or more characters collating as a single entity

#010# This seems to mean that two strings can be ordered by one
single character?

#011# The "entity" at the beginning of this sentence seems to be of a different
kind than the "entity" at the end of the sentence.

#012# What is the "not-normal" situation?

NL15. If this definition is ever preserved, it will be changed to "a single weight or a series of weights attributed to a character at a specific level of ordering"

concatenation

logical operation which consists in adding an element
at the end of a string to consider the result as a new

string, longer than the first one.

#013# What is a "logical" operation? Does it involve the logical connectives
'and', 'or', 'not'?
#014# In what sense is an "operation" different from a "process"?
#015# Concatenation has here been defined as of type:
Concatenation : X-string x X -> X-String
Instantiating X by Char gives a definition different from the earlier one.

NL16. This definition will be removed.

equivalence
in a comparison between two character strings,
a form of partial equality between the two strings.
After decomposition of the two strings in different
levels according to the ordering table ...
#016# It is unclear that the ordering table describes a decomposition
#017# The ordering table is called elsewhere in this document:
- common template ordering table
- Common Template order description
- Common Template table
- common template
- International Common Template Table

.... The typical case ...
#018# It is believed that examples should not be used in a definition clause

NL17. This definition will go away. Consistency for naming the table will be done as far as possible. The inconsistencies are due to multiple modifications of the text over years demanded by national bodies comments.

field for the purposes of this International Standard,
a single character string or any other data type
which may be ordered alone(?) or in conjunction
with other fields of a record, each field of a record
being compared to the same field of another record;
#019# Is "a single character string" the same as
"a string containing one character" or
"a single string containing characters"?
#020# It is unclear how such a string can be ordered "alone"
#021# It is unclear what a "record" is; see later comment
#022# Precisely the "same" field? Perhaps corresponding?
#023# A data type (in SC22 terminology) can not be ordered; values of a data
type may perhaps be ordered.
in case of absolute equality of two equivalent fields,
other fields of the records ....
#024# Yet another definition of equivalent?
#025# What is absolute equality?
#026# It is believed that is not proper to distribute the description
of the algorithm throughout the document.

NL18. This definition will go away.

first order token
an absolute number used as a comparison element,
obtained out of tables for the first level that
describes a character; ...
#028# What kind of element is this?
#029# Absolute number; does this have the mathematical meaning of |x| ?
#030# It is unclear to what part of the sentence "that" refers to;
the number, the element or the level?
#031# Does a level "describe" a character?

fourth order token
an absolute number ....., obtained either out of,
or specifying the position ....;
tokens of the fourth order level are always in pairs
the first token being a position, the second one
being a weight ...

#032# The first sub-sentence states that a fourth order token consists of "one" absolute number; the second sub-sentence states that a fourth order token is a pair.

#033# The first sub-sentence states that a fourth order token is an "either-or" the second sub-sentence states that it is a pair.

level
Whenever used ....level .."key level" .."precision
level" ... conformance level...
normally ...

#034# Why such a difficult terminology? Why not just use precision level and conformance level throughout?

#035# What is the abnormal situation?

#036# It is again believed that it is not appropriate to detail the algorithm here.

NL19. These definitions will be simplified, made consistent and aligned with Japanese comments. See disposition of comments JP7. NNI questions will be considered.

numeric relative value
the relative .. in its final numeric and processable
form

#037# Are there non-final forms?

#038# Are there unprocessable forms?

NL20. This definition will go away.

ordering key
a series of bits, the numerical value of which
determines its order;
to a character a ..... characters

#040# The second sub-sentence seems unconnected to the first one and is again discussing the algorithm

NL21. If this definition is ever preserved, it will be changed to "a series of numeric values used to determine an order"

posthandling
a process in which ... internally ... straightforward
comparisons ...

#041# What is a process?

#042# What is meant by internally? Internal to what?

#043# Why are the comparisons done according to this standard "straightforward"?

NL22. This definition will go away.

record
the exhaustive structured set of fields that
form a monolithic block in a file ....

#044# What is exhaustive here?

#045# In mathematics sets are unstructured. What is the structuring criterion?
#046# What is meant by monolithic?
#047# What is a block?
#048# What has a file to do with all this?

NL23. This definition will go away.

reference string .... a base reference ..
#049# Are there also non-base references?

NL24. This definition will go away.

string
a series of individual elements which form a whole,
when they are concatenated ...
#050# It seems that this definition disallows character strings consisting
of a single character

NL25. This definition will go away.

transformation
A(a?)n operation performed prior to comparison or
ordering ...
#051# What is the difference between comparison or ordering?
This definition helps to have a more general discussion on the
terminology used in this standard.
There are three verbs that are central to this standard: to order,
to compare and to collate.
Looking up the meanings of these verbs in this definition clause and
in a dictionary does not bring much light in the question of what the
precise definition used here is.
Looking ahead in the document one finds that the API only allows for
comparisons to be done.
The mechanism is such that first strings are mapped to unique values
(isomorphic to integers; what is called an injection in mathematics).
These values are then compared using the underlying ordering of
the set of integers.
The fact that the strings can then be (re)ordered seems to be outside
the scope of this standard.
Collation has not defined here at all. Looking up its meaning in Webster's
gives again 'to compare' and 'to order', but does not give any insight.

NL26. This definition will go away.

5.1 Prehandling phase ...

This requirement shall be met for conformance levels greater than 1

#052# This requirement -> The requirement specified in this clause ...
#053# Conformance requirements have been indicated by letters!

It is recommended ..
#054# This recommendation conflicts with "shall" in the first sentence.
#055# But, this whole sentence seems misplaced here.

The symbolic table, ...
#056# No symbolic table has been provided in this document
shall be provided in a numeric form
#057# What kind of numeric form? floating point numbers?
.... each element of the matrix being a numerical token indicating
one or more relative weights.
#058# Numerical token has not been defined
#059# Relative weight has not been defined
#060# Can "one" token indicate "more than one" weights?

#061# What is the meaning of "to indicate a weight"? As a balance?

#062# Generally speaking, it seems somewhat of an overspecification to require
that the table be built in a particular(matrix) form (n *m array).
As long as the outcome of the comparison of two strings conforms to the
requirements of this standard, the internal form used for data seems
not significant.

NL27. Prehandling considerations will remain but most of it will be moved to an informative annex already treating that subject. See disposition of comments CA1 and IE2.

However(,?) the values used shall respect the order specified in the
symbolic table data.
#063# No symbolic table can be found in this document.

NL28. SC22/WG20 experts take note. However there is nothing we can do without a better name being proposed for the "symbolic table".

Conformance level C requires that the API shall allow to choose a table
built at a preceding time
#064# What is "a preceding time"? Earlier than prehandling?

It may be necessary to transform a field of a record
#065# Into what?
... or to transform unstructured records
#066# What are "unstructured" records? Please note the definition of records
in clause 3
... into structured fields
#067# Given the definitions of record and field in clause 3 this
definition is incomprehensible
... before the actual process can begin.
#068# Yet another process?
The implementer is responsible for ensuring ...
#068# The implementer of this standard? How can that be?
This is a global operation
#069# What is meant by global?
#070# So an operation is a process?
that may involve "exploding" records before ordering them
#071# How does one explode an "exhaustive structured set" or
a "monolithic block"?
#072# Are these records ordered, or are the strings ordered?
Therefore, the prehandling phase ...
#073# Why a conclusion here?
Thus, prehandling is not part of the comparison operation API.
#074# Why a conclusion here? We already knew that.
The comparison API will not contain any default method related to prehandling.
#075# What is a default method?
#076# Isn't this derivable from the previous sentence?
...for allowing the use of this IS in higher layers of the application?
#077# It seems inappropriate for this standard to state anything about the
structuring of an application using this standard.

The prehandling phase ... transform the actual coded characters used on input
#078# In the previous section it was said that prehandling operated on records?
....Then the prehandling phase can correspond to the empty process
#079# So a phase is equal to a process?
Ideally, all control characters
#080# Ideally = shall/recommended/optional ?
#081# Elsewhere, control characters have been named control functions
.. in case of absolute homography ...
#082# Homography between ? and ?
#083# Homography means "equal in writing"; what is absolute equal in writing?
... in the common template
#084# No common template can be found in this document

5.2.1 ....
The general interface consists in (of?) three ...
#085# Is there a specific interface too?
#086# Several typos here
#087# It is unclear why these procedures have been numbered
#088# It is unclear how CARABIN can _produce_ a _PRE_fabricated bit string.
#089# It is unclear whether there are bit strings that are
processable indirectly.

