Approved Minutes – UTC #78 & NCITS Subgroup L2 # 175 Joint Meeting
San Jose, CA – December 1-4, 1998

Approved as amended, February 5, 1999

Chair Aliprand convened the joint meeting of the UTC and L2 (L2 Ad Hoc) on Tuesday, December 1, 1998.

Administrative Items

Call for Proxies

UTC Membership Roll Call -- See Attachment 1 for list of Attendees

PRESENT: Apple Computer, Inc.; Compaq Computer Corporation; IBM Corporation; Microsoft Corporation; NCR Corporation; Novell, Inc.; Oracle Corporation; The Research Libraries Group, Inc.; SAP AG; Sybase, Inc.; Unisys Corporation.; Xerox Corporation
(Total members represented: 12)
Quorum = 10

NOT PRESENT (at time of roll-call): Booz, Allen, Hamilton, Inc.; Hewlett-Packard Company; Justsystem Corporation; Mathema Software, GmbH; Reuters, Ltd.; Silicon Graphics, Inc.: Sun Microsystems, Inc.;
(Total not represented: 7)

Approval of the Minutes of the previous joint meeting and review of Action Items was deferred.

Consent Docket on WG2 Resolutions at Meeting #35

[Document L2/98-389]

Davis asked which items were previously accepted by the UTC and which are new changes. Aliprand and Whistler replied that there are no characters in this document which had been previously accepted.

The following amendments resulted from discussion:

- Deletion of Resolution M35.2 (no differences in Sinhala as accepted by UTC);
- Insertion of "except the SOFT SPACE" in UTC action re Resolution M35.12 (it was asserted that ZWSP serves same function as SOFT SPACE);
- Deletion of Resolution M35.17 (UTC seeks explanation of CJK Radical Supplement); and
- Editorial changes (correction of typographical errors in note to Resolution M35.6, and addition of L2 equivalent of WG2 documents).

Suignard explained use of the SOFT SPACE character:
In Thai and Khmer there are no spaces to separate words. Spaces separate sentences, and sometimes phrases. The soft space character is needed for justification of text in columns. Davis argued that zero width space was designed for this purpose, and the soft space is duplication. Suignard felt it dangerous to combine the two concepts. McGowan thought we need to clarify the semantic of zero width space being allowed to have spacing associated with it. Mansour said we need to insure the soft space won't break existing applications. Davis added that justification rules are always script dependent, e.g. Kashida in Arabic, and recommended going back to WG2 on this. He agreed to be editor for U.S. comments on this .

Moved by McGowan, seconded by Moore
[#78-M1] Motion: To accept the consent docket L2/98-389 as amended
Unanimous

Motion approved.

Action item 78-1 for Davis: Be editor for US comments on PDAM on Amendment 30, to propose use of ZWSP instead of proposed SOFT SPACE.

Action item for Aliprand: Prepare revised version of L2/98-389, incorporating amendments.

Prioritization of scripts

[Document L2/98-348]

Becker said there is no category for scripts that we are not looking at. McGowan said that the list contains only those scripts under study. Other scripts can be added. Mansour asked the case where something in category 4 is ahead of 2. McGowan said it would be moved.

Mittelstein said that Klingon causes a problem. It causes Unicode to not be taken seriously. Aliprand said that the UTC is on record as saying that invented scripts have lowest priority. Davis suggested highlighting modern scripts in the list, so that this can be an executive summary.

David reported in BiDi. He rolled the results of errata into the document on the web, plus comments from the bidi ad hoc group meeting in Redmond. Still needs more work to clarify, and there is some controversy over edge conditions. He suggested a separate meeting outside of this meeting, to report back to the UTC by February.

Action item 78-50 for Moore: Schedule an Ad Hoc meeting to discuss BiDi issues.

Newline (nl) handling

[Document L2/98-402]
Moore- IBM has an additional new line character NL in C1 space, that is not reflected in the doc. Moore had sent mail to Davis a while back.
Sargent asked: What is NLF? Ans: The newline function. Aliprand said it is defined in the document but should be clarified as an acronym for NewLine Function.
McGowan would like to see discussion of selection, e.g. when you select a line nl gets moved with text. If we defined how this works, it would help interoperability. For example, when highlighting a paragraph, do you get leading or trailing or both separators?
Davis: It is out of scope, but would be useful to note behavior as included with previous line.
Dürst: It is useful to clarify if the characters should be displayed or not
Davis: There are 3 types: plain text, marked up text, and "out of bounds" describing flow.
Davis will include some notes about how to treat nl in marked up text, such as html.

Whistler- With respect to interpreting chars, there is not a clear distinction between word processing and text. Word has the concept of paragraphs, as opposed to editors those that do not have the concept of paragraphs.
Davis- Autowrapping also is a distinction
Sargent- Microsoft Word maps ps to lf , maps crlf just to cr when exporting. It uses the current platform's convention- crlf on pc, mac just cr. (He will confirm this.)
Mittelstein- How to address nl at end of doc? Should it be stripped off, or not, if at end of file?
Davis will explain difference between terminators and separators. If the final nl is a separator , there is a null paragraph at the end of doc. Otherwise not.

Moved by Davis, seconded by Whistler
[#78-M2] Motion: To progress the proposed draft UTR #13, *Unicode Newline Guidelines*, to draft status after all amendments have been incorporated.
13 for; 0 against; 1 abstention (SAP)
Motion approved.

Action item 78-2 for Davis: Incorporate changes suggested to Proposed Draft UTR #13, *Unicode Newline Guidelines*, and post the revision as Draft UTR #13 on the web site.

Action item 78-3 for Aliprand: Put Draft UTR #13 on agenda for February meeting.

