SC22/WG20 N666

Input document for the disposition of comments for the FCD2 14651 ballot

1999-04-22

This document includes SC22/WG20 N663 (SC22 N2911) and formatted versions of comments from National Bodies (Ireland, Germany, Sweden (SC22 N2912), USA). The applied numbering will help in referencing comments during the disposition of comments.

ISO/IEC JTC 1/SC22 - Programming languages, their environments and system software interfaces --Secretariat: U.S.A. (ANSI)

TITLE:
Summary of Voting on Second FCD Ballot for FCD 14651: Information technology - International String Ordering and Comparison - Method for Comparing Character Strings and Description of a Common Tailorable Ordering Template

DATE ASSIGNED: 1999-04-16

SOURCE: Secretariat, ISO/IEC JTC 1/SC22

BACKWARD POINTER: N/A

DOCUMENT TYPE: Summary of Voting

PROJECT NUMBER: JTC 1.22.30.02.02

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

ACTION IDENTIFIER: FYI

DUE DATE: N/A

DISTRIBUTION: Text

CROSS REFERENCE: SC22 N2844, N2912

DISTRIBUTION FORM: Def

Address reply to:
ISO/IEC JTC 1/SC22 Secretariat
William C. Rinehuls
8457 Rushing Creek Court
Springfield, VA 22153 USA
Telephone: +1 (703) 912-9680
Fax: +1 (703) 912-2973
email: rinehuls@digex.net

_____ end of title page; beginning of overall summary _____

SUMMARY OF VOTING ON

Letter Ballot Reference No: SC22 N2844
Circulated by: JTC 1/SC22
Circulation Date: 1998-11-30
Closing Date: 1999-04-08

SUBJECT: Second FCD Ballot for FCD 14651: Information technology - International String Ordering and Comparison - Method for Comparing Character Strings and Description of a Common Tailorable Ordering Template

---

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

"P" Members supporting approval without comment	6
"P" Members supporting approval with comment	4
"P" Members not supporting approval	6
"P" Members abstaining	2
"P" Members not voting	4
"O" Members supporting approval without comment	1
"O" 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 comment accompanying the abstention vote from Australia was: "No expertise in this area."

The comments accompanying the affirmative vote from Austria, Canada, France and the United Kingdom are attached along with the comments accompanying the negative vote from Denmark, Germany, Ireland, Japan, the Netherlands and the United States of America.

The comments accompanying the negative vote from Sweden were provided only in pdf format and are being distributed as document SC22 N2912.

_____ end of overall summary; beginning of detail summary _____

ISO/IEC JTC1/SC22 LETTER BALLOT SUMMARY

PROJECT NO: JTC 1.22.30.02.02

SUBJECT: Second FCD Ballot for FCD 14651: Information technology - International String Ordering and Comparison - Method for Comparing Character Strings and Description of a Common Tailorable Ordering Template

Reference Document No: N2844 Ballot Document No: N2844
Circulation Date: 1998-11-30 Closing Date: 1999-04-08

Circulated To: SC22 P, O, L Circulated By: Secretariat

SUMMARY OF VOTING AND COMMENTS RECEIVED

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

___ end of detailed summary; _____

This page is intentionally left blank.

1 Austrian comments

On page 35, paragraph 4, line 1, word 2 should read: "public-domain" rather than "plublic-domain".

2 Canadian comments

Canada SUPPORTS the document with the attached comments:

"Canada wants to make sure that relative weights in the template respect those of special characters as in Canadian standard CAN/CSA Z243.4.1.-1999. Furthermore Canada insists that this International standard shall allow to declare a minimal Canadian delta without having to do prehandling (the delta shall be specifiable simply by declaring a modification of the table), in order to fit with Canadian industry practice. Canada will not accept any change that would jeopardize that objective."

3 Danish comments

We can inform you that Denmark votes NO on ISO/IEC FCD 14651, N 2844 with the following comments:

1. The main table should be included in the standard ad verbatim.

2. The weights on the second level should include a <BLANK> weight for all letters with accents, to ensure as equal treatment as possible of fully composed characters and split-up characters, in non-normalized text. This addresses 6.1.1 note 1, which should be removed.

3. In clause 5, The notation "UXXXXXXXX" should also be allowed.

4. In the main table, the control characters of ISO/IEC 6429 C0 and C1 should be included, and ISO 6429 be added to clause 3, references.

5. in 6.2.2.2 description of level 1, please change "basic letter" to "first-level letter". any basic letters of for example the Latin script are not sorted uniquely at level 1, eg: Æ, Ø, Å. Also for the description of 2nd level: it is culturally dependent what "diacritics" means, and the term should be avoided in an international standard. For example "Ø" and "Å" are not diacritic letter, but base letters, in some languages. There is no diacritic in these letters.

5. in 6.3.1 - the BNF should be terminated with a semicolon.

6. in 6.3.1 rule 13 should also allow for a '<U' eight_digit_hex '>'

7. 6.3.1 and 6.3.2 should be explained in terms of a narrative description as the 14652 LC_COLLATE category specification.

8. 6.3.1 should be aligned with the 14652 BNF for LC_COLLATE,
also in terms of terminology used..

9. There should be tokens "LC_COLLATE" and "END LC_COLLATE" to
surround the whole specification in 6.3.1.

10. 6.3.1 rule 8: space should consist of one or more spaces or tabs.

11. 6.3.1 rule 28: The name should be "section-symbol".

12. in 6.4 references to 6.3.1 terms should be in italic.

13. The examples with reorder-after should use "-" instead
of "_" in the keywords.

14. 6.5 - The name should be following ISO/IEC 15897 naming.

15. in Annex B.1 the line 5 should have <> around TABLE, as in
        order_start <TABLE>;....

16. Annex B.2 : change "assumption that character mnemonics
are resolved into UCS identifiers" to "mnemonic identifiers for UCS
defined in ISO/IEC 14652"

17. Key generation on-the-fly should be described, eg as a note
at the end of 6.1.2, saying that comparison with keys generated
on-the-fly character for character is an equivalent way of
implementing the key generation,and may eliminate elaborate
key generation when a difference is to be found in the first few
characters.

18. Position should be specifiable on all levels, as it is legacy from
POSIX.

19. Toggles "ifdef" etc as in 14652 should be reintroduced.

20. The conformance clause needs to be reformulated. It should not
be possible to claim conformance to 14651 if full tailoring
is not available with the application. That would mean that
eg. Danish specifications cannot be accomodated by the application
and that defeats the main purpose of this standard. The conformance
clause does not read as English. Ith should also be possible
for a specification to claim conformance - possibly in the
way of 6.4 tailoring.

21. The Danish test data in annex B should be replaced with the
following:
A/S
ANDRE
ANDRÉ
ANDREAS
AS
CA
ÇA
CB
ÇC
DA

ÐA
DB
ÐC
DSB
D.S.B.
DSC
EKSTRA-ARBEJDE
EKSTRABUD
EKSTRAARBEJDE
HØST
HAAG
HÅNDBOG
HAANDVÆRKSBANKEN
Karl
karl
NIELS JØRGEN
NIELS-JØRGEN
NIELSEN
RÉE, A
REE, B
RÉE, L
REE, V
SCHYTT, B
SCHYTT, H
SCHÜTT, H
SCHYTT, L
SCHÜTT, M
ß
SS
SSA
STORE VILDMOSE
STOREKÆR
STORM PETERSEN
STORMLY
THORVALD
THORVARDUR
ÞORVARÐUR
THYGESEN
VESTERGÅRD, A
VESTERGAARD, A
VESTERGÅRD, B
ÆBLE
ÄBLE
ØBERG
OBERG

4 French comments

France votes YES on FCD 14651, with the following comment:

Insufficient effort has been done to define an acceptable ordering for some lesser-used scripts.

A lot of scripts are actually ordered based just on Unicode code values. When WG20 can find some existing practice of a culturally accepted

ordering not conflicting with another one, these practices should be included in FCD 14651 default template ordering.

We suggest that experts of those scripts should be invited to define a correct default ordering.

For example, this is the case for Tamil (like most other indic scripts) and Thai scripts, where evidence of existing practice has been demonstrated and no evidence of other equally valid practice has been found.

However, considering these issues are more of a concern for national bodies where those scripts are in widespread use, and even if there is a Tamil community in the French territory Reunion Island,
we suggest that this work should be done, perhaps in a future amendment to this forthcoming standard.

As the same problem exists with any new codepoints added in the UCS, we also suggest that we should contact ISO/IEC JTC1/SC2/WG2 to ensure the existing procedures to register new characters are adjusted to include the needed informations to update the forthcoming collation standard.

5 German comments

The German member body vote is "No" with comments.

If the technical comments are resolved satisfactorily, the German "no" vote will be changed to a "Yes" unless other significant changes be made to the standard in an unsatisfactory way.

5.1 General

Germany wishes to thank the editor for many fundamental improvements of this draft over the previous FCD. They greatly increase the usefulness of the future standard and render void many essential German concerns.

German comments touch upon two principal points:

Technical comments on the body of the draft and on Annexes_B-E;
Comments on the normative Common Template Table (Annex_A).

Germany does not comment on matters of English style as it is expected that this will be improved by native English speakers. Lack of explicit comments on this should not be taken as endorsement of a style that is, as yet, not always a paragon of clarity. There are many paragraphs where "loose ends" are noticable, caused probably by numerous cuts and reworkings over time. Furthermore, Germany does not comment on purely typographic deviations from the ISO drafting rules (e._g. semicolons ought to be used to terminate items of unordered lists). It is confident that these points will be addressed by the editor at a later stage.

5.2 Comments on the body of the draft

5.2.1 Introduction, 2nd paragraph

This paragraph should best be removed altogether, or at least reformulated in such a way that it does not imply any more that the syntax of the Common Template Table (hereafter CTT) is in any way normative. The current formulation of the whole paragraph is unfortunate in this respect. The draft does not -- and must not -- mandate that conformant applications can either directly exchange ordering specifications or even use the CTT in the syntax used in Annex_A.

To stress this point, it is advisable to add another annex with the specification of another possible syntax. The XML-conformant Swedish suggestion can serve as a useful starting point.

5.2.2 Introduction, 4th paragraph

Remove 2nd sentence.

5.2.3 Scope: 1st dash

Remove text in brackets ["(independently of coding)"]. Change the formulation in the remainder of that paragraph to stress that mappings from ISO/IEC_10646 to any other coding scheme are also permissible.

5.2.4 Scope: 2nd dash

Remove phrase "using a variant of the Backus-Naur Form (BNF)" as the reference format as such does not use the BNF. It is simply defined using the BNF syntax.

5.2.5 Scope: Note

Remove note.

5.2.6 Scope: Additions

Add an entry under the heading "This International Standard does#/+not#/-mandate" to stress that no preparatory procedures are prescribed, but is normally necessary. Give a reference to Annex_C.

5.2.7 Definitions: 4.9

The term depth does not elucidate the problem but rather explains an X with an Y. Either define the term or chose a different formulation.

5.2.8 Definitions: 4.10

The reference comparison method should be defined or explained in more detail before.

5.2.9 Definitions: 4.11</CommentOn>

In the context of this draft the "set of strings" can always be understood as having one and only one member (no preparatory procedures are part of the standard itself). Therefore change the formulation accordingly.

5.2.10 Definitions: 4.11 (suggestion)

Replace the word order by sequence and reformulate the phrase accordingly.

5.2.11 Symbols and abbreviations

Simplify the matter of code-dependence on ISO/IEC_10646. Any application is conformant that is able to achieve identical results as those of section_6, but not necessarily in the same way. A mapping between some encoding system and the UCS and back can be seen as a special case of the preparation of character strings (cf._6.1.1) and of the presentation of the resulting sequence after ordering. Therefore, without loss of generality, a character can be seen as being part of the UCS. In consequence, the 2nd paragraph except the last sentence should be removed and the 3rd paragraph can be reformulated accordingly, i._e. it can refer to the private-zone UCS coding without further preconditions.