The implementer may choose appropriate values for the application.
#090# The implementer of this API chooses values for an application
he does not know anything about?

#091# General remark: As far as understood, all three 'procedures' are
effectively functions (ie they have no side-effect).
It is unclear why they are called procedures then.

5.2.1.1 .....
table .... If this parameter is not provided, the common template
will be assumed.
#092# It is unclear how this should be implemented in a language
that requires all the parameters being present.

COMPCAR process:
This procedure shall be processed to give results equivalent to the following:
This procedure shall be processed to give results equivalent to the following:
#093# Two times the same sentence
#094# Yet another use of equivalent
#095# Process or procedure?

Submit character strings string1 and string 2 and table 'table' to
procdure CARABIN ...
#096# CARABIN does not have such parameters.
After having looked up the definitions, it is clear what has been
intended, why not describe it precisely?
Execute procedure ....
#097# Perhaps process the procedure?

A minimum interface for conformance level A or B can very well(?) use
the standard C-language function model of ...
#098# Why "minimum"?
#099# The C-language has no notion of function model

This limited interface ...
#100# Is a limited interface the same as a minimum interface?

COMPBIN process:
.... coded character strings ..
#101# Undefined notion
... octet by octet ...
#102# Unclear what octets have to do here

5.2.1.3.
....to a comparable bit strings
#103# All bit strings are comparable, aren't they?

... not to compile this table 'table' every time this function is called.
#104# It is unclear what compilation model is used here.
Is this compilation the the same as the earlier mentioned prehandling?

NL29. The API specification will disappear in the second FCD.

Disposition of comments from Sweden

As an O-member Sweden submits following comments for consideration. If not the comments are not considered. Sweden may vote NO on the FDIS.

MAJOR comments

Each of the following points must be satisfactorily resolved, and implemented in the next draft, in order to turn a possible NO vote at FDIS level into a yes vote.

All API requirements MUST be completely removed.

It must be possible to follow the International String Ordering standard, or base national string ordering standards on it, WITHOUT requiring implementations to provide a particular API. Secondly, before standardising on a new API for this there should be considerable experience and consensus on how it should work. This is not yet achieved. In either case a string ordering API must be moved to another later standard, if standardised at all.

SE1. Accepted.

All FILE FORMAT (14652 or other) requirements MUST be completely removed.

It must be possible to follow the International String Ordering standard, or base national string ordering standards on it, WITHOUT requiring implementations to provide or use a particular file format.

SE2. A specific format will be defined and used but will not be mandated for implementation.

All other dependencies on ISO/IEC 14652 must also be removed, including the normative reference. Those wishing to follow 14651 should NOT be forced to also follow 14652.

SE3. The standard will not refer to ISO/IEC 14652 any more and will be self-contained.

Combining non-spacing characters, as well as other canonical equivalences (in the Unicode 2.0 sense) in general, MUST be handled properly.

SE4. Accepted in principle.

I.e. character sub-sequences that are *canonically equivalent* according to Unicode 2.1 MUST be regarded as equivalent at *all* ”levels” (in the 14651 sense) when compared as text (i.e. character) strings according to ISO/IEC 14651.

The easiest way to do this is to do full canonical decomposition and combining marks canonical reordering according to Unicode 2.0.

An appropriate table of (full) canonical decompositions is publicly available from the Unicode consortium, and should be a normative table in an annex of 14651. This must include the algorithmic decomposition of Hangul syllables into conjoining Hangul Jamo. (See also comment 9 below on Japanese.)

SE5. Convergence with the Unicode consortium recommendations will be achieved on a mutual agreement basis in the respect of national bodies. The Dublin meeting has worked remarkably well in harmony in this sense. This spirit of collaboration is seen as an excellent contributing element which has some historical ground in JTC1/SC2.

Compatibility equivalent (in the Unicode 2.0 sense) sub-sequences must be regarded as different ONLY at the least significant ”level” (”level 4”), NOT at any more significant level. E.g. level 3 should only handle ”case” variants, no other kind of variation (like full-width/half-width) which is not based on orthographic needs, only on backward compatibility needs.

SE6. Not accepted.

An appropriate table of (full) compatibility decompositions is publicly available from the Unicode consortium, and should be a normative table in an annex of 14651.

14651 must cover all scripts and characters currently covered by Unicode 2.1 and 10646 including all amendments to date.

The current draft misses many scripts. It is our understanding that the Unicode consortium can provide fuller tables, covering all scripts currently in 10646.

SE7. Data provided by the US national body will dynamically be used in the template of the second FCD as far as possible, in the respect of other national body considerations.

Arabic: The suggestion (annex C, and annex 1 p. 94) that the Arabic presentation forms are stored in a different order than the nominal characters for Arabic must be removed (it is false).

SE8. According to ISO/IEC 10646 (and contrarily to the Unicode standard), this is left to implementation. The template of the second FCD will reflect logical order but tailoring will be allowed to fit specific, actual implementations.

Thai and Lao: The standard should, however, take into account the ”visual”, rather than ”logical”, storage order for Thai and Lao characters, and require pre-processing to order them ”logically” before key construction.

SE9. This particular problem will be noted and an explanation will be added to warn implementers that if Thai and Lao is to be ordered, proper preparation of the data should be done.

Japanese:

The half-width voiced marks must be considered combining with the preceding half-width Katakana character (to get correct collation). This affects the mapping from compatibility characters to nominal characters.

SE10. Accepted in principle.

Prolonged sound marks: These take on different weights depending on the preceding kana letter.

Repeat/iteration marks: Again, our understanding is that these should take on different weights depending on the preceding kana letter.

SE11. This will be left to tailoring and preparing of the data, according to Japanese NB advice.

Kanji digits (including "fraud-proof forms"), 0-9: These should be handled like other digits. However, handling of ideographs for ”ten”, ”hundred”, etc., need not be done by default.

SE12. Not accepted because the main usage of Kanji digits are similar to the English words "one", "two", "three" etc. and not similar to the digits '1', '2', '3' etc.

Make certain "old" Katakana sort together with "modern" spellings (va, ve, vi, vo).

SE13. Not accepted because such collation has not been used in the current Japanese daily life.

The standard should explicitly allow for ”transliterative” collation. I.e. the collation is primarily done on transliterated strings (whether manually, ”pronunciation field”, semi-automatically, or automatically derived), and only to a lesser degree depend on the direct input/output strings (which should not be completely ignored). This would be useful for ”pronunciation” collation often used for Korean and Japanese, or indeed even for ”phonebook” collation. I.e., levels 1–3 would be levels 1–3 on the transliterated string, and levels 4–6 would be levels 1–3 on the original string, level 7 would be level 4 of the original string.

SE14. Not accepted. This is seen as overspecification at this layer of application.

Sequences of (base ten) digits, up to a settable length (n) which is at least five, in the same script must by default be pre-processed to include leading zeroes so that the length of the digit sequence is n digits long.

By default, we limit this to natural numbers written as ordinary base ten numerals. A tailored version may do something more fancy.

Each digit is given a weight that in some way corresponds to its (individual) decimal value, irrespective of script. The weight is such that all digits are collated *after* all alphabetical or ideographic characters). Periods and digits are to be collated after spaces (if spaces are not ignored). We would then, without any tailoring, get the expected ordering of strings like these:

1. Intro
1.1 bbbb
2. dsfsdfsdf
2.1 fsdfds
2.2 sdfsfsd
...
2.11 dfsd
2.11.1 vxvxcv
3 vxcvxc
3.1 dfsf
...
10. cvsdf
11. sdfsda

Many things are sequence numbered (and sub-sequence numbered) just like this. It would be a bad idea if they by default came out in the wrong order, and additional effort had to be made every time such strings needed to be properly ordered.

SE15. Same comment. Not accepted. This is left to prehandling. There are too many requirements in this domain to generalize the approach at this layer of the application, from positional sorting of catalog numbers to sorting interpretations of Roman digits and all kinds of numerical representations.

14651 must make ordering compromises ONLY if there is a need to compromise among existing nationally/linguistically acceptable orderings (transliteration aside). This means that for many scripts, one would not, if this principle were followed, need to specify any national variation.

SE16. Accepted as far as possible.

The normative reference to the programming language C must be removed. There must be no requirement that one must use C as the programming language.

SE17. API specification will be completely removed. Accepted.

Many programming languages can be used and there is no reason to single out C, even it is used in an example (which should be deleted). Indeed, with the deletion of the API part, there is no need to refer to any programming language at all, even for examples.