The document L2/98- 407 on line-break notes, is a compendium of mail on the subject of line breaking.
Dürst- The Newline doc and the line break are related and may confuse people.
Davis: Newline is a hard break, while line break is more about wrapping.
Davis suggested that the title of L2/98-407 be changed to wordwrapping or a similar term.

UTF8 EBCDIC

Moore offered to bring comments to Umamaheswaran, who was sick.

Moore- Uma sick with flu. Moore can bring comment back to Uma.
Davis- I suggest some restructuring. Move the algorithm guts to the front and the rest to appendix.
Mittelstein- concerned about having too many UTF-like standards.
Moore- IBM's plan is to use this internally, and not for interchange.
Moore action/note To add statement of purpose at the front. We agreed in last UTC, but it hasn't been reflected in the doc.
Dürst- IETF, W3C require support of UTF-8. This is not uft-8 and is confusing.
Moore- UTF8-EBCDIC was suggested name at last UTC.
Davis- I suggest table on page 2 of 31, should have a left column indicating ranges supported by those rows.
Honomichl- This report references "shortest string rule" and implies null can be more than one byte. All references to this should be removed.
Moore- OK. We will welcome a revised version from Uma.

Action item 78-4 for Moore: Convey UTC comments on Proposed Draft UTR #16, currently UTF-8-EBCDIC, to Umamaheswaran. Let him know UTC would welcome a revised version for the February UTC/L2 joint meeting.

Action item 78-5 for Aliprand: Put placeholder for Proposed Draft UTR #16 on agenda for February meeting.

UCS-4 Unicode Conformance

(No document for this discussion)

Davis- The issue is that a UCS-4 implementation is not Unicode conformant. Do we want to extend notion of conformance so UCS-4 can be included. E.g. Solaris is already using 32-bit.
Ed Hart- HP has same issue. We should accept UCS-4. Doesn't make sense not to.
There was discussion of the encoding for the Byte Order Mark.
Whistler- I am against this proposal. It affects the book schedule. We need to consider implications for 10646 implementation with Unicode semantics. What does it mean for API, on the wire, or file sharing if we do this?
Honomichl- Which parts of conformance clause does it not comply with.
Davis: 16 bits.- Implementation is important to the interpretation of the characters.
McGowan word size is irrelevant.
Davis- The problem is surrogates -
Honomichl A 32 bit word and 2 half surrogates is ok.
Hart: UCS-4 and the BOM for it, are 2 separate issues. Can we get around this by allowing systems to interpret as UCS-4?
Davis: No.
Hart: Can we expand context to UTF-8?
Davis: No. Then we would include EBCDIC, UTF-8…

McGowan said Apple is opposed to the proposal. Even UTF-8 and UTF-16 both being conformant has implications for implementations. Adding UCS-4 has significant impact. Maybe in a few years.
Davis argued that we have vendors that are using 32 bits today. There is a need to understand how to interpret their data.
Hiura- I am strongly in favor of the proposal. Excluding these systems that are doing this doesn't make sense. We should make Unicode and 10646 agree since meanings are already established
Sargent: Win64 is on the horizon. Porting of browsers will require 32 bit words. We need the guidance on how to use these in the reality of today's requirements
Mittelstein- 10646 talks about 4 byte chars, so Unicode should too. SAP is not interested in 32 bit, because it requires more memory, so it would be better if not supported.
Becker- We should write up advantages and disadvantages so we can weigh decision
Whistler- People proposing this should write up what things should be changed.
Moore felt the proposal would introduce confusion.
Whistler- it is a clearly stated idea, but implications are unclear, to simply make UCS-4 conformant.
Are any chars beyond 10FFFF being used today? That would cause a problem for interoperability today.
Honomichl- Is there a real world implication for these vendors, that someone will point out that they are not compliant?
Davis- Procurement standards might require Unicode compliance, the answer should be no.
Whistler- If you support UTF-8 then you are compliant.
Davis- Then it is the same work as was done for UTF-8
Aliprand- We need a written proposal
Whistler- Editorial committee spent a lot of time on UTF-8 and UTF-16 to get it right in the book. We shouldn't have to do this for UCS-4, we need a proposal.
Davis- Call it UTF-32
Whistler- How do you deal with constraint of not having values greater than 10FFFF? UCS-4 allows this, Unicode doesn't.
Hiura said that Sun does not use values greater than 10FFFF.
Discussion centered on the need for a written proposal, giving advantages and disadvantages, and what needs to be changed. In particular, it should focus on the constraint of not having values greater than 10FFFF. McGowan suggested targeting for completion by the end of 1999. Davis volunteered to write it.
Whistler- It is a goal for Unicode to be the only semantic interpretation of these characters, that is a long term strategic goal. (Unicode and 10646 should not have different semantics.)

Moved by Davis, seconded by Long
[#78-M3] Motion: The UTC allows UCS-4 implementations that restrict themselves to characters less than 10FFFF to be compliant in Version 3.0.
4 for; 6 against, 4 abstentions (Xerox, RLG, NCR, SAP)
Motion failed.

Action item 78-6 for Davis: Prepare proposal to make UCS-4 a new conformant encoding form of Unicode.

WEDNESDAY, DECEMBER 2

PRESENT: Apple Computer, Inc.; Compaq Computer Corporation; IBM Corporation; NCR Corporation; Novell, Inc.; Oracle Corporation; The Research Libraries Group, Inc.; SAP AG; Sun Microsystems, Inc.; Sybase, Inc.; Unisys Corporation.