5.2.12 Requirements: 6.1.1</CommentOn>

Clarify 1st sentence of the 2nd paragraph. Recommendation: <recom>At minimum, the preparation shall guarantee that either only precomposed characters or only combining sequences, which in the context of the conformant application are deemed equivalent, are presented to the comparison method ...</recom>

5.2.13 Requirements: 6.2.2.1

This section is not explained in necessary detail and clarity. Concepts like stacks are suddenly implied ("stacking of the token will be done"), push and pop operations appear. None of these operations have been referred to before nor are they explicitely used thereafter.

Technically, the algorithm which the editor obviously has in mind, is, of course, correct. It should, however, be elaborated in more detail. The reader which the editor should have in mind here is the programmer who knows basic devices, but has never worked on ordering.

Typographically, it is difficult to understand why the three paragraphs in question are printed with identation.

5.2.14 Requirements: 6.2.2.2

The part from Generally to the end should be handled as a note or alternatively as a section (6.2.3) of its own.

Level_3: The topic of #/+variant character shapes#/- ("modified letters") must be dealt with on level_2 to ensure maximal compatibility with pan-European requirements. It has no conceptual likeness to "case" and is not normally used on level_3 (cf._also the tayloring of Informative Annex_B.1).

5.2.15 Requirements: 6.3.2

Make all text of the explanatory [I.e....]-statements into notes to stress their informative character or consider other means to achieve that end. Such a solution might be to add an informative annex that explains these and other points which concern the syntax of the CTT.

5.2.16 Requirements: 6.3 and WF1

<tt>hex^_symbol</tt>'s are not defined.

5.2.17 Requirements: 6.3.3, items I4 to I6

The terms normal form, evaluated [weight table] and collation-element-weighted are implicitly defined here, but are used nowhere else. Either the definitions are considered to be of sufficient importance to be included in the "Definitions"-section proper or they should be removed altogether. In part, they can also be incorporated in the specifications themselves, as they explain some requirements more concicely then the corresponding specification itself.

5.2.18 Requirements: 6.4

Remove 2nd sentence of 1st paragraph.

5.2.19 Annex_B.2

Align the presentation of the delta with that of Annex_B.1 (as it stands the presentation is not conformant to 6.4) and remove all references to the mnemonics which are altogether irrelevant in this context.

5.2.20 Annex_C (general)

Add a remark on the importance of higher level protocols (e._g. markup system SGML) for correct evaluation of numerals and other prehandling objects (e._g. units -- keys -- in a phone book). Context rarely suffices to achieve anything like #/+total certainty#/-. Many of the tasks are quite trivial if we assume an internal tagging like ^<TemperatureInC^>-9^</TemperatureInC^> (cf._C.2.4), but bordering on the impossible to solve reliably without them (In C.2.4 the word Temperature: can be regarded as an implicit tag, but most texts are not nearly that schematic as the examples in this annex assume).

It is to be considered if Annex_C really needs to be quite as detailed and extensive as it currently is.

5.2.21 Annex_C.1, 1st dash (minor)

Why are the names of the strings in capitals?

5.2.22 Annex_C.1, 2nd dash (minor)

The example text is somewhat obscure (e._g. the remark "according to noble origin or not" presupposes knowledge that this is of importance when ordering).

5.2.23 Annex_C.2

The text needs to be clarified to some extend (e._g. what are "Run-together numerals"?).

5.2.24 Annex_C.2.2

A cautionary note should be added to stress that these preparatory steps have in some cases (e._g. ordering of telephone numbers in phone books)

undesirable consequences and should then be avoided.

5.2.25 Annex C.2.3, 3rd paragraph

The 2nd sentence ought to be modified. "total certainty" can rarely be achieved even with information on the context.

5.2.26 Annex_D, item V.2

Change the formulation of the last sentence of the 1st paragraph. German dictionaries usually employ the German norm DIN_5007. Some dictionaries explicitely refer to this norm, others simply use it without further clarification, still others explain their ordering principles in some detail.

5.2.27 Annex_D, item V.3

Remove phrase for the first time in the fourth paragraph.

5.2.28 Annex_D, item VII

Remove this item.

5.3 Comments on Annex_A: Common Template Table</H1>

5.3.1 General: Names of internal symbols

Either reduce all names to a maximum of five letters for consistency or (preferably) give less cryptic names to all of them (e._g. <tt>^<MACRON^></tt> instead of <tt>^<MACRO^></tt> and <tt>^<DOUBLE^_TILDE^></tt> instead of <tt>^<D0360^></tt>). Names should best be derived from their description in the UCS.

5.3.2 Variant letter shapes

As mentioned above, variant letter shapes must be distinguished on level_2 instead of level_3. Letters such as <tt>F WITH HOOK</tt> (<tt>^<U0192^></tt>) should best be treated as second level letters. Ideally, only a-z and thorn should be treated as first level letters, though Germany sees this last statement as a strong suggestion for discussion.

Relative order of scripts (point of discussion)
It is seriously to be considered if the relative order of scripts should not follow a general East-to-West scheme as proposed by the last UK comments. This could easily be achieved by "internal tailoring" the CTT as already done for the special characters of CAN/CSA_Z243.4.1-1998. Germany sees this, however, only as a strong suggestion for an internal discussion in WG20.

5.3.3 Script: Greek

Maximum compatibility with the specifications of ELOT as presented in WG20/NXXXX is to be sought. To achieve this the breathing marks Psili and Dasia should precede the other diacritics. This is also in line with usual Greek (cf. the study CEN/TC304/Nyyy. <tt>COMBINING COMMA ABOVE</tt> and >tt>COMBINING REVERSED COMMA ABOVE</tt> (with which Psili

and Dasia are -- unwisely -- unified in the UCS) are diacritics which appear infrequently in languages other than Greek, whereas in Greek they are very frequent indeed. Cf._also the approach of the E.

5.3.4 Script: Cyrillic

The order for Cyrillic is not in line with pan-Cyrillic requirements and contains numerous errors. The sequence must be brought in line with the specifications from GOST as reflected in the current edition of the European Ordering Rules (cf._EOR). Detailed documentation both from GOST itself and from other sources will be made available to WG20 before the May meeting.

5.3.5 Script: Georgian

The ordering of Georgian should be coordinated with the results of ongoing discussion with experts in the field both from Georgia itself and in academic organizations.

6 Irish comments

Although Ireland voted positively on the draft on 1998-01-26, we now wish, because of subsequent review of the document, to reverse our position. Ireland votes No on the FCD draft.

Many of our objections are editorial in nature, and we believe that our No vote can be turned back to Yes easily if the following points are addressed appropriately by SC22/WG20:

6.1 Requirements for YES vote:

1 The English text must be revised so that it is in all cases unambiguous and grammatically correct.
2 Informative text in the Common Template must be revised so that the implication is not made that French backwards-ordering of accents is not a special case.
3 The assertion that small letters ordered before capital letters is the normal practice for the English language is not made and is removed from informative annex D.
4 The Canadian and Danish example benchmarks must provide enough examples to interpret the specifications from which they are derived.
5 The Common Template should contain orderings for all Amendments to 10646 up to Amendment 31, not up to Amendment 7. Ogham, Cherokee, and Runic are already in order (except for the Ogham and Runic punctuation); Canadian Syllabics will require some work to get it right.

6.1.1 1. Editing for proper English

We have remarked on earlier drafts of this International Standard that the use of the English language is in many cases either ambiguous or grammatically incorrect. We had offered to prepare a corrected version, but because text was not provided to us in time before the last meeting WG20, we were forced to withdraw our offer of making the corrections. We offer now again to provide a new version with document revision annotations. We feel strongly about this because in reviewing the draft, we were often forced to stop and read aloud certain passages in order to decipher the intended meaning. Examples of grammatically incorrect or ambiguous sentences:

1 It is demonstrated that by tailoring the Common Template Table to add extra token values at level 2 for all precomposed characters affected by a ~~diacritics~~ diacritic, it is possible to accomplish identical results for combining sequences without requiring that preparation.
2 The scanning properties for the level *i* being processed needs to be carefully monitored. When there is a change in scanning direction at level *i* (~~this implies~~ implying that the character being processed

comes from a block ~~that~~ which is different from the preceding character processed and which has different scanning properties) and the new direction is backward, stacking of the token will be done at the position where the change of direction has occurred.

3 If the `order_start_entry` does not ~~uses~~ use the position value at level *m* of a block (the ***`position`*** value is explicitly used in the template for the only block defined) then the formation of subkey level *m* is done in exactly the same way as the above-defined formation.

4 WF7. No two ~~*section_definition_entry's*~~ instances of *section_definition_entry* in a *tailored_table* may contain the same values in their *section_identifier's* instances of *section_identifier*. [~~I.e.~~ That is, multiple definition of section's is prohibited; *section_identifier's* instances of *section_identifier* must be unique.]

5 [~~I.e.,~~ That is, if one takes two strings, builds keys for each based on table 1 and compares them, one should always get the same results as when one builds keys for them based on table 2 and ~~compare~~ compares them.]

6 In cases where ~~the applications~~ an application has provision to allow the end-user to tailor the table himself or herself, any statement of conformance shall indicate which ~~ones~~ of the 4 elements of the previous list are tailorable and which ~~ones~~ are not tailorable.

7 Whenever the Common Template Table is ~~refered~~ referred externally as a starting point in a given context, either applicative or contractual [WHATDOESTHISMEAN???], it shall be referenced using the name ISO14651_1999_TABLE1.

8 For very ~~big~~ large, or very ~~tiny~~ small, values, one often uses formats like 2.5*107 ~~(to just pick one possible way of writing these for the purposes of the examples here)~~.

9 But the Common Template Table ~~has digits as~~ specifies digits to be level 1 significant.

10 Such processing is beyond the scope of this International Standard, ~~though~~ however.

11 A ~~plublic-domain~~ public-domain reduction technique is described in ~~details~~ detail (with ~~ample~~ numerous examples) in *Technique de réduction - Tris informatiques à quatre clés*, Alain LaBonté, Ministère des Communications du Québec, ~~June 1989~~ 1989-06 (ISBN 2-550-19965-0).

12 To illustrate this ~~(without discussing context analysis which is not necessary in what follows)~~, examples of dictionary sequences are given here for two languages ~~which~~ whose native order is not in the Common Template table:

6.1.2 2. The Common Template states:

```
% To tailor for French accent handling, or not to make French
% a special case add an order_start statement
% and order_end for Latin in the Latin section, as follows:

% order_start Latin;forward;backward;forward;forward,position
```

In Ireland we consider French to be a special case, which in fact yields incorrect sorting for our first official language, and we disagree with the implication here, namely, that "not making French a special case" does no harm. French is a special case of the default template, just as Danish and Swedish are. The Common Template must read:

```
% To tailor for French accent handling, add an
% order_start statement and order_end for Latin
% in the Latin section, as follows:

% order_start Latin;forward;backward;forward;forward,position
```

6.1.3 3. Annex D states:

3. The third decomposition breaks ties for quasi-homographs different only because upper-case and lower-case characters are used. This time, the tradition is well established in English and German

dictionaries, where lower case always precedes upper case in homographs, while the tradition is not well established in French dictionaries, which generally use only accented capital letters for common word entries. In known French dictionaries where upper and lower case letters are mixed, the capitals generally come first, but this is not an established and stated rule, because there are numerous exceptions.

This is, as we have said many times to SC22/WG20, incorrect. Lower case does not precede upper case in English. *The concise Oxford dictionary of current English*, cited in the JTC1 and CEN directives as a standard for the English language, consistently gives, in its 8th edition (1990) and its 9th edition (1998) the following:

August (month)	May (month)
august (venerable)	may (be able)
March (month)	Polish (of Poland)
march (tread)	polish (shine)
Mass (ritual)	
mass (heap)	

So for a Common Template it is advisable to use English and German traditions, if one wants to group the largest possible number of languages together.