SE18. Accepted.

The notion of ”posthandling” must be removed since no posthandling is, or need be, specified in this standard.

SE19. Accepted.

All non-empty *sequences* of space characters (there are about 10-20 different character codes), paragraph separators, line separators, C0, and C1 ”control characters” must be regarded as equivalent at all but the last ”level”, and regarded as either a single space or be ignored. Even if not in general ignored, they should be ignored for collation purposes at all but the last level if they occur in the very beginning of a string. (This can be handled by the pre-processing.)

SE20. Not accepted. There is a requirement both for ignoring certain spaces and not ignoring certain others. This will be entirely tailorable.

There is no normative mathematical description on how the comparison keys for strings are constructed from the (symbolic) weights for individual characters.

There is an informal description (annex C), but it is very incomplete, imprecise and based on some strange but weak implementation experience. The current draft is far from satisfactory in describing the string ordering and how it depends on the weight tables and other data. What is needed is a mathematically precise specification, so that we have a firm semantics for ”multi-level” string collation.

SE21. Modifications will be made to the standard but it will not be a completely mathematical specification. It has to be made clear for software engineers who are in general not mathematicians. However, as far as possible the second FCD will be more formal with some help provided by the US participating national body.

Map each (pre-processed) string to a value in R (the mathematical ”real” numbers), in such a way that the values mapped to are ordered, in R, in the same way that the strings should be ordered.

The standard should allow implementations to use any strictly monotonically increasing mapping from the specified values in R to other values in R that can be more compactly represented. The standard need not normatively specify any such mapping, but can give a few as examples (it is our understanding that the Unicode consortium can provide a few representations). The Swedish NB can help with the detailed specifications of the mapping from strings to R (using tabulated character weights, whether symbolic, as in the current annex 1, or direct), and example strictly monotonous mappings from R to R.

SE22. The standard will be oriented toward results that should be identical in all implementations of ordering strings based on the transformations implied by a same tailored ordering table. The standard will make clear that exact numbers and exact representation of keys is not mandated. SC22/WG20 experts do not feel that a pure mathematical expression of the model is required.

Parsing strings into ”lines”/”records” and ”fields” (or any other structure) must be considered out of scope for this standard. If done, it would be done prior to, not included with, collation proper.

SE23. Accepted.

The major comments above, and the minor comments below, strongly indicate the need to rewrite the entire text from scratch. We see no way in which minor changes here and there can correct the collection of errors, omissions, and misdirected requirements contained in the current text. Suggested table of contents:

1. Scope [see below]
2. Conformity [see below]
3. Normative references [only 10646]
4. Symbols and Definitions [see below]
5. Multi-level string ordering
   5.1 Included pre-processing [canonical equiv., digit seq., etc.]
   5.2 The levels
   5.3 The weights
   5.4 Symbolic weights
   5.5 Basic key construction from weights [mathematical description]
   5.6 Strictly monotonic mappings of keys [allowance, not details]
6. Tailoring of the string ordering [how it would work; what can be done]
7. Transliterative string ordering
8. Additional explanation about the collation tables [format of tables not normative except for the standard itself]

Annex A (normative): Combining non-spacing marks.
Annex B (normative): Canonical decomposition, and canonical reordering.
Annex C (normative): Compatibility decomposition. [+special for half-width voiced marks]
Annex D (normative): Thai and Lao reordering for collation.
Annex E (normative): Multi-level character weights.

Annex F (informative): Rationale for multi-level string collation.
Annex G (informative): Example collations. [NOT normative]
Annex H (informative): Example suitable monotonically increasing mappings. [NOT normative]
Annex J (informative): Example collation tailorings. [at least one complete example]

SE24. This suggestion will be taken into consideration as a positive contribution. Some elements will not be included as per the current disposition of comments. To take into consideration all NB comments some freedom must be left to the editing committee.

It is our understanding that the US NB may present other major comments, e.g. regarding such things as the principles by which the weight tables are constructed, that we too may support as major.

SE25. We take good note.

Minor comments

The comments in this section are recommendations for change, but do not one by one constitute reason for a no vote. Taken together, however, they strongly indicate the need to completely rewrite the entire text of 14651 from scratch. Since we do suggest a complete rewrite, there are no references to specific sentences in the comments below. Rather, it is a number of items for the editor to keep in mind when writing the new text.

Change the title. Suggestion: ”International string ordering — Multi-level collation method and international compromise string order”.

SE26. Not accepted. The keyword "multi-level" shows only a technical detail of this work which intends to support culturally acceptable ordering and the common template should not be considered as an international compromise..

Section 1: Please make two fairly detailed lists: one with items that are in scope for this standard, and one with items that are, or could be seen as, related, but that are out of scope.

The first list (in scope) would include items such as

- An internationally acceptable collation order for strings that include characters that are normally not used in a particular region.
- Compromises between (expected) national collation orders for strings with scripts used for many languages and/or conventions of collation.
- Certain pre-processing (*internal* to the collation) to obtain actual strings to compare. The pre-processing can reorder certain characters, and replace sub-strings by equivalent, but canonical, sub-strings.
- Proper handling of combining sequences of characters.
- Transliterative collation, where the transliteration itself is *external* to the collation.

The second list (out of scope) would include items such as:

- Application programming interface (API) for invoking string comparison or collation.
- File formats for specifying collation order to an actual implementation.
- Parsing of a string to obtain sub-strings to collate.
- Escape-sequence or control code interpretation.
- National variations from the compromise string ordering (national standards are expected to cover these).

There should be an informative annex indicating at least one complete, medium complex example national tailoring (which of course is/are *not* normative even for the nation(s) taken as examples).

3) All examples and side remarks must be made into ”Note”s, to clarify their non-normative nature. (Since we suggest a complete rewrite, we see no point in listing the ones in the current text here, but the editor should keep this in mind when writing the new text.)

4) Section 3: ”API” is out of scope and should not occur in the definition list.

5) Section 3: ”character string”; strange definition, in what sense ”logical”?

6) Section 3: ”concatenation”; strange definition, in what sense ”logical”?

7) Section 3: ”equivalence”; the text there constitutes a lot of ranting about string comparison, but contains no definition of ‘equivalence’.

8) Section 3: ”field”; this is out of scope for this standard.

9) Section 3: ”posthandling”; no such thing is specified, nor need be specified, by this standard.

10) Section 3: ”record”; this is out of scope for this standard.

11) Section 3: ”reference/referenced string”; bad terminology, please do not use it.

12) Section 3: ”procedure”; this is out of scope for this standard.

13) Section 3: ”transformation”; the text there constitutes a lot of ranting about string comparison, but contains no definition of ‘transformation’.

14) Section 5: OUT OF SCOPE, DELETE. See above.

15) Section 6: Rewrite completely, current one refers only to out-of scope material.

16) Other sections: see major comment 16.

17) B.t.w.: We can do without the cryptic symbolic weight/script names (like ”Hy” for Armenian, and ”a8” for ”a’s weight” (having a very American pun)).

SE27. The scope section will be reedited for clarity. We take note of suggestions except where it goes against the disposition of comments.

Disposition of comments from the UK

The UK votes against ISO/IEC FCD 14651 with the following technical and editorial comments. These comments were compiled by members of IST/5: Programming languages.

Contents

Part 1: Technical Comments
Part 2: Editorial Comments
Part 3: Technical and Editorial comments on tables in Annex 1

* * * * * * * *

Part 1: Technical Comments

1.A. Open-endedness of this standard

The meat of ISO/IEC FCD 14651 is in Annex 1 (normative) International Common Template Table and the accopmanying lengthy tables. The text of Annex 1 says that:

> "In this ordering table constituting a common template, a number of scripts are missing, in some cases due to lack of data at time of editing, in other cases due to the non-inclusion of those scripts in ISO/IEC 10646 at time of publishing. It is the intent of ISO/IEC to complete ordering of those scripts explicitly in the common template whenever data becomes available by way of amendments to this International standard."

This makes the project open-ended, and it will be difficult or impossible to vote via the normal CD, FCD, DIS, FDIS stages if different things are going to be added at various stages to a normative annex, and appears to be against ISO procedures.

There are no indications from ISO/IEC JTC1/SC22/WG20 as to how these additions would be progressed, either via a process of draft amendments which are subject to voting by national member bodies, as is used in ISO/IEC 10646, or via some registration process.

Clause 4 states that:

> Addenda to ISO/IEC 10646 will be published from time to time; these addenda may then also give way to addenda in this international standard if necessary.