BY PROXY: JustSystem Corporation (Hideki Hiura, proxy)

(Total members represented: 12 (one by proxy))
Quorum = 10

NOT PRESENT (at time of roll-call): Booz, Allen, Hamilton, Inc.; Hewlett-Packard Company; Mathema Software, GmbH; Microsoft Corporation; Reuters, Ltd.; Silicon Graphics, Inc.*:* Xerox Corporation;
(Total not represented: 7)

Version 3.0 Code Charts

Freytag distributed a review draft of the code charts of Version 3.0, and requested that comments go to him. Comments on the text in the names list, should be copied to Whistler. The code charts will get mailed to companies were not represented at this meeting.

Whistler: Four Sinhalese characters (which Ireland asked to have withdrawn after the WG2 meeting) are already removed from the next version of the draft, so there is no need to report this as a problem.
Becker: The Hebrew font is unacceptable.
Freytag: We need a font of acceptable technology, and to have approval from WG2, to change the font now.

Action item 78-7 for Becker, Aliprand, others: Pursue acquisition of different Hebrew True Type font for code charts.

Becker has a font and had submitted this earlier.

Changes to Unicode Data

[Document L2/98-390]

Dürst: With respect to changing Indic characters, will changing them from being fixed position to not being fixed, cause problems for characters with vowels above and below?
McGowan: This is a leftover problem which still exists. The Tibetans and others went over this in excruciating detail, and this is the best we can do.
Whistler: Burmese, Khmer and other languages also introduce a lot of problems with fixed position classes. There are not enough classes.
Davis: Fixed position design was not well thought out. This fixes these problems.

Suignard: We have to be careful about how these normative changes affect conformance. How you can be conformant with both v2 and v3? For example bidi changes.
Freytag: We need to recognize some things are not fixable, simply because they break conformance. We need to describe what kind of changes we permit ourselves to make and those we don't.
For example, we won't move character positions. For algorithms where we don't have a workaround that we can standardize, then we cannot make the change.
McGowan: This is a place where we should consider reference implementations. Changing character properties, for my implementations, are table driven and are run-time loadable and changeable. Field upgradeable.
Roberts-When case changes, we have to ask customers to offload all of their databases and then reload, so there really is a big impact. An alternative is to create a new property that does it right, rather than working around with something that is not right.
There is a conflict between the Read Me (informative) and the book (normative).
Dürst suggested that we need a stabilization period. Perhaps we need a statement that some things are not stable for one year after they are documented. This may not be practical, but we need some kind of solution like this. He agreed with the idea of creating new properties. The meaning of certain properties need to be clarified.
Hart asked if a history is maintained of changes and why they were made. Whistler said there is no audit trail per character, but there is one per file: a list of all changes that went into the file.
Dürst: Downloading files from the internet is viable, but allowing these changes doesn't allow for interchange.
Davis: The officers have an action to clarify what the versions of Unicode are, so users can find out what a particular version means. One of the purposes of this work is to lock down identifiers, and character definitions. For sorting, there is not a fundamental relation between sorting keys and the original compatibility information. Sorting is tuned.
Adding properties, new categories is more difficult for people, because it is a partition. You have to change your code, and your API, not just changing an assignment.

Dürst asked about identifiers (needed for XML).
Whistler: Identifiers are in section 5.15. The text is stable enough.
Roberts: Our Japanese experts have said they did not want middle dot as identifier.
Moore: This was discussed in Japan and there was a general consensus in Japan that this was wanted.
Freytag: We should identify which properties are locked down, and which are not so stable such as bidi.
Moore: With respect to bidi it makes a difference whether we are discussing algorithm itself or just properties. Just changing property of a single character does not necessarily have a major impact.
Freytag: The fact that the impact is not easily visible is the problem and we may understand it, but our users will not.

Moved by Moore, seconded by Whistler
[#78-M4] Motion: To accept the changes to the Unicode character database specified in document L2/98-390.
12 for; 0 against; 2 abstentions
Motion approved.

Action item 78-8 for Whistler: Fix read Me file of UnicodeData to say that Case is Normative.

Whistler raised the issue of properties specified within ISO, specifically, WG20 believes it owns properties. There is a political movement to WG2 where people are closer to the defining organizations for characters. UTC views on properties need to be conveyed to other groups that are just beginning to understand properties and standards around them, e.g. identifiers.
Hart: How does WG2 decide to add characters or not without understanding properties of the characters? They should be taking action to understand these.
Davis: I would like to see a concrete proposal for the February meeting on properties we should be more strict about, and which not.
McGowan argued that we should have an implementation to prove a property is a good or bad thing. We learn from our implementations, which is why they have been changing.
Davis said that the idea of temporary properties would be especially appropriate for some of these newer scripts e.g. Thaana.
McGowan agreed, because without an implementation, we do not have a guarantee it is right.
Whistler: If instead of a file, we had a true database, we might add a calculated field with a metric for how stable and reliable a property is.
Suignard: Strongly agree. I worry about how procurers make use of the word normative. How do we express they are normative, but they are going to change?
Freytag: Michel's comment ties to chapter 3 on conformance
Davis took the action to suggest language for Chapter 3 on properties and Chapter 4, the definition of which properties are normative, and not.

Action item 78-9 for Davis: Draft text for Chapter 3, Conformance, covering the issue of levels of conformance.

Special Casing Properties

[Document L2/98-398]

Davis: Because certain processors presume one letter mapping, the proposal is to add an additional file for locale sensitive or special conditions for characters such as Ess-zet, Greek, Turkish letters, iota subscript. Whistler said this would mean an addition to the case tables, not just the conditional tables. McGowan: So this is a case where uppercasing changes a nonspacing mark to spacing? Davis: Yes. Casing is not reversible.