This rationale is therefore unacceptable, as it is untrue. The reason the Common Template has smalls before capitals (which we do not prefer) is because that is what is specified in the Unicode template. This text must be revised.

Let's note here by the way that in Denmark, upper case comes before lower case, a different but well established rule. This is a second fact calling for adaptability in the model used in this standard.

This same rule is used for the English language.

Example: to have the following order: "august", "August", numbers could be assigned indicating respectively "llllll", "ullllll", where "l" means lower case and "u" upper case.

This example is not sufficient. The actual syntax for ordering smalls before caps which appears in the Common Template should be repeated here, along with the actual syntax for ordering caps before smalls.

6.1.4 4. Canadian delta

The Canadian delta specifies treatment of THORN and ETH but the benchmark does not contain examples containing these characters. Please add: ¨orsmörk, Thorvardur, ¨orvar±ur, medal, me±al. The Danish benchmark examples of REE and RÉE are not sufficient to demonstrate E vs. É. Please add more examples as well as examples of such as Ree and Rée.

6.1.5 5. Examples

The draft is a bit overloaded with references to English, French, and German. A few more examples from other languages would be preferred.

7 Japanese comments

Subject: Japan's vote on SC22N2844

Comments on FCD 14651.2

The National Body of Japan disapproves FCD 14651.2 for the reasons below.

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

7.1.1 J.1) Global:

This draft contains many errors and is too difficult to understand because it throws away a great deal of the material developed in FCD 14651.1 and the LC_COLLATE section in FCD 14652.1.

Japan agreed to make FCD 14651.2 independent of 14652 assuming that the well discussed and sophisticated part of 14652 would be imported in the second FCD thus enabling us to review it as FCD. But the current draft is far from that. We request to put it back to a mixture of FCD 14651.1 and the LC_COLLATE section in FCD 14652.1 which have been studied by many people. If our request is rejected, the project should be put back to the CD stage.

7.1.2 J.2) Global:

There are many inconsistencies about tailoring and "delta".
Japan considers that the following principles should be reconfirmed in the FCD disposition before any other detailed discussion:

a) The Common Template Table (CTT, hereafter) is not a table to be used by the ordering method -- the CTT always needs tailoring.

b) Tailoring is always described as a delta to CTT.

c) The tailored table is a result of applying a delta to CTT,

d) The tailored table is a table assumed in the reference method description.

7.1.3 J.3) p.iv, Introduction, the first sentence:

The sentence

> This International Standard provides a method for ordering text data worldwide, and provides a Common Template Table whose tailoring eases adaptation of a specific script while retaining universal properties for other scripts

should be changed to

> This International Standard provides a method for ordering text data worldwide, and provides a Common Template

Table whose tailoring eases adaptation for culturally specific handling of some scripts with minimal efforts.

because tailoring of the Common Template Table usually deals with two or more
scripts and the wording "universal properties for other scripts" may be interpreted as if there were an universally accepted set of collating properties for each script.

7.1.4 J.4) p.1, 1 Scope, bullet 1:

In the first bullet

- A simple method of reference for comparing two characters strings in order to determine their respective order in a sorted list. The method is applicable on strings that exploit the full repertoire of ISO/IEC 10646 (independently of coding).

"10646" should be changed to "10646-1" because the syntax "Uxxxx" allows only to refer to BMP.

7.1.5 J.5) p.1, 1 Scope, bullet 1:

The sentence

> This method uses transformation tables derived from either the Common Template Table defined in this International Standard or from one of its tailorings.

should be changed to

> This method uses transformation tables derived from table specifications tailored from the Common Template Table defined in this International Standard.

because the Common Template Table without tailoring should not be used as a source of transformation tables.

7.1.6 J.6) p.1, 1 Scope, bullet 4:

7.1.7 p.11, 6.5 Name of the Common Template Table:

The fourth bullet in the scope and the subclause 6.5 should be removed because defining the reference name for Common Template Tables is not a matter of this standard but a matter of the referencing systems.

> NOTE) The addition of the reference name does not depend on the NB comments to the first FCD.

7.1.8 J.7) p.1, 1 Scope:

Add a bullet

> - Requirements for a declaration of the differences between the comparison table used in applications and the Common Template Table,

in order to cover the contents of subclause 6.4.

7.1.9 J.8) p.2, 2. Conformance:

An application is not appropriate as a target for defining conformance. We propose to define the conformance of "a text data", "an ordering service with built-in table", and "an ordering service without built-in table" as follows:

> 2 Conformance
>
> The order of a text data according to a declared tailored table is conforming to this International Standard if the text data coincides with the output of the referenced method prescribed in clause 6. with some input data and the tailored table input.
>
> An ordering service with a built-in and declared tailored table is conforming to this International Standard if the order of each output for an input data according to the built-in tailored table is conforming to this International Standard.
>
> An ordering service without built-in table is conforming to this International Standard if the order of each output data for a pair of an input data and a declared tailored table is conforming to this International Standard.

7.1.10 J.9) p.2, 2 Conformance:

> NOTE: This comment needs not be considered if the comment J.8 is accepted.

The sentence

> More specifically, it is the responsibility of implementers to show how their delta declaration is related to the table syntax described in clause 6.3, and how the comparison method they use.

should be simplified to

> More specifically, it is the responsibility of implementers to show how their delta declaration is related to the table syntax described in clause 6.3.

because the phrase "how the comparison method they use" is not grammatically correct and implementers need not to make open their inner mechanisms if only their outputs are conforming.

7.1.11 J.10 p.2, 2 Conformance:

> NOTE: This comment needs not be considered if the comment J.8 is accepted.

The sentence

> Any declaration of conformity to this International Standard shall be accompanied by a declaration of the tailoring delta described in clause 6.4 in case tailoring is not provided by the concerned application

should be changed to

> Any declaration of conformity to this International Standard shall be accompanied with a declaration of the tailoring delta described in clause 6.4

because the Common Template Table will not be in work without tailoring.

If this request is rejected, the words "in case" in this sentence should be replaced by the word "unless".

7.1.12 J.11) p.2, 2. Conformance, 2nd para.:

> NOTE: This comment needs not be considered if the comment J.8 is accepted.

The last sentence, which lacks the subject, should be removed because it is covered by the first sentence of this clause.

7.1.13 J.12) p.3, 4.7 "glyph", 4.8 "graphic character":

The second sentence in 4.8 "graphic character" should be removed because its meaning is already introduced in the first sentence by "that has a visual representation ..."

The definition 4.7 "glyph" should be removed because it is used only in 4.8 thus the first part of the following UK comment on the first FCD

> 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.

becomes meaningless now.

7.1.14 J.13) p.4, 6.1.1 Preparation of character strings:

This subclause 6.1.1 should be put out of the subclause 6.1 (say the new clause 7) because the subclause 6.1.1 discusses about the outside of the reference method.

7.1.15 J.14) p.4-7, 6.2 Building the ordering key used in the reference comparison method:

Although there are descriptions for building subkeys, there is no description for building a numeric key to be used in 6.1.

Japan considers that the drastic change of the algorithm from the first FCD produced many fatal deficiencies.

Japan recommends to put back the whole content as a merge of FCD 14651.1 and the related part of CD 14652.

7.1.16 J.15) p.7, 6.3 Common Template Table: formation and interpretation:

The relation between the syntax defined here and the semantics in the previous subclause is too poor as a standard and this subclause 6.3 contains many errors in itself. See the detailed comments below.

J.15-1, Global) The production rules should be presented in a top-down manner.

J.15-2, Global) The names of the terms should be exactly the same as are used in other places e.g. the name "untailored_template_table" in Rule 46 should be changed to "common_template_table".

J.15-3, Rule 44) The two lines in CTT

```
section CANSpecials
```

and

```
reorder-section-after CANSpecial <U001F>
```

are illegal according to the BNF. They should be changed as simple_line's or they should be removed from CTT.

J.15-4, Rule 24, 20) The multiple symbol weight definition in CTT such as

```
<U4E00>..<U9FA5> <S4E00>..<S9FA5>;<BLANK>;<MIN>;<U4E00>..<U9FA5>
```

is illegal according to the BNF. The production rules should be supplied

J.15-5, Rule 24) "line_completion" should be removed.

J.15-6, Rule 14, 13, 12, 11, 5, 6) From the current definitions, all the ucs_symbols are recognized also as simple symbols.

J.15-7, Rule 41, 40) The lines consisting of "line_completion" only are recognized as "simple_line" and "tailoring_line".

J.15-8, Rule 38) Remove the second appearance of "space" in order to match with CTT.

J.15-9, Rule 38) There is no explanation throughout this document for the use of "identifier" here.

J.15-10, Rule 28) "line_completion" should be removed.

J.15-11, Rule 29) "line_completion" should be removed.

J.15-12, Global) The functionality which is supported by "collating-element" should be supported as a tailoring line.

J.15-13, Rule 1, 10) Make clear that "line_delimiter" is not included in "character".

J.15-14, Rule 43) This production rule should be removed because it is not referenced.

J.15-15, WF1) This condition should be modified to

> WF1. Any "simple_symbol" occurring in a "multiple_level_token" must be defined in a "symbol_definition" line in the table.

because there may be a "symbol_weight_entry" such as

> <a> <a1>;<a2>;<a3>;<a4>

where <a1>, <a2>, <a3>, or <a4> needs to be greater than <a>.

J.15-16, WF1) The term "hex_symbol" does not appear in BNF. It should be changed to "ucs_symbol".

J.15-17, WF2) This condition should be replaced by an explanation

> An empty level_token shall be interpreted as the collating element itself.

in the same way as POSIX because the current condition prohibits defining a collation which needs more than four levels.

If this proposal is rejected, the sentence

> All multiple_level_token's in a tailored_table must contain the same number of delimited_level_token's

should be changed to

> All multiple_level_token's in a tailored_table in a normal form (see I4 later) must contain the same number of delimited_level_token's

J.15-18, I1) The text should be changed as follows:

> I1. There are two types of sections.
> One type, "simple definition", consists of the list of simple_line's
>
> following a section_definition_simple_entry in a tailored_table.
>
> Another type, "list definition", is defined by a "section_definition_list_entry". It is equivalent to a "simple definition" consisting of a list of "symbol_definition" lines which are regarded as an expansion of the symbol_list.
>
> Example)
>
> section FOO <ABC>;<DEF>;<GHI>
>
> is equivalent to
>
> section
> <ABC>
> <DEF>
> <GHI>
> (non simple line)

J.15-19, I2, I3) Usage of the word "same" here is confusing.

J.15-20, I2, I3, I4)

The explanations for tailoring here need some improvements because applying a number of operation sequentially causes a problem of their order and side-effects.

For example, when a symbol <Uxxxx> in CTT is redefined by a "reorder-after" directive and the symbol is a target symbol in a successive operation, it is not clear which position, old one's or new one's, is preferred.

J.15-21, I5) It should be explained how to deal with multiple occurrences of a symbol to be evaluated -- e.g. only the last one should be valid.

J.15-21, I6) The term "hex_symbol" does not appear in BNF.

J.15-22, I6) The sentence

> All hex_symbol's are assumed to map to an integral
> weight value equal to that hex_symbol interpreted
> as a hexadecimal number

is a source of problems. The term "hex_symbol" does not appear in BNF.
If hex_symbol's are equivalent to ucs_symbol's or ones like <S0200> in
CTT, the sentence is wrong
because ucs_symbol's and ones like <S0200> should be numbered in the
sequence of table lines along with simple_symbol's and their numbers
have no relation with the hexadecimal values except the incremental
nature in each range specification.

J.15-23, I6) The sentence

> All hex_symbol's (ucs_symbol in our understanding!)
> are assumed to map to an integral
> weight value equal to that hex_symbol interpreted
> as a hexadecimal number

is wrong, because ucs_symbol's should be mapped to an integral also
in the sequence of table lines along with simple_symbol's and
the values have no relation with the hexadecimal values.