Harmonizing the repertoire of ISO/IEC 10646 and ISO/IEC FCD 14651 should be a major priority for ISO/IEC JTC1/SC22/WG20: the statement that "these addenda [in ISO/IEC 10646] may then also give way to addenda in this international standard if necessary" seems far too weak, and admits too many unresolved possibilities or users of this standard.

Until a methodology to establish this link is developed - preferably in conjunction with the development of ISO/IEC 10646 (which could be achieved through developing parallel addenda for both ISO/IEC 10646 and ISO/IEC FCD 14651) - the UK vote should remain a NO vote.

GB1. The intent is to maintain ISO/IEC 14651 in synchronization with the ISO/IEC 10646 development, including amendments. The Ordering standard will mention to which level of the amendments to ISO/IEC 10646 the common template ordering table applies.

1.B. Conformance, and the Default multilingual sorting table

(international Common Template)

B.(a) Clause 6 - Conformance

The Conformance Clause needs complete revision to be precise (rather than stating "implies" etc. and should not use the partial table in Annex 2 since this would allow an API to be conformant if it ONLY processed those characters actually used in Annex 2, which is clearly not the complete set even used in English) and also to place Conformance requirements on the application program calling the API.

GB2. The conformance section will be revised.

B.(b) Status of the Default multilingual sorting table (international Common Template)

The status of this Annex needs to be made more clear.
ISO/IEC FCD 14651 states in Section 5.5:

"Normative Annex 1 gives the international Common Template ordering table used as a template for tailoring localized applications working on the full repertoire of ISO/IEC 10646 (the Universal multi-octed coded character set)."

Because there is nothing else available in any ISO standard, there is a possibility that "the international Common Template ordering table" will be taken as THE default ISO multilingual ordering standard.

It is vital for users of this standard to know whether ISO/IEC FCD 14651 DOES intend to provide THE default multilingual ordering standard in these tables. ISO/IEC FCD 14651 is very unclear in this respect: although the tailoring is the aim of the standard, an apparent default table forms the bulk of this draft standard, which is not its stated aim.

Section 7.1 states the following:

> Although the data in table 1 defines a specific ordering, it should not be considered as an internationally acceptable default ordering but only as a starting point for the tailoring process prescribed in the next paragraph.

On the other hand, 5.2.1.1 and 5.2.1.3 both state:

> - table: the name of the ordering table to be used to make the comparison. This is only required for conformance level C. If this parameter is not provided, the common template will be assumed. The default is the name "ISO14651_1998_TABLE1" which shall invoke the unmodified Common Template table described in annex 1 of this International Standard. Conformance level A requires the use of the latter table.

Section 5 - Requirements - states that:

> This international standard can be implemented with three conformance levels of increasing complexity. Hence the first conformance level (conformance level A) is limited to a comparison API using the fixed common template specified in annex 1 (table 1) with the result precision fixed;

Given the open-endedness described in Comment A above, and the open-endedness of the table described there, this makes it extremely difficult to conform to this Level.

GB3. The conformance section will be revised. To make sure and very clear that the template is not seen as a default, the standard will mandate that any

application conformant to ISO/IEC 14651 shall declare the delta applied to the template.

It is also important that ISO/IEC JTC1/SC22/WG20 and other groups (in ISO/TC46, ISO/TC37, ISO/IEC JTC1 and CEN/TC304, the Unicode Consortium and the Java developers community, for example) should rationalise any overlapping or competing efforts in defining any multilingual sorting standards.

The UK notes that in ISO/TC37/SC2 processing has been suspended for ISO FDIS 12199 to allow an assessment of overlapping or competing initiatives to take place, and liaisons to ensure that no conflicts in ISO standards in the area of multilingual sorting are possible.

In view of this, and particularly because of other concerns noted in A. and B. above, the UK considers that following the FCD stage of voting, similar liaison activities should be explored by ISO/IEC JTC1/SC22/WG20 with the same aim, before proceding to the next stage of draft preparation and voting.

GB4. Liaisons should be pursued. SC22/WG20 is very open to this convergence.

1.C. Symbols, nomenclature, and abbreviations

The relationship of ISO/IEC 14651, ISO/IEC 14652 and ISO/IEC 10646 remains unclear, especially in relation to the nomenclature of individual scripts and individual characters. That is there are several places where new names/abbreviations for characters and scripts are used which do not match those listed in ISO/IEC 10646. This is confusing for users.

Clauses 5.4 (Table formation) and 5.5 (Common Template table) provide some brief information on this, but the conventions used in abbreviations and naming of entities in ISO/IEC FCD 14651 is not spelt out anywhere in ISO/IEC FCD 14651.

Clause 4, which describes Symbols and abbreviations thus:

> Identification of characters of the ISO/IEC 10646 repertoire will be by means of symbols of the form <U[XXXX]XXXX>. The occurrences of XXXX which follow the letter "U" ...

should be changed to incorporate notation in ISO/IEC 10646, amendment 9 to say:

> Identification of characters which are included in ISO/IEC 10646 will be by means of symbols of the form <U+[XXXX]XXXX>. The occurrences of XXXX which follow "U+" ...

In the statement:

> Whenever possible, in the Common Template ordering table, glyphs are used in comments alongside with character ordering definitions. This gives a more accurate understanding of characters in question. It is understood that these glyphs may be removed in machine-readable files.

Glyphs do not appear in the draft received.

In the statement:

> For easy cross-referencing the various weights, numeric relative values (informative) will be shown in the table as comments. A system of short mnemonics intended to replace glyphs when it is not possible to transmit them will also be used in tables alongside with glyphs representing characters, whenever possible.

These mnemonics do appear in ISO/IEC FCD 14652, but not in ISO/IEC FCD 14651. This inconsistency should be dealt with.

GB5. No more dependency on ISO/IEC 14652 will remain in the second FCD. Symbols, for most of them, will be autogenerated from a table provided by the Unicode consortium, possibly with some minor adjustments. A note should be added to indicate that if one wants to change these symbols, tailoring can be used to make them more mnemonic in a given natural language.

1.D. Sample bindings

It would be appropriate to give sample bindings to more than one programming language, to avoid giving the impression of preferring that language. (This point has been raised about other language-independent standards in the past).

GB6. API specification will disappear.

1.E. Informative Annexes

Annex E is a somewhat discursive essay and has no place in the standard. The same could be said of parts of other informative annexes.

GB7. Accepted.

1.F. Definitions

In Section 3 - Definitions - there are problems, which in part relate to the scope of the standard.

(a) The definition of "field" makes reference to record, whereas the API is purely concerned with strings. The only obvious reference to record is in 5.1.2. However, this reference is about the way an application program should handle a whole set of input records and is therefore outside the scope of the CD. The usage of the API MIGHT be about sorting a series of records, but on the other hand it might be used solely to store an individual object in a database, in an indexed fashion, and then comparing a new object with the existing ones to determine its correct position.

Thus the use of the concept of a record is bound to a specific usage of the proposed standard and not the generality. It should therefore be removed from any normative part and perhaps included in an example or exposition of the usage of the standard.

(b) "procedure" should use the general definition from ISO/IEC 13886 Language Independent Procedure calling, which is: "A closed sequence of instructions that is entered from, and returns control to, an external source"

(c) A definition of "glyph" is required (Clause 4 para 3) if this term is to be used. Alternatively, the use of the term "graphic symbol" (as in ISO/IEC 10646, section 4.19) may be preferable.

(d) In addition, the word "precision" is used in a special sense. in several places, particularly in clauses of section 3 and section 5, and in particular in section 5.2.1.1.

It should be defined in the glossary, even if the full definition is given later in 5.2.1.1 - this definition could be used in Section 3.

NB: Is there an alternative term to "precision" in English that might be more useful here?

(e) In addition, some of the definitions make no sense, e.g. "concatenation", "record" or the first sentence of the definition of "equivalence", or idiomatic (and therefore inappropriate in a standard), e.g. the definition in "string."

* * * * * * * *

GB8. Definitions will be revised. Terms which are not considered necessary to be defined, such as "field", "procedure" and "precision" will be removed.

Part 2: Editorial Comments

2.1. Numbering of Annexes: having normative annexes labelled 1 & 2 and informative annexes labelled A, B, C, D, E is rather confusing, and conflicts with ISO/IEC JTC1 Directives.

GB9. Accepted.

It is clear that this has been done in order to separate Normative and Informative Annexes. However, this could be done by referring in the text to "normative annex n" or "informative annex x".