Iota subscript
Combining class change to 240.
Chart is to show middle form of letter.
UniData will show an uppercase of upper subscript Iota
Title case either goes to capital iota or leave U+1FBE

Verify decomposition

Whistler: This breaks the rule for constancy of title and uppercase.
The rule that only digraphs change between title and uppercase was discussed.
Whistler: So the rule now becomes both digraphs and combining characters…
Davis: We need to clarify whether this is a hard fast rule, or an accidental relationship between these characters.
Davis: Will take action to include discussion of Iota handling and a recommendation for which option to take in the documentation.
Whistler: The documentation should highlight that this is informative not normative.

Moved by Davis, seconded by McGowan
[#78-M5] Motion: To adopt special casing specified in document L2/98-398, *Special Casing Properties*, except for the iota subscript.
Unanimous
Motion approved

Action item 78-12 for Davis & Whistler: Incorporate special casing as approved into the Unidata directory as another file. Including heading, disclaimer, etc.

Normalization

[Document L2/98-404]

W3C and ECMA Script are looking for what to do with normalization
Davis said that hangul characters decompose double-k as some other characters. If you compatibility decompose and then recompose with canonical you get kak+kfinal. To address this, doubled consonants have no decompositions. Alternatively, we could say they have canonical decompositions. If we do either, it will resolve. Davis recommends, if they are not canonically equivalent we should describe it.

Whistler: These are also a problem for collation tables. It was easier to ignore compatibility and not decompose to final forms and just use canonical. With decomposed it is very hard to weight.
I am in favor of removing the compatibility decomposition since it is not useful for normalization or collation. However, we should recognize them in some other way since they are useful for input methods.
Davis: Koreans would not be unhappy, since they think of characters as having 3 pieces and not more, due to the double consonants having two.

Dürst, speaking for the W3C, agreed with what was said about Korean. Also, a composed form is not wanted for Hebrew. These should be resolved for Version 3. It is important that IETF, W3C, and ECMA Script all use the same way of normalizing things to facilitate interchange. Dürst will raise the issue at the IETF. If the IETF thinks it is too difficult, then Dürst said the W3C would have to agree and support them. Having a uniform solution everywhere is that important. Davis offered a contact.

Dürst: In a prior meeting with Mike Ksar for WG2, and others, we said this would be available for Version 3, and we therefore told some W3C working groups that it would be available with data.
Davis: Other than new characters, we believe the data is mostly final, except the Hebrew cases.
The timeline should be 3.0 for this report and having all fixes to data table.
Hart: What do the Koreans want vs. what somebody else may be implementing today? Would they be equivalent?
Freytag: Even the largest MS Windows fonts do not contain these characters. We don't get feedback from the Koreans.
Davis: The base characters are in Jamo. The obscure ones are not in any font. It was very clear with all interactions with the Koreans, they really wanted the 3 letter form.
Whistler: I agree with Martin, that this should not be a final tech report until the data table is final. The document is improved though.
The definition of “combining character” in this context was discussed. Whistler suggested restriction to something that decomposes to more than one unit, or recursively decomposes.

Davis: Angstrom decomposes to A-ring to A + Ring. Composing, the preferred one is A-Ring. This is accomplished by looking in the table to see that the character decomposes to more than one other character.
Mittelstein: How do we decide whether to normalize or decompose? It is important for interchange.
Dürst: - In W3C, for XML and other things that have to work together we specify the form you have to use. So it wouldn't be good for Unicode to prescribe which to use. We suggest normalization form C. It would be good if we could cross-reference each other.
There are many cases where you can't specify this. Sometimes you want compatibility and sometimes you need to distinguish and it is important to do so. For example for sorting vs. printing.
Davis: Decomposition is easy. Composition is not, you will get 5 different approaches with different results. We need to make sure composition is well defined.
Whistler asked for a general recommendation as to when to use form C or form D.
Freytag: We should be clear that decomposed form is the recommended form for interchange, to avoid having some composed one way and some the other.
Whistler: It depends on the script. For Latin, composed makes sense. For Hebrew we hear very strongly that decomposition is the best. We should allow it to vary as the implementation needs.

Davis: I will update the documentation with feedback from this meeting. Everyone needs to read the doc and give feedback to Davis. Please put "normalization" in subject line, before Jan 15.

Action item 78-13 for all except Davis & Whistler: Provide feedback (use subject NORMALIZATION) to Davis on Proposed Draft UTR #15 before January 15.

Action item 78-14 for Davis: Revise Proposed Draft UTR #15 to incorporate feedback.

Action item 78-15 for Whistler: Revise UnicodeData file in accordance with L2/98-390.

Action ten 78-16 for Aliprand: Put Proposed Draft UTR #15 on agenda for February meeting.

Collation

[Document L2/98-400]
Changes after discussion with 14561 people.
2 main changes:
a) Some to the main algorithm on page 7 step 2, to handle an edge condition with multiple non-spacing characters.
For example, Z < A-ring and Z <A-ring Cedilla. Issue is that canonically equivalent strings are treated equivalently.
b) step 3 Pushing
French requires accents be processed in reverse order. However, Arabic goes forward. So when mixing the two, go forward and then use pushing to insert backwards-sorting accents ahead of their base characters.

Roberts: Do any other languages have reverse processing at other than second level?
Davis: No, except perhaps some French-influenced orthographies.

Davis: For Thai, you need to look at the following character, to properly sort vowels that precede consonants, but should sort after.

Dürst: We should provide some supplementary text to make this more clear. There also should be a statement that this supports these languages well, and which languages, such as Thai, Sanskrit, etc. that it might not be suited for.