J.15-24, Rule 19) CTT includes many lines which have two or more "space"s
immediately before "comment".
They should be modified or the BNF should be modified.

J.15-25, Rule 5, 11) CTT includes illegal identifiers such as

```
<2AIGU>  % COMBINING DOUBLE ACUTE ACCENT
<2GRAV>  % COMBINING DOUBLE GRAVE ACCENT
```

They should be modified or the BNF should be modified.

J.15-26, Rule 21 and other places) The Rule 21 allows an expression like

```
<ABC>..<XYZER>
```

It should be clarified in syntax or in well-formedness or in interpretation
what are allowed for "symbol_list_item_range" and how they are interpreted.

7.1.17 J.16) p.10-, 6.4 Declaration of delta, 1st sentence:

The first sentence

It is recommended that tailoring be done starting with the
Common Template table described in annex A.

is wrong because all the tailoring shall start from the Common Template Table.

If this standard allows to define some collating specification from the scratch, there are many places to be changed.

7.1.18 J.17) p.17, Annex B.2, Example 2 - Danish delta and benchmark:

This is a wrong example because it contains no valid order_start entry and it contains some illegal lines starting from "collating-element".

7.1.19 J.18) p.10, 6.4 Declaration of a delta:

p.12, Annex A Common Template Table:

Two of the three toggling switch, which was the major achievements until the first FCD and got no NB comment to remove them, are omitted in this draft.

It should be revived in 6.4 and Annex A.

7.1.20 J.19) Global:

The word "conformant" should be replaced with the word "conforming".

8 Netherlands comments

22N2844 FCD14651
International String Ordering and Comparison
Method for Comparing Character Strings
and Description of a Common Tailorable Ordering
1999-04-08 DISAPPROVAL WITH COMMENT

The NNI votes NO on FCD 14651 for the reasons detailed below.
The vote from the NNI will turn into yes when the defects indicated below have been repaired.

8.1.1 -1-

Apart from FCD 14651, another document standardizing string sorting is available:
Draft Unicode Technical Report #10: Unicode collation algorithm
Comparing both documents, the following (partial) reasons for a NO-vote appear:
-a-

The Unicode Report is much clearer and better defined than the 14651 document.

-b-

Both documents describe the algorithm(s) in informal English.
It is therefore impossible to present a formal reasoning or mathematical proof that the algorithms are equal (if they are supposed to be) or are not equal and implement different functionality (if they are supposed to be different) It is similarly impossible to proof that a program correctly implements one of these algorithms (or both algorithms).

-c-

It seems that both descriptions are not equivalent.
There seem to be differences in particular regarding level 4.
This is said with some prudence given the issue -b- above.

Summary of -1-:

The NNI is of the opinion that the world has no need for having two (almost) equal sorting standards. The current situation is seen as a source of confusion and a waste of standardization resources.
The NNI thinks that only one of these developments should be continued.

8.1.2 -2-

Quite some comments have come in on the previous FCD.
This has led to a large delta between the previous and the current document. Because this delta was to be expected, the NNI had requested that the current document is issued as a CD instead of an FCD.
WG20 has decided to issue an FCD, therewith neglecting what the F in FCD stands for.
After this round, a similar delta is to be expected. The NNI therefore repeats its request to issue the next document as a CD.

8.1.3 -3-

The previous document contained many unclear definitions and clauses.
While some improvement has been noticed, the rewriting that has taken place has introduced many new ambiguities.
Below we will first give some general remarks and then some remarks related to the paragraphs in the document.

8.1.4 General remark 1:

There are still quite a few sentences in the document that are clearly not written in proper English. This makes the document difficult to understand.

8.1.5 General remark 2:

There are quite a few occurrences of words that do not belong in an IS. We mention just a few: minimum of efforts, fundamental choices, highly recommended, straightforward, challenge, simple, a lot of, excellent, carefully.

8.1.6 General remark 3:

The precision of definitions and wording still leaves much to be desired. Some of the detailed issues below are consequences of the textual ambiguities in the document.

Detailed remarks:

8.1.7 Re Introduction:

There is still confusion about the precise meaning (or difference in meaning)
of 'ordering', 'collation' and 'comparison'.

The example of 'English as a poor exception' sounds negative
and is unintelligible.

8.1.8 Re 1 Scope:

Is 'a method of reference for comparing two character strings' (first dash) the same as 'the comparison method' (third dash)?

....any equivalent method giving the same results is acceptable.
Are there equivalent methods giving different results?
Are there non-equivalent methods giving the same results?

8.1.9 Re 2 Conformance:

section => clause

paragraph 2: crippled English

8.1.10 Re 3 Normative References:

8859 and 14652 are missing.

8.1.11 Re 4 Definitions:

The notions of 'object', 'element', 'comparison element' and 'internally' have not been clarified.

4.10 discusses 'the reference comparison method'. Is this the same as 'a method of reference' in clause 1?

4.11 states that ordering affects two SETS OF strings, whereas clause 1 states that ordering affects TWO STRINGS.

8.1.12 Re 6 Requirements:

6.1 states 'Reference method' whereas 6.1.1 states 'comparison method'
Are these the same?

Although not part of the scope of this IS, ...
It is unclear whether this part is normative or not.
If this part is not normative, requirements as presented under 6.1.1 should be moved to an informative annex.

....described in 6.1....
This is unclear as this is clause 6.1.

...are meant to be equivalent.

The notion of equivalent is unclear.

6.1.2 .....the algorithm of key formation described in clause 6.2 ...
6.2 does not describe 'key formation'; 6.2.2 describes 'key composition';
has that been intended?

6.2.1.1
We have here 'ordering table', 'transformation table' and
'matrix of n lines'. None of these notions is particularly clear;
in particular the last one is quite ambiguous.
It seems only one notion would be sufficient.
For a precise notion, WG20 is referred to the notion
of 'map' as used in VDM-SL.

6.2.1.2
...A tailored table may be separated into blocks.
This seems to imply that a non-tailored table may not be separated
into blocks. This seems odd.
'May' is not allowed in an IS.
The notion of a block is unclear. Is a diagonal sub-matrix a proper block?

6.2.1.2 Note:
The notions of 'logical sequence', 'presentation sequence' and 'logical
order of the presentation forms(?)' are unclear.

6.2.2 Key composition:
The notion of 'comparison field' is unclear.
The notion of 'successive sequence' is unclear.

The whole issue of 'stacking a token' and 'push position' is unclear.
As far as understandable, the stack seems never to be popped; the use of
the values in the stack stays unclear.

The discussion under 'Level 4' is incomprehensible.
Additionally, it is unclear what differentiates 'logical string sequence'
from 'logical sequence'.

6.3.1 BNF Syntax Rules:

This is NOT BNF; it is not EBNF either, but a local variation.
Why not use the SC22 document available?

There are various kinds of quotes in this table.

I5. .... order in this file.
It is unclear which file is used here.

It would have been most helpful when the notion of a block as introduced
in clause 6.2.1.1 would have been present in the BNF.

The notions of combining character and precomposed character have not been
defined.

6.3.4
C1. (full stop missing)
C1. Two collation weighting tables...
What on earth are these?

```
... is exactly matched by ...
What is the difference between
'exactly matched', 'exactly equal' and 'equal'?

6.4 Declaration of a delta:
...14652, which uses a syntax that is compatible with the one described
in this IS.
Why having two partially overlapping standards?

...that occur in the comparison table used relatively to the Common
Template Table if a fixed table is ...
The number of tables gets (relatively) overwhelming.

....as defined in 6.2.1 => 6.3.1 (two times)
```

8.1.13 Re Note:

```
It is unclear why two imprecise forms are allowed here when a precise
one is available also.
```

8.1.14 Re Annex A:

```
It is unclear what a 'common template' is.
```

8.1.15 Re Annex B:

```
It seems the lines containing
  order_start TABLE;forward;backward;forward;forward,position
cannot be derived from the BNF.

It seems the line
  copy ISO14651_1999_TABLE1
cannot be derived from the BNF.

It seems the lines containing sequences of <U....> cannot be derived from
the BNF as line 15 of the BNF requires double quotes.

There are some formatting problems here.
```

9 Swedish comments

```
Secretariat Note:  The Sweden comments are contained in document SC22
N2912.
```

9.1 Definitions (major comment)

The definitions (section 4) are not always to the point, and sometimes unclear. **Please change the definitions to something very close to the following** (and alter subsequent text accordingly):

abstract glyph	a recognizable abstract graphic symbol which is independent of any specific design.
character string	a sequence of (coded) characters (((considered as a single object?)))

collation ordering of elements based on ordering of character strings.

collation delta list of differences for a specific collation table relative to one of its ancestor template collation tables. Each collation table can have only one immediate ancestor.

collation element sequence of *n* weight strings, where *n* is the number of levels in the collation table. The weights may be given as symbolic weights.

collation item non-empty sequence of characters that has an entry in the collation table.

(collation) key a real value (strictly) between 0 and 1, formed by concatenating the collation subkeys for a given string after an initial '0.', and regarding the result as a fractional numeral (in the radix of the digits used). The reference method puts a level separator weight between each pair of the concatenated subkeys. The collation keys 0 and 1 can be used as special collation keys, respectively strictly less than and strictly greater than any collation key formed from any character string by the reference method. (Note that hardware supported floating point datatypes are not suited for representing these values, since these datatypes rarely will have sufficient precision, unless the strings compared are limited to two or three, maybe four, characters.)

(collation) level whenever used without qualification in this International Standard, *level* stands for the number of the 'pass' done over a string to compute its reference collation key.

collation subkey a sequence of weights computed for a character string for a particular level.

(collation) preparation

a process in which character strings are mapped to (other) character strings logically before using the key calculation specified in the reference method of this International Standard.

(collation) weight length *b* digit sequence. For the reference method, the value of *b* must be fixed for each level (but may be different for different levels) and the radix of the digits must be the same for all levels.

graphic character a character that has a visual representation normally handwritten, printed, or displayed.

(level) separator weight

a (non-zero) collation weight smaller (when regarded as an integer) than all weights used in collation elements at the preceding level, and with the same number of digits as used for the weights in the preceding level. A level separator weight is inserted by the reference method between each collation subkey.

ordering a process in which a set of strings are assigned a lexicographic order

symbolic weight name bound to a weight. Each symbolic weight is defined for a particular level.

symbolic collation item

a name bound to a non-empty character string. The name may be used in specifying collation items.

9.2 Table well-formedness (major)

1. Currently, each collation element that has a non-empty string of weights at level *i* also has a non-empty string of weights at level *i*+1 (The empty string of (symbolic) weights is called IGNORE in the balloted table). **This rule seems to be of no purpose.** Instead the well-formedness rules expressed in N639, and as comments in N641, should apply. These allow, or rather mandate, that level 2 items, combining accents mostly, have empty weight strings also at level 3 and 4.

2. In N641 **all modifier weights at levels 2 and 3 are heavier than any base weight at that level. This is in order to avoid edge case anomalies that will result if this is not followed**. In order to implement a check on this criterion, it facilitates if base and **modifier weights are declared as such** for each level. The current POSIX based syntax does not allow for that, but N639 does.

9.3 Key construction description in main text (major)

1. The key construction in the main text loosely refers to computing the 'numeric key', **but does not explain in sufficient detail how that numeric key is formed**. Some text is given in the above definitions, but this may need to be moved and/or expanded.

2. **Please delete section 6.2.2.2.** The main text (in section 6.2.2.2) suggests that level 4 (or in general the last level) should be treated differently from the other levels. This is both unnecessary and confusing, and the net effect (or, preferably, better!) should be produced by other means. Make a normative change of level 4 in the template table (see below, point 8, and level 4 as given in document N641) and the addition of an informative annex on key reduction (see document N642).