2.2 References to the word "draft" (e.g. on page 23 of ISO/IEC FCD 14651) should be removed from the whole text of ISO/IEC FCD 14651.

GB10. Accepted.

2.3 Editorial style: English text is rather lengthy, and would benefit from review by a native English-language speaker. It would also benefit from a spellcheck in places.

GB11. English text will be revised by coeditor Michael Everson.

2.4. In Clause 4 line 1 "will be" is future tense and should be the present tense, e.g. "within this International Standard is". (There are also many other usages of the future tense which should be changed.)

Clause 4 refers only to the usage of Symbols within Annex 1 and should probably be moved there.

GB12. This will be revised.

2.5 It should also be noted that in some places the standard refers to Conformance Levels A, B and C, and in others to Conformance Levels 1, 2 and 3. This confusion should be removed.

GB13. Conformance clause will be revised.

2.6. The purpose of Clause 5.1 is unclear, especially from the title. It needs to be clearly defined in two parts - the existing 5.1.1 as API installation/implementation requirements and 5.1.2 as Application Program recommendations. There may be some normative items within the existing 5.1.2, but it is not clear what they are.

GB14. Most of this text will be moved to an informative annex.

2.7. Clause 5.1.1 line 1: conformance level 1 is not defined.

2.8. Clause 5.1.1 para 3 seems to specify a physical implementation technique and it is not clear that this is appropriate. This may not be suitable for non-C/POSIX implementations of the API. The CD should be at the logical, not physical level, if it is to be cross-language.

2.9. Clause 5.1.2 para 1 refers to "implementor". This should be "caller program" or some such term.

2.10. Clause 5.1.2 para 1 seems to be a note or recommendation and should be labelled as such (apart from the last sentence, which could be taken as a requirement for the caller program).

2.11. Clause 5.1.2 para 1 last sentence : It is not clear who "user" is.

2.12. Clause 5.1.2 para 2 sentence 1 seems to be predicated on the assumption that the calling program actually does input. The "input" process may be totally detached from the API usage. Replace "used on input in" by "to"

2.13. Clause 5.2.1.1 and 5.2.1.2 "precision" para 3: This parameter should be flexible, so that if specified at a non-zero value when an API only conforming to level A and B is provided that it will be interpreted as 0, i.e. the application program can use the highest level available without having to change if the API implementation changes. [This leads to a separate issue - the application probably needs another API call to determine the conformance level that the API provides.]

2.14. Clause 5.2.1.1 and 5.2.1.3 "table" parameter: The naming of this may cause problems with revision of this Standard or the API, since it ties a usage to a specific version. The objective should be to allow an application program to run unchanged with an enhanced version of the standard/software without change (i.e. using any new table provided, unless it requires the usage of a specific version.)

2.15. Clause 5.2.1.1 "process": Delete "be processed to."
Also in Clause 5.2.1.1 "process": top line is duplicated.

2.16. Clause 5.2.1.1 and 5.2.1.2 "C binding Example": This needs to show how a program written to use the API at conformance level C can automatically fall back to level A/B without any change being made.

GB15. All API specification will be removed in the second FCD.

2.17. Clause 5.5 para 2 should be a Note.

GB16. This will be moved into the introduction to the table.

2.18. Clause 5.2.1 para 1 belongs in Clause 6, and Clause 5.4 para 2 belongs in Clause 6

GB17. This text will disappear.

Part 3: Technical and Editorial comments on tables in Annex 1

3.A. [Section beginning approximately line 9 of the tables, quoted below]
Note: the ISO/IEC JTC1/SC22/WG20 website has a text file for Annex 1: line numberings are taken from that.

% Script symbols/Symboles des syst=E8mes d'=E9criture

script <Xx> % Special/Sp=E9cial
script <Xy> % Numbers/Num=E9rotation
script <La> % Latin/Latin
script <El> % Greek/Grec
script <Cy> % Cyrillic/Cyrillique...

3.A.1. Names of scripts

Names of scripts should match those used in Annex A of ISO/IEC 10646. This section, or appropriate text in Annex 1 of ISO/IEC FCD 14651, should also identify the UCS Collection numbers and/or row numbers involved.

There seems to be no requirement to include names in both French and English: one name appears to be sufficient, especially when names are similar, or in many cases identical. In script <Hn> % H=E0n (CJK)/H=E0n (CJ=
K)
there would in any case be no accented letter a in Han.

3.A.2. Order of scripts

In line with expectations of Cultural Adaptability, as in the beginning of the listing Latin, Greek, Cyrillic, Georgian, Armenian, the order would be improved by maintaining a consistent East through West order, to ensure that where possible scripts of adjacent countries are kept together. In general this is followed, with the exception for syllabaries (Ethiopic, Canadian Syllabics, Cherokee, Hangul) which have nothing in common with each other, and were all developed independently of each other.

This is contrary to the stated aim explained in section 5.5 that

"scripts have been ordered in the same kind of "logical" order in which similar or related scripts are kept near to one another in groups... This order is less arbitrary and therefore considered more user-friendly."

An improved rationale for ordering which does meet these expectations would be:

[Specials/Numbers]

script <Xx> % Special/Sp=E9cial
script <Xy> % Numbers/Num=E9rotation

[European scripts, related to Greek]

script <La> % Latin/Latin
script <El> % Greek/Grec
script <Cy> % Cyrillic/Cyrillique
script <Ka> % Georgian/G=E9orgien
script <Hy> % Armenian/Arm=E9nien

[Semitic scripts, closely derived from Phoenician]

script <He> % Hebrew/H=E9breu >>>>
script <Et> % Ethiopic/=C9thiopique >>>>
script <Ar> % Arabint (Arabic intrinsic/Arabe intrins=E8que)
script <Ax> % Arabfor (Arabic forms/Formes de pr=E9sentation arabes)

[Indic scripts, derived from Brahmi script]

script <Dn> % Devanagari/Devan=E2gari
script <Bn> % Bengali/Bengali
script <Pa> % Gurmukhi/Pendjabi
script <Gu> % Gujarati/Goudjarati
script <Or> % Oriya/Oriya
script <Ta> % Tamil/Tamoul
script <Te> % Telugu/T=E9lougou
script <Kn> % Kannada/Kannara
script <Ml> % Malayalam/Malayalam
script <Th> % Thai/Tha=EF
script <Lo> % Lao/Laotien
script <Bo> % Tibetan/Tib=E9tain

[East Asian scripts]

script <Yy> % Yi/Yi
script <Hn> % H=E0n (CJK)/H=E0n (CJK) >>>>
script <Hg> % Hangul/Hangul
script <Xk> % Kana/Kana

[North American scripts]

script <Jl> % Cherokee/Cherokee
script <Sl> % Canadian Syllabics/Syllabaire canadien

GB18. All script identification and order will now be entirely left to tailoring with simplification of the syntax and by the same occasion of the table. Most of the table will be autogenerated, the order of characters of different scripts following mostly the UCS order, unless eventually otherwise decided by SC22/WG20 experts.

3.A.3. Maintaining parallel development with ISO/IEC 10646

It should be made clear that ISO/IEC FCD 14651 includes amendments which are still in draft stages of voting within ISO.

Collections like Tibetan, Ethiopic, Cherokee and Canadian Aboriginal Syllabics are OUTSIDE the repertoire of ISO/IEC 10646-1: 1993.

There are other amendments in progress in ISO/IEC FCD 14651 which are not included below. These are:

Thaana script (Divehi language, Maldives Republic), Ogham, Runic, Burmese, Khmer, Glagolitic scripts.

Braille Symbols (already agreed to form part of ISO/IEC 10646) are a

special case: should these be sorted as symbols, or as their
alphabetic equivalent? This will need to be sorted out in the future
in ISO/IEC FCD 14651.

There may be some further amendments as a result of the Seattle
meeting in May 1998 of ISO/IEC JTC1/SC2/WG2 and ISO/IEC JTC1/SC2
which should also be taken care of.

ISO/IEC JTC1/SC22/WG20 should declare how it will keep ISO/IEC 10646
and ISO/IEC FCD 14651 in step, otherwise there are permanent
difficulties in the various voting stages of ISO/IEC FCD 14651.

GB19. See disposition of comment GB1.

3.A.4. Script codes

The two letter Script code enclosed in angle brackets is the subject
of a new work item in ISO/TC46/SC2/WG8 - ISO NP 15924: Codes for
representation of names of scripts. This has not yet even reached the
CD stage in ISO/TC46/SC2/WG8 and it is quite possible that these
codes might change.