Whistler: WG20 removed many of the things that we took exception too, such as the API. Their template table is generated from the same data, but is reformatted, and key construction is not the same. It is not as clear as this approach, and they have not bought into normalization prior to key construction.
The default behavior, given the data has the same parentage and the algorithms are same (number of passes, etc.), should be equivalent. But the tailoring approach is different and therefore they may not produce the same results.

Hart: We want one standard, not 14651 and Unicode collations.
Whistler: We can't have it for the same reason you have two character set standards. The real question is: how do you specify that the tailorings are equivalent? Ultimately users don't care -- they just want to know that they specify German1 and they get what they expect regardless of the architecture underneath.
Hart: There could be a registration authority for the tailoring.
Whistler: There is one. There will be a POSIX registry.
Dürst: If there was a tailoring registry, then you could contrast tailorings across the web.

Mittelstein: The weights are 16 bit values. So there is an endian impact.
Whistler: Yes.
Mittelstein- How about surrogates? Will they need bigger key elements?
Davis: No. These are covered by the discussion on page 18, weight derivation.

Action item 78-17 for Whistler: Once the 2.1.8 data table is defined, regenerate the collation weighting table, and post it in the Working Groups section.

Action item for all except Davis & Whistler: Provide feedback (use subject COLLATION) to Davis on Draft UTR #10 before January 15.

Action item 78-19 for Davis: Revise Draft UTR #10 to incorporate feedback.

Action item 78-20 for Aliprand: Put Draft UTR #10 on agenda for February meeting.

Line breaking

[Document L2/98-407]
Referred to an ad hoc caucus.

Proposal to fix Greek Iota-subscript

[Document L2/98-412]
Whistler: Favors 3b- you can get the title shape without a change in decomposition. Also the forms with Iota subscript and Iota after Omega, are not the same, and people would expect them to have the same representation. McGowan concurred.
Davis: However, if I type capital Omega and follow it by an adscript, it is not the same as the Omega with a subscript. I don't think people want to have shaping for the Greek characters. They are all precomposed so you don't have to do that. They are used to changing character codes for uppercasing, not used to changing rendering.
Freytag: We don't want two character codes to distinguish the two font forms for the character, unless my perception is wrong about these characters and they have equivalent values and it is a pure rendering difference. I am not sure that people will use decomposed forms for input, and am concerned that this is being based on something that many people will not be doing.
Davis: Most scholarly systems use fully decomposed.
Mansour: Most people would say that omega with iota subscript capitalizes to omega with iota after and not as the capital omega with an iota subscript. The scholars were mostly using 7 bit systems which in part is why they use decomposed, but also it is natural for them to think decomposed.
Freytag: We don't have the right expertise here. We should write this up and have experts decide between the 3 options.
Mansour: As this is a scholarly language issue and not one of modern usage, going to the experts will just generate a debate. We should just look at the literature.
Some discussion of other iotas and possible second order effects.
Aliprand: Do we need to decide today?
Whistler: The decompositions do needed to be decided today. We have an implication for casing of the iota subscript which is not reflected in the case table. We need to address this.
Davis: I would rather choose B and leave the database in a coherent state than have the current state and leave it broken.
Whistler: B is a one line change in the database.

Moore: We could do that and change later if we need to.
Whistler: We could do this without prejudicing the vote in February.

Moved by McGowan, seconded by Davis
[#78-M6] Motion: To fix the iota subscript as in document L2/98-412, option 3b, but to revisit this non-prejudicial decision at the UTC/L2 joint meeting in February. .
Unanimous
Motion approved

Action item 78-21 for Whistler: Apply fix for iota subscript (as in L2/98-412, option 3b) to UnicodeData.

Davis noted that this will put the database canonical decomposition in order for the first time.

Action ten 78-22 for Aliprand: Put iota subscript on agenda for February meeting.

Action item 78-23 for all: Find out about treatment of iota subscript in implementations for polytonic Greek.

Enclosing Triangle

The action item to submit this character to WG2 was never carried forward. We should not try to cram this in to 10646 just to make it into Version 3.0.

Action item 78-24 for McGowan: Prepare summary proposal form for Enclosing Triangle character and submit to Mike Ksar.

East Asian Character Width

McGowan: The comment about UTC and WG2 in the last paragraph before the section on conformance needs to change. We shouldn't put words in WG2's mouth. He suggested making this an informative technical report rather than normative, to help people with transformations and display of legacy data.
Suignard agreed. Because it is unstable, characters will change, there are some ambiguous Latin characters. This is a guide, and might help you for example with line breaking. Ambiguity depends on locale. For example, a character may be ambiguous in Japan but not in Korea.
Long: Why are the C0 characters (U+0000-U+001F) ambiguous? These are not ever wide.
Freytag: I will check the original implementation to see if there was a reason for this.
Whistler: Why is Won half-width and Yen is narrow?
Freytag: Because ISO8859-1 has Yen sign. No other character set has legacy half-width Won sign.
Whistler: Then this should be an informative report, with information on legacy character sets as of a certain date, and state that every assigned character that is not listed here is neutral.
Davis offered to generate the name of the first char of each range in the list and provide to Freytag.
Becker: "What Unicode standard says today" should be replaced with a reference to Unicode Standard, Version 2.0.

Moved by Freytag, seconded by Davis
[#78-M7] Motion: To make Draft UTR #11, *East Asian Character Width*, into a Unicode Technical Report after incorporating editorial changes received at the meeting and these changes:
1. Drop X-unassigned list and include a prefatory note about characters that are unassigned;
2. Add the Euro to the ambiguous category.

11 for; 1 against; 2 abstentions (Justsystem, Sun)
Motion approved.

Action item 78-25 for Davis: Help Freytag to add name of first character in range to lists in UTR #11, *East Asian Character Width*

Action item 78-26 for Editorial Committee: Fix text on p. 6-30 per comment in Draft UTR #11.