3. **N642 is a suggested annex** giving detail for two alternative methods to reduce the length of a subkey, without changing the ordering of strings as given by the collation keys as computed by the reference method. They are similar in spirit and internal key structure to what current section 6.2.2.2 would produce, but does correct a number of details. We strongly suggest instating into this standard this informative annex as part of the replacement of flawed section 6.2.2.2.

9.4 Table format (major)

Though there is no formal link from 14641 to 14652, there are still strong formal and informal links from (CD of) 14652 to 14651. Though we hope that 14652 will be very substantially revised

before turning into a standard, the existing link will taint the interpretation of the current table in 14651. Since these interpretations are greatly dissimilar, it would be highly preferable to use a table format in 14651 that **cannot be directly referenced** by (current) 14652, nor by the POSIX standards.

In order not to invent a completely new syntax for this, we suggest **basing the new table format on XML** (or SGML). At the same time one can address some of the **shortcomings** of the current table format (like that symbolic weights are not associated with a particular level, that well-formedness criteria are not enforceable at the syntactic level, that the 'auto-weighting' of symbolic weights is not explained, nor eliminable).

Document N639 gives a draft XML DTD for such a new table format (this has been updated, and the updated version can be supplied by the Swedish delegate). Document N641 gives a draft XML data file for the template table (some modifications has been done to this to follow the updated DTD).

Changing the table format should not incur significant additional delay in passing 14651 as a standard, considering that major changes need be done to level 2, 3, and 4 of the data in the table, whatever the format.

9.5 Level 1 in table (major)

4. The US delegate has done some changes to level 1. Some additional changes for Indic scripts may be needed. Though the Swedish representative has no expertise in Indic scripts, Jeoren Hellingman has been asked to supply comments on this point, and has done so. These comments have been forwarded to the US delegate for change in the data table. (See also N641, where these changes have been done by moving the entries to the suggested order; note however, that the symbolic weights have not been corrected accordingly).

5. Some generation errors afflict the balloted table. They occur when a punctuation character is at the beginning of a decomposition, but there is a letter (or digit) thereafter (degrees-C, degrees-F, parenthesised numbers and letters). (This has been fixed in a later version of the table; it is *partially* fixed also in N641.)

6. (minor) While handling of numeric order collation of digit sequences is to be taken care of in the preparation stage in general, it seems unnecessary to do so for certain pre-isolated numbers, e.g. parenthesised numbers, and month numbers, where the parentheses (etc) and digits are made into a *single* character. Here it is known that there will be at most two digits, so we can easily have a "virtual" 0 as the initial digit for the one-digit isolated numbers (see N641, where this has been carried out).

7. Again for numbers, annex C gives informative details on how to handle numerical order collation of numerals in general, it also needs to have PLUS and MINUS as first level significant characters. We see no reason not to have it that way in the template, in order to avoid additional special tailorings to take care of this (see N641).

8. (unclear) It is unclear to this reviewer if the Greek lowercase letters with ypogrammeni (and the combining ypogrammeni) should include a level 1 weight corresponding to *iota.* But since the uppercasing of combining ypogrammeni is an uppercase *iota,* it seems plausible

that this combining character should have a level 1 weight the same as that for *iota* (with corresponding changes for the precomposed forms with ypogrammeni), and a level 2 weight of VRNT1.

9.6 Level 2 in table (major)

9. There is a systematic error in the balloted version of the template table at level 2 (missing BLANK; or as it is renamed BASE). This has been corrected in later versions of the table, including in N641).

10. (unclear) TONOS and AIGUT are mixed up at level 2 in the balloted table (tentatively fixed in N641).

11. (minor) The symbolic weights at level 2 for the accents are often in French, while the name of that accent in the 10646 character names are in English. It may better to take the accent name used in the character name as the level 2 symbolic weight of an accent.

12. All base weights at level 2 MUST be smaller than any level 2 modifier weight (as in N641).

13. (minor) More base weights at level 2: for tailorings it would be helpful to have a number of predeclared lighter and heavier variant weights at level 2 (see N641). This would relieve tailoring from declaring them.

14. Some ligatures have orthographic significance, like the oe ligature (tentative list below). Level 2-4 should consider these as single characters, even though they are collated as two letters at level 1. This makes the table more logical, since these letters are considered to be single letters, rather than two letters. (See COMB2 and COMB2L in N641.)

```
<ci1 mtc="0133" v1="L79D L7B1" v2="COMB2" v3="MIN"   cmt="LATIN SMALL LIGATURE IJ"/>
<ci1 mtc="0132" v1="L79D L7B1" v2="COMB2" v3="CAP"   cmt="LATIN CAPITAL LIGATURE IJ"/>
<ci1 mtc="0153" v1="L815 L72F" v2="COMB2" v3="MIN"   cmt="LATIN SMALL LIGATURE OE; COMB2L?"/>
<ci1 mtc="0152" v1="L815 L72F" v2="COMB2" v3="CAP"   cmt="LATIN CAPITAL LIGATURE OE; COMB2L?"/>
<ci1 mtc="00DF" v1="L86D L86D" v2="COMB2" v3="MIN"   cmt="LATIN SMALL LETTER SHARP S"/>
<ci1 mtc="FB4F" v1="LB21 LB2C" v2="COMB2" v3="MIN"   cmt="HEBREW LIGATURE ALEF LAMED"/>
<ci1 mtc="05F0" v1="LB26 LB26" v2="COMB2" v3="MIN"   cmt="HEBREW LIGATURE YIDDISH DOUBLE VAV"/>
<ci1 mtc="05F1" v1="LB26 LB2A" v2="COMB2" v3="MIN"   cmt="HEBREW LIGATURE YIDDISH VAV YOD"/>
<ci1 mtc="05F2" v1="LB2A LB2A" v2="COMB2" v3="MIN"   cmt="HEBREW LIGATURE YIDDISH DOUBLE YOD"/>
<ci1 mtc="FB1F" v1="LB2A LB2A" v2="COMB2 PATAH" v3="MIN"   cmt="HEBREW LIGATURE YIDDISH YOD YOD
PATAH"/>
<ci1 mtc="0950" v1="LBD0 LBBD" v2="COMB2" v3="MIN"   cmt="DEVANAGARI OM"/>
<ci1 mtc="0AD0" v1="LC90 LC81" v2="COMB2" v3="MIN"   cmt="GUJARATI OM"/>
```

9.7 Level 3 in table (major)

15. In the balloted version of the table, Arabic ligature characters wrongly get the same weights at levels 1-3 as sequences of shaped Arabic letters, *of the wrong shape*. This is fixed in N641.

16. In the balloted version of the table, single characters with two digits in a circle wrongly get the same weights at levels 1-3 as two circled digits with a circle each. This is fixed in N641.

17. For simplicity, squared ligatures should be treated in the same way as other ligatures. (See N641.)

18. In order to make tailoring to get capital letters before minuscule letters easier, it is preferable to have only two weights indicating capital and miniscule status at level 3. (See N641.)

19. (minor) in order to ease tailoring for such things as Danish "Aa" and Spanish "Ch", it would be helpful to predeclare a CAP-MIN weight (see N641).

20. (minor) The NOBREAK and VERTICAL weights are not used, since they apply only to punctuation, which only have a level 4 weight anyway. These two weights may be deleted.

21. The balloted version of the table has only one weight for FONT, whereas there are sometimes **multiple font variations of the same character**. To remedy that N641 uses several different 'FONT' weights (ITALIC, SCRIPT, BLACK_LETTER, BOLD, DOUBLE_STRUCK, SANS_SERIF). This should be done also for the final version of the template table.

22. In order not to get a large number of possible combinations weights for **level 3**, N641 uses an approach similar to that used on level 2: **base weight and a sequence of modifier weights**.

23. In the balloted version of the table, some of the **square ligatures get the wrong level 1-3 weights, where Katakana or punctuation occurs** in the expansion of the square ligature. This is fixed in N641, and should be likewise fixed in the final version of the template table.

9.8 Level 4 in table (major)

24. C0 and C1 control characters (except tab/nl/cr) should be **ignored at all levels**; they should NOT affect even level 4. Similarly for BiDi control characters.

25. Currently level 4 consist of the 10646 character code (or a string of such). This leads to very strange behaviour if used right off. E.g. "it's" and "its" get ordered in the given order if the apostrophe is the ASCII one (a vertical glyph with mixed usage), but if one uses 02BC (modifier letter apostrophe, preferred character for this usage, the order becomes "its" followed by "it's". Former section 6.2.2.2 tried to fix this with a hack (including some edge case anomalies), but it is much preferable to use a proper solution: **give all letters and digits a level 4 weight called PLAIN that is heavier than all level 4 weights for symbols and punctuation**. Then we get a consistent and explainable order, also when punctuation is involved.

26. Weights of symbols/punctuation should **NOT be their 10646 code point**. Indeed, the "Canadian specials" hack in the balloted table indicate that a code point weight approach is unacceptable. All of the symbols and punctuation (that is ignored at levels 1-3) should have a level 4 weight such that they are grouped fairly logically together, which may give the "Canadian specials" weights such that their ordering is conforming with the Canadian standard, **but still groups similar symbols/punctuation together considering all of 10646.**

9.9 Example tailorings (minor)

There are two example tailorings of the template table given in an annex. However, neither of them is a “full” tailoring based on the template table. This makes them nearly useless as examples. N640 is a, in some sense, **“full” tailoring based on the template table (in XML format)**. (This has been updated to follow the updated DTD.)

In addition the two tailorings already present should be made “full”, and in particular be made to be based on the template, and it would also be helpful to have a tailoring for Japanese where the length marks are collated as a variant of the vowel each represent (depending on the preceding letter). (N641 has, in comments, so tailored 3 (of about 80*2) kana letters with length marks.)

9.10 Editorial comments

We have a number of editorial comments that can most easily be found by a difference-annotated version of the 14651 text. **(to be supplied)**

10 UK comments

```
The UK votes Yes with comments

 - UK comments GB(a)-GB(b) refer to editorial issues in sections 1-6;
 - UK comments GB(c) refers to a technical issue:
 - UK comments GB1-GB8 refer to details of the default table in section 7.

General: the UK notes that Michael Everson (NSAI, Ireland) had
volunteered to ISO/IEC JTC1/SC22/WG20 to undertake the task of improving
the English text, and hopes he will be able to continue that task.

UK comments GB(a)-GB(b) are intended to assist him in that task.

-----------------------------------------------------------------
```

10.1 GB(a) Editorial (mainly English problems)

```
-----------------------------------------------------------------

1. Scope para starting "Specific symbols" insert "for" after "except"

4.8 Second sentence replace "To a" with "A"

5. Second para second sentence delete "ever"

6.1.1 Note 1 replace "It is demonstrated" by "It can be demonstrated";
"not typically" by "typically not" and "required" by necessary"

6.2.1.2 Note para 4 replace "to code Arabic completely" with "the
complete coding of Arabic"


-----------------------------------------------------------------
```

10.2 GB(b) Editorial (mainly English problems, but without a recommended solution since the meaning of the original text isn't clear

---

5. Second para second sentence the usage of "all the coded graphic characters"

6.1.1 Note 1 "economy of means in the general case" isn't right

6.1.1 Note 2 "constitute very sensitive to interpret" isn't the correct English phrase, perhaps "are context sensitive data"?

6.2.1.1 "in a special way according to what is described in what follows"??

6.2.1.1 Note para 4 "presentation forms be coded in" is unclear

6.2.2.2 Level 4 "common to all scripts or the level not specifically belonging to any script"??

6.2.2.2 Level 4 para 3 It is not clear what the subject "these characters" actually is.

10.3 GB(c) Technical

BNF Syntax Rules should be those of the approved IS and this should be included in the References Clause 3

10.3.1 GB1. Cyrillic letters used in Old Church Slavonic and Macedonian:

Prefer altering position of character DZE, so it follows in the order ZHE, DZE, Z. Rationale:

If the default order uses that, it provides for old Church Slavonic (with a considerable literature, over many centuries) without any tailoring being required.