These codes should be removed. This feature is undocumented in any
part of ISO/IEC FCD 14651.

GB20. See disposition of comment GB18.

3.B. [Section beginning approximately line 742 of the tables, quoted below]

% Case and size/Casse et taille

<BLK>

<%MIN>

% The previous statement is deliberately wrong...
% ...If small letters are to be ordered before capital letters,
% then the % in the previous statement must be removed. If capitals are
% to be ordered before small letters, then the previous statement must be
% removed.

There is no description of what happens if the statement is left in
unchanged, only what happens if the statement is changed or removed.

NB: Is there any way to indicate what should be done other than to
say "the previous statement is deliberately wrong...?"

3.C. [Section beginning approximately line 780 of the tables, quoted below]

% Xz (Accents/Accents)

There is no description of how the 5-character codes are derived, or
a description which relates these 5-character codes to ISO/IEC 10646
character names.

3.D. [Sections beginning approximately line 1137 and 4490, quoted below]

% Cy (Cyrillic/Cyrillique)

<acyril8>
<acyrilbreve8>
<acyrildieresis8>...

order_start <Cy>;forward;forward;forward;forward,position
%
<U0410> <acyril8>;<BLANK>;<CAP>;IGNORE % CYRILLIC CAPITAL LETTER A
<U0430> <acyril8>;<BLANK>;<MIN>;IGNORE % CYRILLIC SMALL LETTER A
<acyril8>...

Although a massive task has been achieved in providing a logical order for symbols, digits, Latin, Greek, Cyrillic, Georgian, Armenian, Arabic and Hebrew, the order should be improved in two regards:

GB21. Symbols will be mostly numeric after autogeneration of the table. Changing them will be left to tailoring.

3.D.(a) Hebrew should precede Arabic on a West through East basis;

3.D.(b) After Hebrew and Arabic, no further scripts are specified in detailed ordering. Further details are needed for these.

3.D.(c) There also seems to be a false distinction between two types of Arabic script, when in fact there is only one type. Distinctions between characters in UCS collections 14, 15, 64 and 68 (defined in ISO/IEC 10646, Annex A) should be dealt with so that at the first pass Arabic characters in UCS collections 64 and 68 should be equated to their partners in UCS collections 14 and 15.

GB22. See disposition of comment GB18.

3.D.(d) Inconsistencies in Greek ordering

In Annex A there are inconsistencies between the full listing of accented Greek characters, and the listing of Greek characters in earlier parts of Annex A.

The full listing of accented Greek characters - indicated at the end of this comment (d) - seems to be the most consistent and should be adopted in the other parts listed below, with notes on errors indented.

Excerpts from Annex A, observing the same order given there, with notes.
~~~~~~

```
collating-symbol <TONOS> % tonos
collating-symbol <PSILI> % psili
...

        tonos normally near the end of the sequence, not at the
        beginning

collating-symbol <PSILI+VARIA>
collating-symbol <VARIA+YPOGE>
collating-symbol <VARIA+DIALY>
collating-symbol <OXIAA> % oxia
collating-symbol <OXIAA+YPOGE>
collating-symbol <OXIAA+DIALY>
collating-symbol <PERIS> % perispomeni
collating-symbol <PERIS+YPOGE>
collating-symbol <YPOGE> % ypogegrammeni
collating-symbol <PROSG> % prosgegrammeni
collating-symbol <DIALY> % dialytika
collating-symbol <DIALY+PERIS>
collating-symbol <DIALY+TONOS>
...
```

Dialytika is normally the initial grouping, not secondary grouping. Some of the names above also reflect this inconsistency.
~~~~~~

```
<TONOS>
<PSILI>
...
```

TONOS is normally near the end of the seqeunce, not at the beginning

```
<PSILI+VARIA>
<VARIA>
<VARIA+YPOGE>
<VARIA+DIALY>
<OXIAA>
<OXIAA+YPOGE>
<OXIAA+DIALY>
<PERIS>
<PERIS+YPOGE>
<TONOS>
<YPOGE>
<PROSG>
<DIALY>
<DIALY+TONOS>
<DIALY+PERIS>
...
```

DIALYTIKA normally the initial grouping, not secondary grouping

```
% Accents
```

Several of these characters are dual-purpose Latin/Greek combining accents. Could this lead to any complications in mixed Latin/Greek text?

```
<U00B4> IGNORE;IGNORE;IGNORE;<U00B4> % ACUTE ACCENT
<U02CA> IGNORE;IGNORE;IGNORE;<U02CA> % MODIFIER LETTER ACUTE ACCENT
<U0301> IGNORE;IGNORE;IGNORE;<U0301> % COMBINING ACUTE ACCENT (Oxia)
<U0060> IGNORE;IGNORE;IGNORE;<U0060> % GRAVE ACCENT
<U02CB> IGNORE;IGNORE;IGNORE;<U02CB> % MODIFIER LETTER GRAVE ACCENT
<U0300> IGNORE;IGNORE;IGNORE;<U0300> % COMBINING GRAVE ACCENT (Varia)
<U02D8> IGNORE;IGNORE;IGNORE;<U02D8> % BREVE
<U0306> IGNORE;IGNORE;IGNORE;<U0306> % COMBINING BREVE (Vrachy)
<U00A8> IGNORE;IGNORE;IGNORE;<U00A8> % DIAERESIS
<U0308> IGNORE;IGNORE;IGNORE;<U0308> % COMBINING DIAERESIS (Dialytika)
<U00B7> IGNORE;IGNORE;IGNORE;<U00B7> % MIDDLE DOT
<U1FBF> IGNORE;IGNORE;IGNORE;<U1FBF> % GREEK PSILI
<U0313> IGNORE;IGNORE;IGNORE;<U0313> % COMBINING COMMA ABOVE (Psili)
<U0486> IGNORE;IGNORE;IGNORE;<U0486> % CYRILLIC COMBINING PSILI PNEUMATA
<U1FCD> IGNORE;IGNORE;IGNORE;<U1FCD> % GREEK PSILI AND VARIA
<U1FCE> IGNORE;IGNORE;IGNORE;<U1FCE> % GREEK PSILI AND OXIA
<U1FCF> IGNORE;IGNORE;IGNORE;<U1FCF> % GREEK PSILI AND PERISPOMENI
<U1FFE> IGNORE;IGNORE;IGNORE;<U1FFE> % GREEK DASIA
<U0314> IGNORE;IGNORE;IGNORE;<U0314> % COMBINING REVERSED COMMA ABOVE (Dasia)
<U0485> IGNORE;IGNORE;IGNORE;<U0485> % CYRILLIC COMBINING DASIA PNEUMATA
<U1FDD> IGNORE;IGNORE;IGNORE;<U1FDD> % GREEK DASIA AND VARIA
<U1FDE> IGNORE;IGNORE;IGNORE;<U1FDE> % GREEK DASIA AND OXIA
<U1FDF> IGNORE;IGNORE;IGNORE;<U1FDF> % GREEK DASIA AND PERISPOMENI
<U1FEF> IGNORE;IGNORE;IGNORE;<U1FEF> % GREEK VARIA
<U1FFD> IGNORE;IGNORE;IGNORE;<U1FFD> % GREEK OXIA
<U0384> IGNORE;IGNORE;IGNORE;<U0384> % GREEK TONOS
...
```

In most accented characters below, TONOS follows PERISPOMENI. Here, just above, it appears out of order.

```
<U030D> IGNORE;IGNORE;IGNORE;<U030D> % COMBINING VERTICAL LINE ABOVE (Tonos)
<U1FC0> IGNORE;IGNORE;IGNORE;<U1FC0> % GREEK PERISPOMENI
<U1FBE> IGNORE;IGNORE;IGNORE;<U1FBE> % GREEK PROSGEGRAMMENI
<U037A> IGNORE;IGNORE;IGNORE;<U037A> % GREEK YPOGEGRAMMENI
<U1FED> IGNORE;IGNORE;IGNORE;<U1FED> % GREEK DIALYTIKA AND VARIA
<U1FEE> IGNORE;IGNORE;IGNORE;<U1FEE> % GREEK DIALYTIKA AND OXIA
<U0385> IGNORE;IGNORE;IGNORE;<U0385> % GREEK DIALYTIKA TONOS
```