Action item 78-27 for Freytag: Make Draft UTR #11 into UTR, incorporating comments from the meeting, and post revision in UTR section of Web site.

Davis moved to adjourn the meeting. Whistler seconded.

THURSDAY, DECEMBER 3

PRESENT: Apple Computer, Inc.; Compaq Computer Corporation; IBM Corporation; Justsystem Corporation; Microsoft Corporation; NCR Corporation; Novell, Inc.; Oracle Corporation; The Research Libraries Group, Inc.; SAP AG; Sun Microsystems, Inc.; Sybase, Inc.; Unisys Corporation.
(Total members represented:13)
Quorum = 10

NOT PRESENT (at time of roll-call): Booz, Allen, Hamilton, Inc.; Hewlett-Packard Company; Mathema Software, GmbH; Reuters, Ltd.; Silicon Graphics, Inc.*:* Xerox Corporation
(Total not represented:6)

Xerox Corporation representative arrived subsequently.

Philippine Scripts

[Document L2/98-397]

Moved by McGowan, seconded by Moore
[#78-M8] Motion: To accept document L2/98-397, *Revised proposal for encoding Philippine scripts*, for addition to the Unicode Standard after Version 3.0.
Unanimous
Motion approved.

Symbol D with a Tail/German Penny Symbol

[Document L2 98-309]

Moore noted that properties for this character needed to be addressed.

Moved by McGowan, seconded by Sargent
[#78-M9] Motion: To accept document L2/98-309, *Proposal for "Script D Symbol with Tail*," but encode the character in the Currency block.
8 for; 0 against; 3 abstentions (SAP, NCR, IBM)
Motion approved.

Whistler recommended the next available codepoint after the Drachma, which is in the WG2 queue.

Moved by Whistler, seconded by McGowan
[#78-M10] Motion: To suggest 20B0 as the code value for the character accepted in Motion #78-M9.
13 for; 0 against; 1 abstention (Xerox)
Motion approved.

Criteria for Encoding Symbols

[Document L2/98-311]

McGowan: For the case where symbol is trademarked, it should not matter whether the encoding is requested by the trademark owner. Honomichl noted that we rejected the Grüne Punkt trademarked symbol.
Moore recommended that this paper go as input to the Editorial Committee.
Texin: "must be searchable" needs clarification.

Aliprand: Since we feel this is straightforward enough, we don't need a motion, and will refer it to the Editorial Committee.
Action item 78-28 for Editorial Committee: Consider incorporation of *Criteria for Encoding Symbols* (L2/98-311) into Version 3.0 and into *How to propose characters* section of the Web site.

Terminal Emulation

[Documents L2/98-353 L2/98-354 L2/98-355 L2/98-413]

Hart introduced the topic. Frank da Cruz has a background with terminal emulation and PCs. The aim of the proposal is to standardize these symbols used in terminal emulation, so that there is a common standard and vendors of emulators don't create a vast array of symbols.
Aliprand: Why can't they use the codepoints reserved for control symbols?
Hart: The second aim is to define a set of symbols for C1 area.
Hart: wants to define a set of 3270 symbols
McGowan: C1 area has more than one set of symbols, this proposes 6429.
Aliprand: and we should consider 6630 as well.
We need multiple sets of pictures depending on what standard is being applied.
Hart: Are those font variations for these code positions, or do we want a separate set of symbols?
Becker: The symbols that are there now, I put there as placeholder for whatever symbol you want, so that they wouldn't propagate lots of symbols in the standard.
Whistler: If you were using one control set, you would use a font appropriate for it. If you were using more than one set, you would need a sophisticated protocol to switch fonts between them.
Mansour: There are many protocols that don't require pictures at all.
Hart: To debug a datastream you need to have glyphs, so you know what codes are present.
Mansour: But unless the protocol is Unicode, you don't need them
McGowan: No matter what the datastream is, your debugger display engine might be Unicode however, independent of what is in the data stream.
Mansour: that opens the door for glyphs for many datastreams, those that we don't have an advocate for yet.
Honomichl: We could treat it like interlinear, with a begin and end and a set of letters indicating the control code
McGowan said that the Unicode Standard currently provides for this, allowing you to create whatever symbols you want for these codes. Becker suggested referring this to AFII.
Davis: We should encode things that are in common use for running text. This doesn't really qualify. To reduce the pressure to have these defined, we can recommend the private use zone. If the people that want these use the same code points, they can interchange.
McGowan: John Cowan of SIL has all kinds of stuff like this in the CONSCRIPT registry, for people that want to exchange data in the private use zone.
McGowan: For UTC to specify how the private use zone is used would get us involved in many of these proposals. It is the nose of the camel…. We don't want to go down this road. The Conscript group will work with each of the groups that want to use these.
Davis: It is not just their concern. If we reject these, then they will go to WG2, to ask for symbols that way.
Suignard: The private use zone should be for private use. This is not private, it is for interchange. Once we do C1, then there will be EBCDIC, and anyone can use C1 and it is wide open.
Roberts: I need some convincing these symbols are being used for C1.
Aliprand: We need some evidence of these usages. I don't see why they can't just use hex codes.
Davis: People want to see a symbol
McGowan: But they can do that without our defining a symbol
Davis: Agreed.
Mansour: It makes sense if they will be transmitted as such and not as a private code. I like the idea of having a group for them to go to for registration, it doesn't prevent them from coming to you for codepoints later.
Mittelstein: This is a request for standardization of the user interface to represent symbols like lf, cr.
Whistler: There is a contrast between people who are emulating and maintaining legacy systems and people who are designing new software using these points.