The current order involving DZE provides only for Macedonian, which was established as a literary language during WWII (BGN/PCGN information).

It is Macedonian which should use a tailoring here, as tailoring is very likely for Macedonian anyway, due to the interchange of glyphs G_acute and K_acute for DJE and TSHE respectively, but retaining the underlyiong Serbian order despite the glyph change.

BGN/PCGN also has the order Zhe, z, dze - a further variant ordering for Macedonian.

So the more stable Old Church Slavonic order should be adopted as the default order.

10.3.2 GB2. Greek

<U0342 IGNORE;<PERIS;<MIN;<U0342 % COMBINING GREEK PERISPOMENI should be filed following <U0303 IGNORE;<TILDE;<MIN;<U0303 % COMBINING TILDE
The tone mark PERISPOMENI is mis-ordered on most occasions in both ISO/IEC FCD 14651 and the Unicode Ordering Algorithm. It should follow other tone marks, not breathing marks.

Here is an example.

```
<U1FBD IGNORE;IGNORE;IGNORE;<U1FBD % GREEK KORONIS
<U1FBF IGNORE;IGNORE;IGNORE;<U1FBF % GREEK PSILI
<U1FC0 IGNORE;IGNORE;IGNORE;<U1FC0 % GREEK PERISPOMENI
<U1FC1 IGNORE;IGNORE;IGNORE;<U1FC1 % GREEK DIALYTIKA AND PERISPOMENI
<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
<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
<U1FED IGNORE;IGNORE;IGNORE;<U1FED % GREEK DIALYTIKA AND VARIA
<U1FEE IGNORE;IGNORE;IGNORE;<U1FEE % GREEK DIALYTIKA AND OXIA
<U1FEF IGNORE;IGNORE;IGNORE;<U1FEF % GREEK VARIA
<U1FFD IGNORE;IGNORE;IGNORE;<U1FFD % GREEK OXIA
<U1FFE IGNORE;IGNORE;IGNORE;<U1FFE % GREEK DASIA
```

ELOT, in correspondence with the European Ordering Rules Project Team, states that letters with tones but no breathing marks should follow letters with breathing marks.

The ISO/IEC FCD 14651 should provide a justification for the current ordering in a comment, or even alter the ordering.

10.3.3 GB3. Naming conventions

Naming conventions in tables in ISO/IEC FCD 14651, the Unicode Ordering Algorithm SYMDUMP2.TXT and the European Ordering Rules all vary.

The European Ordering Rules are most consistent, fullest, and recogniseably English language in description.

For the English language version of ISO/IEC FCD 14651, the full form used in the European Ordering Rules should be used, rather than any abbreviated French language conventions, for ease of use by those using the tables.

EOR: - uses same naming conventions as in ISO/IEC 10646

```
<U01DF <a;"<DIAERESIS<MACRON";<SMALL;<U01DF % LATIN SMALL
                        LETTER A WITH DIAERESIS AND MACRON
```

ISO/IEC FCD 14651: - uses differnt naming conventions from ISO/IEC 10646

```
<U01DF <S6CD;"<TREMA<MACRO";<MIN;<U01DF % LATIN SMALL
                        LETTER A WITH DIAERESIS AND MACRON
```

Abbreviations are fine, but they should use abbreviations of the first few letters of the name element in ISO/IEC 10646. There should be no ambiguity in doing this, if it is felt necessary for the columns to allign.

Column allignment is not required for a machine readable table, and column allignment seems an unnecessary refinement.

10.3.4 GB4. Inconsistencies

The spacing and non-spacing versions of the same characters (tilde, etc) are filed differently, rather than interfiling. A rationale for this is not given. Ideally they should be the same for consistency.

10.3.5 GB5. Ordering of SPACE

Regarding ordering of SPACE, in the former versions of ISO/IEC FCD 14651, a toggle was forced, so that the user had to decide one way or the other, by decommenting the relevant field. The draft standard had additional comment fields to assist the user in this.

Now, however, SPACE is treated completely differently in the default tables of ISO/IEC FCD 14651 and the Unicode Ordering Algorithm, but without any comments in either case.

In the former, SPACE is ignored in filing: in the latter it is a blank character. The latter reflects general practice in nearly all existing IT systems, at operating system level and in many applications: that is what should be followed in ISO/IEC FCD 14651, i.e. ISO/IEC FCD 14651 should follow Unicode Ordering Algorithm practice in SYMDUMP2.TXT.

If there are differences between these two standards that are reckoned to be a profile one of the other, there should be a justification, in comment fields, or appropriate text in the body of the standard.

10.3.6 GB6. Conventions for describing fields within tables

Given that the Unicode Ordering Algorithm, ISO/IEC FCD 14651 and the European Ordering Rules Project Team are supposed to be harmonised, some conventiuons are unexplaned [1] and there are unnecessary and unexplained differences between them [2]:

```
[14651]   <U0041 <S6CD;<BLANK;<CAP; <U0041 % LATIN CAPITAL LETTER A
[Unicode] <U0041 <S6CD;<BLANK;<CAP; <@0041 % LATIN CAPITAL LETTER A
[EOR]     <U0041 <a;<BLANK;<CAPITAL;<U0041 % LATIN CAPITAL LETTER A
                  [1] (weight)          [2]
```

These should be explained in each case, somewhere in each standard. The EOR weight is different, rather like the previous version of ISO/IEC FCD 14651.

In ISO/IEC FCD 14651, the records in the default table use <COMPAT etc: compatibility characters are defined in Unicode but not in ISO/IEC FCD 14651 or in ISO/IEC 10646:

Please add appropriate definitions/descriptions here.

10.3.7 GB7. Possible errors of ordering in the default table

This apostrophe should go with other apostrophes:
<U055A <S27B;<BLANK;<MIN;<@055A % ARMENIAN APOSTROPHE

There are possible inconsistencies in that some letter-like characters are filed anong the letters, others are filed among symbols in a separate sequence, as below (the <S number show that these are all filed as symbols in that <S order: other characters inserted on the left indicate other characters that they might file among, for consistency:

```
        <U2108 <S2EF;<BLANK;<MIN;<@2108 % SCRUPLE
 L B    <U2114 <S2F0;<BLANK;<MIN;<@2114 % L B BAR SYMBOL
 P      <U2117 <S2F1;<BLANK;<MIN;<@2117 % SOUND RECORDING COPYRIGHT
        <U211E <S2F2;<BLANK;<MIN;<@211E % PRESCRIPTION TAKE
 R      <U211F <S2F3;<BLANK;<MIN;<@211F % RESPONSE
 V      <U2123 <S2F4;<BLANK;<MIN;<@2123 % VERSICLE
 OZ     <U2125 <S2F5;<BLANK;<MIN;<@2125 % OUNCE SIGN
[Omega] <U2127 <S2F6;<BLANK;<MIN;<@2127 % INVERTED OHM SIGN
[iota]  <U2129 <S2F7;<BLANK;<MIN;<@2129 % TURNED GREEK SMALL LETTER IOTA
 e      <U212E <S2F8;<BLANK;<MIN;<@212E % ESTIMATED SYMBOL
 f      <U2132 <S2F9;<BLANK;<MIN;<@2132 % TURNED CAPITAL F
```

Some of these Latin numbers should go with other alphabetic filing, as indeed other ones do in the main Latin (etc) sequence, e.g.

```
 CD     <U2180 <S2FA;<BLANK;<MIN;<@2180 % ROMAN NUMERAL ONE THOUSAND C D
        <U2181 <S2FB;<BLANK;<MIN;<@2181 % ROMAN NUMERAL FIVE THOUSAND
```

```
        <U2182 <S2FC;<BLANK;<MIN;<@2182 % ROMAN NUMERAL TEN THOUSAND

Here are Latin numerals which are mostly in a more predictable filing
sequence:

        <U217D <S6F9;<BLANK;<COMPAT;<@217D % SMALL ROMAN NUMERAL ONE
        HUNDRED
        <U216E <S705;<BLANK;<COMPATCAP;<@216E % ROMAN NUMERAL FIVE
        HUNDRED

 vi     <U2175~<S8C7<S79B";"<BLANK<BLANK";"<COMPAT<COMPAT";"<0076<0069" %
        SMALL ROMAN NUMERAL SIX

<U2165~<S8C7<S79B";"<BLANK<BLANK";"<COMPATCAP<COMPATCAP";"<0056<0049" %
ROMAN NUMERAL SIX
 vii
<U2176~<S8C7<S79B<S79B";"<BLANK<BLANK<BLANK";"<COMPAT<COMPAT<COMPAT";"<0076<
0069<0069" % SMALL ROMAN NUMERAL SEVEN

<U2166~<S8C7<S79B<S79B";"<BLANK<BLANK<BLANK";"<COMPATCAP<COMPATCAP<COMPATCAP
";"<0056<0049<0049" % ROMAN NUMERAL SEVEN
 viii
<U2177~<S8C7<S79B<S79B<S79B";"<BLANK<BLANK<BLANK<BLANK";"<COMPAT<COMPAT<COMP
AT<COMPAT";"<0076<0069<0069<0069" % SMALL ROMAN NUMERAL EIGHT

<U2167~<S8C7<S79B<S79B<S79B";"<BLANK<BLANK<BLANK<BLANK";"<COMPATCAP<COMPATCA
P<COMPATCAP<COMPATCAP";"<0056<0049<0049<0049" % ROMAN NUMERAL EIGHT
 xi     <U217A~<S8DB<S79B";"<BLANK<BLANK";"<COMPAT<COMPAT";"<0078<0069" %
SMALL ROMAN NUMERAL ELEVEN

<U216A~<S8DB<S79B";"<BLANK<BLANK";"<COMPATCAP<COMPATCAP";"<0058<0049" %
ROMAN NUMERAL ELEVEN
 xii
<U217B~<S8DB<S79B<S79B";"<BLANK<BLANK<BLANK";"<COMPAT<COMPAT<COMPAT";"<0078<
0069<0069" % SMALL ROMAN NUMERAL TWELVE

<U216B~<S8DB<S79B<S79B";"<BLANK<BLANK<BLANK";"<COMPATCAP<COMPATCAP<COMPATCAP
";"<0058<0049<0049" % ROMAN NUMERAL TWELVE


This character should file with 6, not with b:
        <U0185 <S6F5;<BLANK;<BIN;<@0185 % LATIN SMALL LETTER TONE SIX
        <U0184 <S6F5;<BLANK;<CAP;<@0184 % LATIN CAPITAL LETTER TONE SIX

This character should file with 2, not with s:
        <U01A8 <S877;<BLANK;<MIN;<@01A8 % LATIN SMALL LETTER TONE TWO
        <U01A7 <S877;<BLANK;<CAP;<@01A7 % LATIN CAPITAL LETTER TONE TWO

This character should file with 5, not well after Z, between WYNN &
GLOTTAL STOP:

        <U01BD <S917;<BLANK;<MIN;<@01BD % LATIN SMALL LETTER TONE FIVE
        <U01BC <S917;<BLANK;<CAP;<@01BC % LATIN CAPITAL LETTER TONE FIVE


----------------------------------------------------------------
```

10.3.8 GB8. Korean

At the end of the default table, there is information about ordering Han (Chinese) and Hangul (Korean) characters: this comment reproduces the end of the table, and inserts to mark UK comments.

```
<U4E00..<U9FA5 <@4E00..<@9FA5;<BLANK;<MIN;<@4E00..<@9FA5 % Han

    This only gives details about ordering of han characters
        using radical/stroke sequences. There is no information
        given, even in comments, about ordering in the order of Latin
        alphabet equivalents (as in pinyin in Chinese), or as kana
        equivalents (as in Japanese), or as hangul equivalents (as in
        Korean) although each is very common in East Asia.

        By comparison there is some description below about ordering
        hangul syllables.

% <UAC00..<UD7A3 <@AC00..<@D7A3;<BLANK;<MIN;<@AC00..<@D7A3 % Hangul
% Weights for Hangul syllables are built by equivalences to the jamo
  weights.
% A Hangul tailoring for a system which does not use combining jamos
%   may choose to simply weight the Hangul syllables directly as shown
    above.


    However, this does not state explicitly whether the weights
        which are built by equivalences to the jamo weights should
        follow the Hangul jamo in row 11 onwards, or in row 31
        onwards.


% order_end

% END LC_COLLATE

% Decomment the line above to create a 14652-style
%   LC_COLLATE definition.
```

10.3.9 GB9. Script-by-script ordering in ISO/IEC FCD 14651

In the earlier disposition of comments in mid 1998, not all UK comments about providing an order for scripts in ISO/IEC FCD 14651 were taken into account.