In most accented characters below, DIALYTIKA AND TONOS follows DIALYTIKA AND PERISPOMENI. Here, just above, it appears out of order.

```
<U1FC1> IGNORE;IGNORE;IGNORE;<U1FC1> % GREEK DIALYTIKA AND PERISPOMENI
```

```
<U02D0> IGNORE;IGNORE;IGNORE;<U02D0> % MODIFIER LETTER TRIANGULAR COLON
%
order_start <El>;forward;forward;forward;forward,position
%

        This section below, and accented letters listed under other
        vowels, appears to show the correct ordering.

<U0391> <alpha8>;<BLANK>;<CAP>;IGNORE % GREEK CAPITAL LETTER ALPHA
<U03B1> <alpha8>;<BLANK>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA
<U1FB8> <alpha8>;<VRACH>;<CAP>;IGNORE % GREEK CAPITAL LETTER ALPHA WITH VRACHY
<U1FB0> <alpha8>;<VRACH>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WITH VRACHY
<U1FB9> <alpha8>;<MACRO>;<CAP>;IGNORE % GREEK CAPITAL LETTER ALPHA WITH MACRON
<U1FB1> <alpha8>;<MACRO>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WITH MACRON
<U1F08> <alpha8>;<PSILI>;<CAP>;IGNORE % GREEK CAPITAL LETTER ALPHA WITH PSILI
<U1F00> <alpha8>;<PSILI>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WITH PSILI
<U1F0A> <alpha8>;<PSILI+VARIA>;<CAP>;IGNORE % GREEK CAPITAL LETTER ALPHA WITH PS
ILI AND VARIA
<U1F02> <alpha8>;<PSILI+VARIA>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WITH PSIL
I AND VARIA
<U1F8A> <alpha8>;<PSILI+VARIA+PROSG>;<CAP>;IGNORE % GREEK CAPITAL LETTER ALPHA W
ITH PSILI AND VARIA AND PROSGEGRAMMENI
<U1F82> <alpha8>;<PSILI+VARIA+YPOGE>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WIT
H PSILI AND VARIA AND YPOGEGRAMMENI
<U1F0C> <alpha8>;<PSILI+OXIAA>;<CAP>;IGNORE % GREEK CAPITAL LETTER ALPHA WITH PS
ILI AND OXIA
<U1F04> <alpha8>;<PSILI+OXIAA>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WITH PSIL
I AND OXIA
<U1F8C> <alpha8>;<PSILI+OXIAA+PROSG>;<CAP>;IGNORE % GREEK CAPITAL LETTER ALPHA W
ITH PSILI AND OXIA AND PROSGEGRAMMENI
<U1F84> <alpha8>;<PSILI+OXIAA+YPOGE>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WIT
H PSILI AND OXIA AND YPOGEGRAMMENI
<U1F0E> <alpha8>;<PSILI+PERIS>;<CAP>;IGNORE % GREEK CAPITAL LETTER ALPHA WITH PS
ILI AND PERISPOMENI
<U1F06> <alpha8>;<PSILI+PERIS>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WITH PSIL
I AND PERISPOMENI
<U1F8E> <alpha8>;<PSILI+PERIS+PROSG>;<CAP>;IGNORE % GREEK CAPITAL LETTER ALPHA W
ITH PSILI AND PERISPOMENI AND PROSGEGRAMMENI
<U1F86> <alpha8>;<PSILI+PERIS+YPOGE>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WIT
H PSILI AND PERISPOMENI AND YPOGEGRAMMENI
<U1F88> <alpha8>;<PSILI+PROSG>;<CAP>;IGNORE % GREEK CAPITAL LETTER ALPHA WITH PS
ILI AND PROSGEGRAMMENI
<U1F80> <alpha8>;<PSILI+YPOGE>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WITH PSIL
I AND YPOGEGRAMMENI
<U1F09> <alpha8>;<DASIA>;<CAP>;IGNORE % GREEK CAPITAL LETTER ALPHA WITH DASIA
<U1F01> <alpha8>;<DASIA>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WITH DASIA
<U1F0B> <alpha8>;<DASIA+VARIA>;<CAP>;IGNORE % GREEK CAPITAL LETTER ALPHA WITH DA
SIA AND VARIA
<U1F03> <alpha8>;<DASIA+VARIA>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WITH DASI
A AND VARIA
<U1F8B> <alpha8>;<DASIA+VARIA+PROSG>;<CAP>;IGNORE % GREEK CAPITAL LETTER ALPHA W
ITH DASIA AND VARIA AND PROSGEGRAMMENI
<U1F83> <alpha8>;<DASIA+VARIA+YPOGE>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WIT
H DASIA AND VARIA AND YPOGEGRAMMENI
<U1F0D> <alpha8>;<DASIA+OXIAA>;<CAP>;IGNORE % GREEK CAPITAL LETTER ALPHA WITH DA
SIA AND OXIA
<U1F05> <alpha8>;<DASIA+OXIAA>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WITH DASI
A AND OXIA
<U1F8D> <alpha8>;<DASIA+OXIAA+PROSG>;<CAP>;IGNORE % GREEK CAPITAL LETTER ALPHA W
ITH DASIA AND OXIA AND PROSGEGRAMMENI
<U1F85> <alpha8>;<DASIA+OXIAA+YPOGE>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WIT
H DASIA AND OXIA AND YPOGEGRAMMENI
<U1F0F> <alpha8>;<DASIA+PERIS>;<CAP>;IGNORE % GREEK CAPITAL LETTER ALPHA WITH DA
SIA AND PERISPOMENI
<U1F07> <alpha8>;<DASIA+PERIS>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WITH DASI
A AND PERISPOMENI
<U1F8F> <alpha8>;<DASIA+PROSG>;<CAP>;IGNORE % GREEK CAPITAL LETTER ALPHA WITH DA
SIA AND PERISPOMENI AND PROSGEGRAMMENI
<U1F87> <alpha8>;<DASIA+YPOGE>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WITH DASI
A AND PERISPOMENI AND YPOGEGRAMMENI
<U1F89> <alpha8>;<DASIA+PROSG>;<CAP>;IGNORE % GREEK CAPITAL LETTER ALPHA WITH DA
SIA AND PROSGEGRAMMENI
<U1F81> <alpha8>;<DASIA+YPOGE>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WITH DASI
A AND YPOGEGRAMMENI
<U1FBA> <alpha8>;<VARIA>;<CAP>;IGNORE % GREEK CAPITAL LETTER ALPHA WITH VARIA
<U1F70> <alpha8>;<VARIA>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WITH VARIA
<U1FB2> <alpha8>;<VARIA+YPOGE>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WITH VARI
```

```
A AND YPOGEGRAMMENI
<U1FBB> <alpha8>;<OXIAA>;<CAP>;IGNORE % GREEK CAPITAL LETTER ALPHA WITH OXIA
<U1F71> <alpha8>;<OXIAA>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WITH OXIA
<U1FB4> <alpha8>;<OXIAA+YPOGE>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WITH OXIA
 AND YPOGEGRAMMENI
<U1FB6> <alpha8>;<PERIS>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WITH PERISPOMEN
I
<U1FB7> <alpha8>;<PERIS+YPOGE>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WITH PERI
SPOMENI AND YPOGRAMMENI
<U0386> <alpha8>;<TONOS>;<CAP>;IGNORE % GREEK CAPITAL LETTER ALPHA WITH TONOS
<U03AC> <alpha8>;<TONOS>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WITH TONOS
<U1FBC> <alpha8>;<PROSG>;<CAP>;IGNORE % GREEK CAPITAL LETTER ALPHA WITH PROSGEGR
AMMENI
<U1FB3> <alpha8>;<YPOGE>;<MIN>;IGNORE % GREEK SMALL LETTER ALPHA WITH YPOGEGRAMM
ENI
...
```

GB23. Individual Accents will have a separated and consistent weight decided once and forever for the whole template. Tailoring will be possible for individual requirements.

3.D.(e) The order of Cyrillic characters does NOT quite reflect user expectations from the point of view of Cultural Adaptability. The Cyrillic part of the tables were largely provided by NSAI (Ireland) on the basis of expert opinion: as a result of email discussions on the ISO/IEC JTC1/SC22/WG20 email list, the Irish expert has produced revised tables on his own website, and a UK expert has also reviewed this list in parallel.

The ordering for Cyrillic should instead resemble the following (note that the string CYRILLIC SMALL LETTER should replace Cy_ in the table below, and that CYRILLIC CAPITAL LETTER equivalents should also be added, in order to arrive at ISO/IEC 10646 names.