With respect to 413, have you ever seen these symbols before? We should have evidence. It is not clear where Everson got these.
Becker: Yes they need to have evidence of these to meet our criteria for encoding in Unicode.
Mansour: As a font company, we get requests for this. Usually people want to perpetuate symbols that they are used to using.
Hart: The terminal emulation vendors do refer to the original terminals. Keyboard mapping is an issue too.
Mittelstein: Is this a proposal to capture existing symbols or defining new symbols that people can use?
Aliprand: The latter, symbols that are newly defined for these. It is an open ended repertoire.
Davis: Our choices are to encode them, which I think is a bad choice, or suggest that they use private use zone, so they have some guidelines, and we post them to the web. Another option is to give them recommendations about using private use with greater specification, e.g., for this use you must do this, for that use you must do that. Or we could point to some other group where they could be registered.
McGowan: To propose these additional characters reflects they haven't thought through the other reasonable uses. Of Mark's four options, we can go back and give some reasonable explanations why they should not encode, and point them to a registry.
Honomichl: Why the Conscript registry? I don't want to bless one particular registry. I would feel better if there were a list of registries.
Davis: we can say "such as".
Moore: We have these 3 proposals, and I agree with the consensus, but we should look at each proposal for the specifics. We might accept nothing, but there might be some good ideas for individual needs.
Aliprand: We could suggest using Conscript as a starting point, and leave the door open for others.
Hart: I agree with Lisa, we should consider each one independently. We should respond though generally as to why it is a bad idea, and to have them start with the registry which will be a proof of concept.
Becker: Frank da Cruz is looking for a codepoint to be assigned to the glyph and that is why he is coming here. AFII wasn't functioning correctly for him in this way. We need to explain that this need, assigning glyphs their codepoints, is not our responsibility to fulfill.

L2 98-355, *Hex byte pictures for Unicode*
McGowan: This is completely out of bounds.
Honomichl: We could point him to one of the other planes.
Whistler: I concur. There is no indication in the proposal for interchange. He is clearly talking about glyphs to represent data that is being transmitted.
Mansour: Yes, it is a font rendering issue.
Roberts: It is a transcoding, not a transmission of particular bytes.
Whistler: There is already a widespread practice of just using hex-encoded ASCII that satisfies this.
Whistler: It is clear there is no consensus to support 355.

L2 98-354, *Terminal Graphics for Unicode*
Suignard: Without glyphs it is hard to understand this proposal.
Whistler: If you look at document 413:
The 2nd column is for 355
The 3rd column is from table 6-1 and then table 7-1 in the last 2 col
So just replace x by e in 413
Hart: He has 2 sets of symbols. E080-E087 are used for 3270 terminals, used in the status display of the terminal, not the 24x80 display area. The other E0A0-E0B2, is for mathematics, and a third set is graphic symbols for tech drawings and the like. The IBM symbols are not even documented by IBM, and as they are in the status area I question whether they are needed.
McGowan: 3270 emulators exist now, and they aren't using encoded symbols, so why are they needed?
Freytag: Is there a benefit to 1-1 transcoding of these?
Hart: These are status symbols, they aren't transmitted in the datastream, it is all internal to the terminal
Roberts: The emulators that I have seen use bitmaps for this.
Freytag: If the usage is only in the driver, then the only value of these is to replace the bitmaps. So a console font can include these and be easier to represent these terminals.
McGowan: If they are only useful for program status and as the proposal says not used for transmission, then they are not good candidates for encoding.

Whistler: It would be better if they submitted 3270 documentation indicating where and how the symbol was used, e.g. When you do this, you see this…
Davis: That is dangerous, because then any symbol anyone wants encoded, they can find some documentation.
Honomichl, It is a minimal requirement, but not sufficient.
Hart: Does IBM have a position on this?
Moore: We do not yet. I think IBM is not interested in changing our terminal emulator code to use these.
Freytag: A lot of people make use of these, based on the email discussions, and they were looking for standardization. I don't see that these characters are variants, many are generic symbols, like the key symbol. They have a general utility, so that is reasonable request. Given that there are only 8, I wouldn't be as resistant to adding these, as to adding the whole group.
McGowan: So we can go back to IBM and ask them to determine if they are useful or not.
Freytag: But the people who are interested in this are outside of IBM.
McGowan: I just want their opinion, not a decision.
Hart: Should we postpone a decision until we have feedback from IBM.
Aliprand: Does SHARE have an interest?
Hart: Yes.

Action item 78-29 for Moore: Determine IBM's position on L2/98-354, Section 5 (3270 terminal status indicators)

Action item 78-30 for Hart: Contact SHARE for opinion on L2/98-354, Section 5 (3270 terminal status indicators)

Discussion addressed the mathematical characters.
Whistler: Some of these already occur. U+221A is radical sign. 0xB1 radical sign with a mark through it is U+237B which is annotated "not checkmark: for negative acknowledge"
McGowan: This is from Data General, math symbol.
Sargent: It doesn't seem to be right for math.
Whistler: It may be a misinterpretation for ack, and nak
Freytag: Recommend checking 0xB1 for unification. And confirm 0xB0 is unified with checkmark.
Whistler: U+2406 is Acknowledge.
Hart: the 0xB1 is coming from a math set, so you wouldn't want to unify with NAK.
Freytag: But the names may come from anywhere.
Whistler: U+2406 is encoded for this. I think it is misinterpreted as a check, but is really a radical sign.
Roberts: Yes, it doesn't seem right, given the source documents.
Freytag: Just recommend they unify and let them tell us why that shouldn't be done.