Leaving this to tailoring, as indicated in comment GB18 in the Disposition of comments, will not be satisfatory as it is anticipated that many applications and implementations will rely on the default table of ISO/IEC FCD 14651: GB 18 said:

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.

The UK considers that a reasonably predictable order should be implicit in the ISO/IEC FCD 14651 defalttable, and that leaving script order entirely to tailoring is insufficient.

This extended comment (ref. GB9) proposes a rationale, describes such a table, based on other standardisation work in ISO/TC46/SC2, makes a comparison with UCS, and appends the UK's earlier concern in earlier comments.

Such ordering was implicit in earlier drafts of ISO/IEC FCD 14651, as noted in the earlier comments by the UK (see UK comments, section 3.A.2. Order of scripts) but is no longer specified in any single area of ISO/IEC FCD 14651.

10.3.10 GB9.1. Rationale.

- As there is currently no national recognised standard or convention which says where users can expect to find specific scripts in a multiscript listing (increasingly likely as UCS gets adopted and global business increases), and

- As the default order in ISO/IEC FCD 14651 is likely to be taken as _the_ prefered order, as there is no other available guide,

the order in ISO/IEC FCD 14651 should be rational and predictable to users, without reference to other standards, such as UCS, with which many users may be unfamiliar, and to which they may not have access.

The order should also account for the likely repertoire of ISO/IEC 10646-1: 2nd edition and Unicode version 3.0, which incorporates amendments to ISO/IEC 10646, which are likely to be confirmed at the March 1999 meeting of ISO/IEC JTC1/SC2/WG2 in Fukuoka, Japan.

10.3.11 GB9.2. Proposed script order in ISO NP 15921: Generalized conversion methods, suggested for adoption in ISO/IEC FCD 14651

The order below gives (a) priority to scripts used in official languages, broadly similar to the order in UCS (ISO/IEC 10646 and Unicode). There is a broad West through East order, and within that (where relevant) a broadly North through South order, with (b) non-official scripts added at the end of that sequence, in a similar West through East order.

This order is also being adopted in the early drafts of ISO NP 15921: Generalized conversion methods, being developed in ISO/TC46/SC2/WG8:

Transliteration and Computers.

(a) Scripts used in official languages (at country level) *

```
1:   Americas/Europe:        Latin
2-5: Europe:                 Greek, Cyrillic, Georgian, Armenian;
6:   Near East:              Hebrew;
7:   West Asia/North Africa: Arabic;
8:   Northeast Africa:       Ethiopic;
9:   South Asia:             Devanagari,
a-d       "                  Bengali, Gurmukhi, Gujarati, Oriya;
e-h:      "                  Tamil, Telugu, Kannada, Malayalam,
i:        "                  Sinhala;
j:        "                  Thaana;
k-n: Southeast Asia:         Thai, Lao, Myanmar (Burmese), Khmer;
o-p: Inner Asia:             Tibetan, Mongolian;
q-s: East Asia:              Korean, Japanese, Chinese.
```

(b) Scripts used in official languages below country level *
by minorities within countries, and in religious/historical texts

```
t-u: Americas:               Cherokee, Canadian Aboriginal Syllabics;
v-x: Europe:                 Ogham, Runic, Glagolitic;
y:   Near East:              Syriac;
z:   East Asia:              Yi (Southwest China),
```

Notes:

* Country status is taken at the year 1999, and based on the list of countries recognised by the United Nations at that date.

11 USA comments

March 12, 1999

Ballot document: SC22 N2844 (SC22/WG20 N619)

The US votes NO on 14651.
The vote would be changed to YES if the following changes were made.

The main goals of the UTC and US position are to ensure that

(1) Major collation implementations (POSIX, Java, Sybase, etc.) that currently produce satisfactory international orderings for Unicode can be conformant to ISO 14651, and

(2) The proposed Unicode Standard Collation Algorithm (UCA), which pays close attention to the special requirements of Unicode conformance, can be conformant to 14651. The specification of the UCA can be found at http://www.unicode.org/unicode/reports/tr10/.

The main changes that the UTC requires of 14651 can be summarized as:

11.1 A. Levels

Conformant 14651 implementations must not be required to support more than the first 3 levels. (They are free to support more than 3, but not required to.) It is not at all clear from the current conformance clause how many levels a conformant implementation must support. To address this concern, make the following changes:

a. On page 5, 6.2.1.1 Assumptions. The statement that "The number of levels can be extended in the tailoring phase by the end-user." should be modified to: "The number of levels can be extended or reduced in the tailoring phase." (Note also removal of the red-herring use of the term "end-user".)

b. Add the following language to 6.2.1.1
"Conformant implementations of 14651 must support at least three levels. They may support more levels, but they are not required to for conformance. In the absence of such support, fourth and higher level information can be ignored."

11.2 B. Position

Conformant 14651 implementations must not be required to support the position designator. (They are free to support the position designator, but not required to.) In addition, the text following the paragraph in 6.2.2.2 starting with "Generally" is informative, not normative, and does not belong in this section.

To address these requirements, make the following changes:

On page 5, 6.2.1.1 Assumptions. The sentence starting "The user shall take care that,..." should be omitted. It is very strange in that it normatively requires a user to "take care that...", but what they must take care is then expressed as a conditional with a protasis expressed as "so that the last level may processed [sic]". The whole sentence is an incomprehensible admonition as it stands. What we want is a clear statement that the standard does not *require* special processing at the last level, but does *allow* it (see below).

In 6.2.1.2, change "A specific property" to "An optional property"

In the first paragraph of 6.2.2.2, change the condition to read:
"If there is an order_start entry that does not use the position value at level m of a block, or if there is no order_start entry, then the formation of subkey level m is done in exactly the same way as the above-defined formation.
Otherwise..."

Add the following language to 6.2.2.2 after the paragraph starting "During".

"Conformant implementations of 14651 are not required to support the position value. They may support this value, but are not required to for conformance. In the absence of such support, the position value is ignored."

d. Split 6.2.2.2 into two parts. The new part 6.2.2.3 would begin on the bottom of page 6, just above the paragraph starting "Generally," and should be entitled: "General interpretation of each level in the Common Template Table".

e. In the new 6.2.2.3, delete all but the first sentence in the paragraph labeled "Level 4". That would disconnect the interpretation of Level 4 from whether or not keys are constructed for Level 4 using the position mechanism.

f. Move the paragraph following the "Level 4" paragraph (starting "In the table, this behavior is...") up into 6.2.2.2 after the note about forward and backward scanning.

g. Move the new section 6.2.2.3 into some other place in the standard. It is informative, and should not be part of the normative clause 6.

11.3 C. Backward

Conformant 14651 implementations must not be required to support the backward designator at any level but level 2. Moreover, conformant 14651 implementations are not required to have anything but a **global** backwards switch (e.g. that all weights at a particular level are either uniformly forward or backward). (They are free to support the multiple levels of backwards, and fine-grained directionality [on a per character basis], but not required to.) To address this requirement, add the following language to 6.2.1.2:

"Conformant implementations of 14651 are not required to support the 'backward' scanning direction at any level but level 2. In the absence of such support, the scanning direction is treated as if it were 'forward' at every level but level 2.

"Conformant implementations of 14651 are also not required to support different scanning directions for different blocks. In the absence of such support, if any block has a backward scanning direction for any level, then all blocks are considered to have that scanning direction at that level."

To the note at the end of 6.2.1.2 starting "In ISO/IEC 10646-1, Arabic…, add the following text: "However, the Unicode Standard does proscribe the logical order of all characters, including Arabic and Hebrew. Implementations conforming to the Unicode standard will not use the backward scanning property."

[Note: the current description of per-block backward and forwards support in 14651 does not serve the goal it was designed for. Since languages and scripts share a great many characters in common, a choice of either forward or backward will cause those common characters to disrupt the order within text of the other direction. For example, suppose Greek is ordered forwards, and French backwards. If digits, for example, are forward then they disrupt the French accents. If they are backward, then they will disrupt the Greek accents.

Even going to a forward, backward, neutral model, as in UCA Version 2 will not work. No matter which heuristics are used to assign the direction of the neutrals, sometimes the choice will be incorrect.

Mixing blocks of different direction is not well supported in industry practice. Most implementations of POSIX do not support it, nor does Java. Forcing these implementations to revise without solid justification is unwarranted. However, as long as implementations are not forced to implement mixed scanning directions, the current language can remain.]

11.4 D. Unicode conformance

ISO 14651 must permit a conformant implementation to do the following. (These are required for conformance to the Unicode Standard.)

D.1. Treat canonical equivalent strings as precisely equal in ordering.
D.2. Perform Thai/Lao-style character reversal (see UCA Step 1).
D.3. Exclude irrelevant combining marks when looking up matches for contracting characters (see UCA Step 2).
D.4. Exclude unsupported characters from a collation ordering, or cause them to be sorted in Unicode code point order.

Items D.1 through D.3 are probably covered by section 6.1. However, to ensure that they are, these three items must be added in Notes as examples of conformant implementations, with the following language:

"Note: to allow conformance to the Unicode Standard, conformant implementations may
a. Treat canonical equivalent strings as precisely equal in ordering.
b. Perform Thai/Lao-style character reversal.
c. Exclude irrelevant combining marks when looking up matches for contracting characters.
For more information, see Unicode Technical Report #10."

D.4 is commonly implemented as UNDEFINED in POSIX and other standards. It must be included so that implementations working in low-memory environments that do not need the full default collation rules can use a small subset, and have all other Unicode characters sorted by code order. To fix this problem, make the following changes:

In 6.3.1 rule 23, add the text " | UNDEFINED" to the end of the line.

At the end of 6.2.2.1, add the text:

"If there are no tokens corresponding to a character of the input string, then the character is undefined. Undefined characters are sorted with respect to defined characters as if they were at the position UNDEFINED in the Template Table. (If there is no UNDEFINED token in the table, then the table is interpreted as if there were one at the very end.) The ordering of undefined characters with respect to other undefined characters is not specified by this standard.

Note: there are two common treatments of UNDEFINED characters. The first is to sort among them as if their level-one weight differences were based upon their UCS character code. The second is to sort them as if they all had the same level-one weight, and their second-level weights were the same as their UCS character codes."

11.5 F. Stability:

The data for both UCA and 14651 must be updated to the level of symdump-2.1.9.txt on the SC22/WG20 server (incorporating all of the individual changes that the US would be asking for).

No further changes to other parts of 14651 that would substantially affect the current major collation implementations are acceptable to the UTC or the US national body. In particular, the default data for levels 1, 2, and 3 used by 14651 must be consistent with the UCA data (though perhaps not in the same format). The data was synchronized; this must not diverge due to ballot comments.