[In passing, via IST/2, a request should also be made to ISO/IEC JTC1/SC2/WG2 to add missing Cyrillic characters as listed below as ????]

```
 U+0430> Cy_A
 U+04D1> Cy_A_BREVE
 U+04D3> Cy_A_DIAERESIS
 U+04D5> CYRILLIC SMALL LIGATURE A IE
 U+0431> Cy_BE
 U+0432> Cy_VE
 U+0433> Cy_GHE
 U+0453> Cy_GJE
 U+0493> Cy_GHE_STROKE
 U+0491> Cy_GHE_UPTURN
 U+0495> Cy_GHE_MIDDLE HOOK
 U+0434> Cy_DE
 U+0452> Cy_DJE
 U+0435> Cy_IE
 U+04D7> Cy_IE_BREVE
 U+0451> Cy_IO
 U+0454> Cy_UKRAINIAN IE
 U+04D9> Cy_SCHWA
 U+04DB> Cy_SCHWA_DIAERESIS
 U+04BD> Cy_ABKHASIAN CHE
 U+04BF> Cy_ABKHASIAN CHE_DESCENDER
 U+0436> Cy_ZHE
 U+04C2> Cy_ZHE_BREVE
 U+04DD> Cy_ZHE_DIAERESIS
 U+0497> Cy_ZHE_DESCENDER
 U+0455> Cy_DZE
 U+0437> Cy_ZE
 U+04DF> Cy_ZE_DIAERESIS
 U+0499> Cy_ZE_DESCENDER
 U+04E1> Cy_ABKHASIAN DZE
 U+0438> Cy_I
 U+0439> Cy_SHORT I
 U+04E5> Cy_I_DIAERESIS
 U+04E3> Cy_I_MACRON
 U+0456> Cy_BYELORUSSIAN-UKRAINIAN I
 U+0457> Cy_YI
 U+0458> Cy_JE
 U+043A> Cy_KA
 U+045C> Cy_KJE
 U+049F> Cy_KA_STROKE
 U+049D> Cy_KA_VERTICAL STROKE
 U+049B> Cy_KA_DESCENDER
 U+04A1> Cy_BASHKIR KA
 U+04C4> Cy_KA_HOOK
 U+043B> Cy_EL
 U+0459> Cy_LJE
 U+043C> Cy_EM
 U+043D> Cy_EN
 U+045A> Cy_NJE
 U+04A3> Cy_EN_DESCENDER
 U+04A5> CYRILLIC SMALL LIGATURE EN GHE
 U+04C8> Cy_EN_HOOK
 U+043E> Cy_O
 U+04E7> Cy_O_DIAERESIS
 U+04E9> Cy_BARRED O
 U+04EB> Cy_BARRED O_DIAERESIS
 U+04A9> Cy_ABKHASIAN HA
 U+043F> Cy_PE
 U+04A7> Cy_PE_MIDDLE HOOK
 U+0481> Cy_KOPPA                              >>>>
>U+????> Cy_Q [Cyrillic q: used in Kurdish     >>>>     ????
 U+0440> Cy_ER
 U+0441> Cy_ES
 U+04AB> Cy_ES_DESCENDER
 U+0442> Cy_TE
 U+04AD> Cy_TE_DESCENDER
 U+045B> Cy_TSHE
 U+0443> Cy_U
 U+045E> Cy_SHORT U
 U+04F1> Cy_U_DIAERESIS
 U+04F3> Cy_U_DOUBLE ACUTE
 U+04EF> Cy_U_MACRON
 U+04AF> Cy_STRAIGHT U
 U+04B1> Cy_STRAIGHT U_STROKE
 U+0479> Cy_UK
>U+????> Cy_W [Cyrillic w: used in Kurdish     >>>>     ????
 U+0444> Cy_EF
 U+0445> Cy_HA
 U+04B3> Cy_HA_DESCENDER
 U+04BB> Cy_SHHA
 U+0461> Cy_OMEGA
 U+047F> CYRILLIC SMALL LETTER OT
 U+047D> Cy_OMEGA_TITLO
 U+047B> Cy_ROUND OMEGA
>U+????> Cy_SHTE: used in Old Church Slavonic  >>>>     ????
 U+0446> Cy_TSE
 U+04B5> CYRILLIC SMALL LIGATURE TE TSE
 U+0447> Cy_CHE
 U+04F5> Cy_CHE_DIAERESIS
 U+04B9> Cy_CHE_VERTICAL STROKE
 U+04B7> Cy_CHE_DESCENDER
 U+04CC> Cy_KHAKASSIAN CHE
 U+045F> Cy_DZHE
 U+0448> Cy_SHA
 U+0449> Cy_SHCHA
 U+044A> Cy_HARD SIGN
 U+044B> Cy_YERU
 U+04F9> Cy_YERU_DIAERESIS
 U+044C> Cy_SOFT SIGN
 U+0463> Cy_YAT
 U+044D> Cy_E
 U+044E> Cy_YU
 U+044F> Cy_YA
 U+0465> Cy_IOTIFIED E
 U+0467> Cy_LITTLE YUS
 U+046B> Cy_BIG YUS
 U+0469> Cy_IOTIFIED LITTLE YUS                >>>>
 U+046D> Cy_IOTIFIED BIG YUS
 U+046F> Cy_KSI
 U+0471> Cy_PSI
 U+0473> Cy_FITA
 U+0475> Cy_IZHITSA
 U+0477> Cy_IZHITSA_DOUBLE GRAVE ACCENT
 U+04C0> Cy_PALOCHKA
```

>>>>   indicates specific optimum placements based on tables
       in various sources on Cyrillic script.

U+???? indicates characters that need to be added to ISO/IEC 10646
       as they are missing and necessary for some languages
       (in particular for Old Church Slavonic and Kurdish)

GB24. Problems are noted. Coeditor Michael Everson will give some input to help solve those.

3.E. [Section beginning approximately line 5390, quoted below]

%Tous les caract=E8res han de l'=E9dition de 1993 de l'ISO/CEI 10646 sont d=
=E9j=E0 ordonn=E9s;
%All characters of the 1993 edition of ISO/IEC 10646 are already ordered
<U4E00>...X...<U9FA5>   <U4E00>...X...<U9FA5>;IGNORE;IGNORE;IGNORE

There is a major editorial error here: the second comment line should
be alligned with the first to read:

```
%All han characters of the 1993 edition of ISO/IEC 10646 are already ordered
    ^^^
```

It should also be noted somewhere in the text of ISO/IEC FCD 14651
that this is a default ordering for Han (CJK) characters, and that
most applications would generally apply additional sorting programs
to this range of characters.

Discrepancies in spelling of Han above and H=E0n (CJK)/H=E0n (CJK) in
approximately line 39 above should also be resolved.

END OF COMMENT

GB25. Accepted.

Disposition of comments from the US

The US National Body votes to **Disapprove** SO/IEC FCD 14651, Information technology - International String Ordering and Comparison - Method for Comparing Character Strings and Description of the Common Template Tailorable Ordering.

Comments:

The Conformance clause of 14651 "must not mandate:
- more than 3 customizable levels

US1. The 4th level will remain tailorable. However an implementation using any deterministic, fixed order for the 4th level will be considered as choosing a specific, conformant tailoring.

- the use of the precise data or tailoring format specified

US2. A specific format will document the table in the standard but any equivalent way of doing will be considered conformant.

- the use of the precise API specified"

US3. Accepted.

The International Common Template Table must incorporate all of the corrections for syntax errors and other problems reported by Ken Whistler to the Editor. These corrections are required in order for the Table to even be machine processed at all.

US4. Accepted.

Furthermore, the U.S. strongly prefers that the content of the International Common Template Table be inclusive of the *entire* content of ISO/IEC 10646-1 through Amendment 7 (Collection 301), rather than an arbitrary and otherwise unidentified subset of 10646.

US5. Accepted. The version of the UCS used will be identified.

The best way to accomplish this is to replace the content of Annex 1 with the symbolic information in the symdump2.txt table provided by Ken Whistler to the Editor. This would have the additional advantage of aligning the content of the International Common Template Table with the default table in use by vendors which implement the Unicode Collation Algorithm.

US6. Accepted in principle, subject to slight revision in line with national body consensus.

The International Common Template Table must be published in machine-readable format to be usable for implementation.

US7. Accepted.

An expert contribution to WG20 from the US is being prepared
for the WG20 meeting in June 1998.

US8. SC22/WG20 members welcomes the significant and appreciated US contribution, which constitutes a positive step forward in achieving international consensus in the development of this standard.

**************************** END OF THIS DISPOSITION OF COMMENTS ************************