11.6 G. Specific Technical Comments

Section 6.3.3. is not well defined. Rule I2 (reorder_after) must state what the relationship is between the table lines (X) between the entries and the tailored line containing the symbol defintion (S). That is, suppose we have the following rules:

<UA> <A1>;<A2>;<A3>;<A4>
<UB> <B1>;<B2>;<B3>;<B4>
…
<UX> <X1>;<X2>;<X3>;<X4>
<UY> <Y1>;<Y2>;<Y3>;<Y4>

We want to tailor that table by adding a reordering rule:

reorder-after <UX>
<UX> <X1>;<X2>;<X3>;<X4>
<UY> <Y1>;<Y2>;<Y3>;<Y4>
reorder-end

What does the normalized output (I4) look like? According to the rules, it could be:

<UA> <A1>;<A2>;<A3>;<A4>
<UX> <A1+1>;<MIN2>;<MIN3>;<MIN4>
<UY> <Y1>;<Y2>;<Y3>;<Y4>
<UB> …

Or it could be

<UA> <A1>;<A2>;<A3>;<A4>
<UX> <A1>;<A2>;<A3>;<A4>+1
<UY> <Y1>;<Y2>;<Y3>;<Y4>
<UB> …

Both of these operations might be required for a tailoring, but the rules I1 and I2 do not distinguish between them. Moreover, the rules do not say what is the effect on UB--does it have the same level distinction with the last of the new line(s) that it used to with UA?

To address this problem, the following (or equivalent) change must be made.

6.3.1, rule 32. Change to:
reorder_after_entry := 'reorder-after ' target_symbol ' at level ' digit+

6.3.3 rule I2. Add:
" The reorder entry effectively inserts lines X through Y between existing lines A and B, producing the new ordering <A, X…Y, B>. The level of the reorder-after statement determines the level of the differences between A and X. The level of the difference between Y and B is the stronger of the old difference level between A and B and the new difference level between A and X. For example, suppose we have the following lines (where B1 != A1):

<UA> <A1>;<A2>;<A3>;<A4>
<UB> <B1>;<B2>;<B3>;<B4>
…
reorder-after <UX> at level 2
<UX> <X1>;<X2>;<X3>;<X4>
<UY> <Y1>;<Y2>;<Y3>;<Y4>
reorder-end

will produce the normalized result equivalent to:

<UA> <A1>;<A2>;<A3>;<A4>
<UX> <A1>;<A2>+2;<MIN3>;<MIN4>
<UY> <Y1>;<Y2>;<Y3>;<Y4>
<UB> <Y1>+1; <MIN2>;<MIN3>;<MIN4>"

It must be clearly stated that a reorder-entry also *removes* the lines from where they used to be.

In addition, the following text must be added at the end.
"The reorder-entries must be processed in order during normalization, otherwise incorrect results will be obtained."

I3 also unclear in that it doesn't discuss changing the actual numerical values of the weights. Yet the assignment of numerical values to weights doesn't occur until I5. If the assignment is not done in the reordering, then the subsequent assignment of weights would defeat the purpose of the reordering. This must be clarified.

11.7 H.

Given their importance in the development of this standard, and the fact that the vast majority of 10646 implementations are in fact Unicode implementations, the Unicode Standard must be referenced in Section 3, and Unicode 2.0, TR #8, and DTR #10 must be referenced in the Bibliography.

11.8 EDITORIAL

11.8.1 A.

The BNF rules in 6.3.1 should be supplemented by a textual description of the format. The well-formedness conditions can be interleaved with the textual description for clarity.

11.8.2 B.

Examples must be added to 6.3.3 to make the requirements clear, as above.

11.8.3 C.

Change the explanation in 6.3.1 BNF Syntax Rules to use more standard notation (e.g. Aho and Ullman):

"<…> refers to terms not defined in this BNF syntax, and assume general English usage.
'…' refers to literal characters
(…) used for grouping
X Y matches the token sequence X followed by Y
X | Y matches either X or Y tokens
X* matches zero or more repetitions of X
X+ matches one or more repetitions of X
{X} matches one or more repetitions of X "

Replace the use of "{}" by "<>", and "()" by "{}" in the BNF rules

[Note: in standards documents such as XML, X? is used instead of {X}]

11.8.4 D.

Certain word-smithing needs to be done for clarity and accuracy. Take the introduction alone:

- Sentence #2 is untrue--that is not the only purpose; others are mentioned below.
- #4 is has an incorrect reference "English" is not a "past approach".
- The last sentence of para#2 is incorrect--one does not "achieve challenges"; one might "overcome them", if that is what is meant.
- "result discrepancies" must be changed to "discrepancies in results"
- "excellent" sounds like blowing our own horn too much.

A full list would take too long to compile -- marked-up copies will be brought to the Pennsylvania meeting.

11.8.5 E. Section 2.

The requirements imposed by the second paragraph are unclear.

11.8.6 F. Section 4.

The word "token" should be replaced throughout the document by "weight", unless the definition is in error.

Collating symbol and collating element should be change to collation symbol and collation element.

The difference between ordering key and collation element is not clear from the definitions.

"preparation": speaking of the actual source strings being modified here and in 6.1.1 is worrysome--it is copies of the source strings that are modified, if anything.

6.2.1.1 "matrix of n lines. N is the number of characters in the repertoire used."
This would exclude multiple characters sorting as 1. Also, "matrix" is unclear; what is meant? It is also not really a "transformation table". What it is is a mapping table from character sequences to collation elements.

6.3.4. The first paragraph can be simplified considerably to:

Two collation weighting tables are said to be equivalent if any comparison of strings using those tables results in the same ordering.

11.9 More editorial comments

11.9.1 Introduction, page iv, first paragraph

a) The meaning of the word "**universal**" is ambiguous here. It perhaps implies that there may be other non-Universal properties which are not retained during tailoring. Does this paragraph intend to indicate that all scripts have these properties, or does it mean that the particular values of these properties as defined for each script is common to all users of the Common Template Table, if they are not tailored? One can presume the latter, but it should be more clearly stated. A suggestion might be to change "retaining universal properties for other scripts" to "retaining properties already defined for other scripts."
b) This paragraph seems to be saying that the purpose of this standard is to improve on collation algorithms based only on binary coded character values. If this refers to the use of the binary coded values without associating a weight to those values, then the next comment about English, with uppercase characters only and no punctuation, being an exception, makes sense. However, it is a rather weak statement, given that even the simplest collation algorithms generally apply some weighting scheme. A suggestion might be to simply delete the remainder of the paragraph beginning with "The purpose of such a mechanism…"

11.9.2 Introduction, page iv, second paragraph

In the first sentence "this is one of the major flaws that affect portability…" it is not clear what "this" is referring to, or what is "flawed". A suggestion might be to combine the sentence with the parenthetical remark: "That different programs use different ordering specifications is a significant problem reducing portability between countries and between applications."

11.9.3 Section 1 Scope

In the first paragraph "A simple method of reference…" delete "of reference", as the method is for comparing not for referencing. It is understood that this standard is defining a method which can be a reference for international ordering.

In the last bullet in this section, delete the final 2 words "to order" in "A context-dependent ordering which would require complex transformation of data to order."

11.9.4 Section 2 Conformance

In the last sentence “and how the comparison method they use If different” the “I” in “if” should not be capitalized. There should be a comma after the word “use”.

11.9.5 Section 4 Definitions

4.6 delta- change “relatively” to “relative”

4.8 graphic character- change
“To a graphic character normally corresponds a glyph.” to
“A graphic character normally corresponds to a glyph.”

4.9 level- This definition is ambiguous as “depth” is not defined. The author should provide a more meaningful definition.

11.9.6 Section 5 Symbols and abbreviations

The last 2 sentences in the first paragraph can be worded more grammatically correct and “covered” can be clarified by changing
“What is being referenced is a graphic character, independently of its coding, and any character set whose subrepertoire is taken into account in ISO/IEC 10646-1 is covered in this way.” to
“This is a way to reference a graphic character, independent of its coding. Any character set whose subrepertoire is taken into account in ISO/IEC 10646-1, is included in this specification by this nomenclature.”

11.9.7 Section 6.1.1 Preparation of character strings prior to comparison

In the first paragraph, will the reference to telephone-book ordering be universally understood, or should the specific problem referred to in this example be brought out?

In the second paragraph, the words “but not both” should be added to the phrase “An application conformant to this internaitonal standard shall at the minimum prepare the string so that sequences using either combining sequences or using precomposed characters…”

In Note 1 of this section, remove the extraneous “ a ” in “precomposed characters affected by a diacritics,”

The term “double-coding” may be unclear. The last sentence might be restated as follows for clarity:
“However, as it is not typically the case that precomposed and combining characters are both used, and therefore for reasons of table efficiency, it is not a requirement of the standard to always add the extra tokens that represent applying diacritics to precomposed characters.”

11.9.8 Section 6.2.2 Key composition

11.9.9 Section 6.2.2.1 Formation of subey level 1 through (m-1)

This section is very unclear and must be made more precise and would greatly benefit from an example. In particular, references to directionality are made with respect to string processing, levels and characters and is hard to understand. Stacking is described but unstacking is left to the reader’s imagination. In particular it is not clear when to unstack.

For example, in the second paragraph after the parenthetical remark, it states: “and the new direction is backward” it is not clear how many attributes of the algorithm are affected. The character has the property of being backward, this changes the direction of the current level i, and might be presumed to also affect the scanning direction of the input character string, which is described as initially forward in the first paragraph.

If we understand the proposed algorithm correctly, it would benefit the specification to state clearly:
1) That scanning of the input character string is always forward thru the logical sequence of the string.
2) That reaching a character with a backwards property changes the current direction of level i from forward to backward, and commences stacking of position and token.
3) That reaching a character with a forwards property when the current direction of level i is backwards, changes the level's direction to forwards and commences unstacking, with a description of what is involved in unstacking.

11.9.10 Section 6.2.2.2 Formation of subkey level m

The first sentence should change "uses" to "use".

The first paragraph begins with discussion of **order_start_entry** which is not yet introduced . This should be characterized and the subsequent reference to having or not having a position, expanded upon for clarity. The significance of using the table as-is versus changing it in accordance with frequent market practice should also be clarified and the alternative behaviors of the ordering described. An explanation of why the Common Template Table does not follow frequent market practice might also be offered.

In the second paragraph, the sentence "When the character is not assigned at level m in the table, it is ignored for the formation of subkey level m and no pair is concatenated." Might be better moved to the end of the paragraph, so the subsequent sentences cannot be perceived to be part of the condition "when the character is not assigned at level m".

In addition, this paragraph is the first indication that a character might not have entries for every table level. There should be some discussion of this and its impact on behavior of the ordering.

The first sentence in the description of level 4 states: "This level represents the level common to all scripts or the level not specifically belonging to any script." We do not understand what this means. How and why is this level different from the other levels?

In the last paragraph of this section, it is stated: "In the Common Template table, definitions of these characters for level 1 to 3…". We do not understand which characters are referred to by "these characters". Perhaps the author should state: "In the Common Template table, characters that are assigned values at level 4, are exclusively assigned to level 4, and are ignorable, and have no values assigned, at levels 1-3.

It might improve the readability and understandability of the specification, if the actual description of the Common Template table was moved out of this section to the later section on the Common Template table and if the information in level 4, about the formation of the level 4 or level m subkey, was included with the first 2 paragraphs of this section, describing the key formation.

11.9.11 Section 6.4 Declaration of a delta

In the second paragraph, conformance is described as declarable if a fixed table is used by the application. Can an application conform if it does not make use of a fixed table analagous to the Common Template table?

Also, the term "comparison table" is not defined. Presumably this is the name for the transformation table used with the comparison method and this should be stated or clarified. Also the word "relatively" should be "relative" in this instance.

In the first bullet, there is a reference to direction values being dependent on writing systems. Earlier, the specification pointed out that scanning direction is in fact independent of the direction of writing, so this may be confusing and misleading to readers.

In the first paragraph after the 4 bullets, the sentence beginning with “In cases where the applications has…” should be changed to “In cases where the applications have…”.

_____ end of SC22 N2911 _____