Whistler: 0xAE and 0xAF are given as right ceiling and right floor corner. U+231B and U+231F are top and bottom right corner.
Sargent: They don't look like they are part of the segment.
Whistler: Are they corner characters, or are they part of the brackets? 0xA0-0xAD, we have seen these and rejected a long time ago.
Sargent: It has been in legacy character sets for years. Don Carroll put them in the math 8B HP character set. So there is legacy support for these. From a practical point of view it is handy to say "this is the character I mean," as it shows up in many fonts e.g. HP, and elsewhere. I have used these in a technical word processor.
Becker: Same lineage as the form character.
Whistler: The corners might be the corners of extended Sigmas. McGowan and Sargent disagreed.
Freytag: Hart can you go back and ask for better glyphs for 0xAC and 0xAD?
McGowan: These are just sans serif parts of Sigma. We don't need better symbols to know what these are.
Beeton: If these are ceiling and floor fragments, why aren't the left equivalents there?
Hart: Aren't the left ones already APL characters?
Whistler: Yes, they are already encoded.

Freytag: We need the complete set so we can see the unification of the sources. We don't need images, just the names. I want to see the mapping tables for the character sets, mapping from source set to Unicode.
Hart: So that will guide us in understanding the relationship between these characters.
Freytag: And then we will understand why some of these can't be unified.

Discussion on symbols for tech drawing.
Whistler: These should be considered along with the bracket pieces from PC printing and legacy set.
Freytag: Are there vertical lines that go along with these that they have mistakenly unified, and are D0 and D1 not extending vertical? If not, then we need an explanation for what they are.
Whistler: Rather than asking this question, someone should go out to all of the printing sets, and make a complete proposal around these. Sargent volunteered.
Sargent: If we add them, then we will have the complete symbol set from Windows.
Whistler: A use for table 7-1 would be to take the block pieces and have a complete set. I wouldn't oppose including these, they may be nice to have. I like the framises.
Hart: That doesn't mean they will be used though.
Whistler: U+25C6 the diamond can be unified. And we have arrows to bars. It is just different aspect ratios.
Freytag: Which are not normative.
Whistler: U+21E5 is right arrow to bar is 0xEC, U+21E4 is left 0xED. We don't have up and down. U+2396 is 0xD2. U+2396 is a standalone symbol.
Freytag: U+25E2-U+25E6 work together.
Whistler: Is the caret next to a bar, not arrow to a bar?
Aliprand: So we need to know what mappings they have done, and why the remainder are not mapped.

Action item 78-31 for Hart: Ask Frank da Cruz to provide source documents and better glyphs for characters x0AC and x0AD (shown in Document L2/98-413, p. 3)

Action item 78-32 for Hart: Ask Frank da Cruz to supply mappings of terminal control characters to Unicode equivalents

Action item 78-33 for Sargent: Develop proposal for a complete repertoire of parts for braces, etc. found in printer fonts

Summary of discussion: We will propose unification and request response from Frank da Cruz. We also need to see their complete mapping, so we know what is already unified.

Moved by McGowan, seconded by Freytag
[#78-M11].The UTC states that encoding hex byte pictures (as proposed in document L2/98-355) is permanently out of scope.
13 for; 0 against; 1 abstention (Sun)
Motion approved.

Action item 78-34 for Hart: Ask Frank da Cruz to respond to unifications proposed by the UTC.

L2/98-353, ***Additional control pictures for Unicode***
Aliprand: We need a recommendation for what feedback we will provide.
Moore: We should just recommend using the private use area.
Freytag: Suggesting a registry is out of scope for UTC.
Moore: We can officially recommend private use, and Ed when he gets back to them can suggest using a registry.
Davis: It is not out of our charter to recommend a practical solution.
Freytag: You can imagine people start using UTF-8, PUA1, and others. They can have so many of them that people feel they need to pay attention to them. It may not be legally improper but it is practically improper.

McGowan: It is improper for UTC to say anything about what would or should happen in the private use zone.
Whistler: I like Mark's reasoning. It prevents UTC from having to deal with these things over and over again. It is easier to handle those, if there is an alternative. Otherwise people keep having to come back with things we don't want to standardize. UTC does not endorse any particular use of the private use zone.
Becker: We have to avoid implying we standardize in this range, and they standardize in that range.
Freytag: If we state that they go register there, then we cross the line.
Jenkins: Adobe has a web site describing their use of private use zone. Apple could have a website explaining their uses.
Mansour: Microsoft has one too.
Sargent: It is used for round-tripping Japanese characters that aren't in the codepage. We use almost all of it.
Whistler: If it is full then we can't be recommending.
Mittelstein: It is a problem if one vendor is so powerful then others can't use it.
Davis: It is ok as long as each use is separate.
Freytag: When you start straddling communities of users, it becomes a problem
Aliprand: Our recommendation is they use private use zone. Then we have the issue of how to address the private use zone.
Whistler: Our recommendation is that they not encode in Unicode. And we don't need to recommend the private use zone, although that is an option to them.

Moved by Jenkins, seconded by Honomichl
[#78-M12] Motion: That the characters proposed in document L2/98-353, *Additional control pictures for Unicode*, not be encoded.
13 for; 1 against; 0 abstentions
Motion approved.

Moore: Ed should thank Frank for producing 3 nicely prepared proposals.

Action item 78-35 for Hart: Communicate UTC decisions on proposals related to terminal control characters to Frank da Cruz. Thank him for a nicely organized, detailed proposals.

Moved by McGowan, seconded by Winkler
[#78-M13].Motion: Motion #78-M12 is not to be construed as ever preventing the UTC from accepting a picture of a bell for encoding.
11 for; 1 against; 2 abstentions
Motion approved.

Mathematical Symbols

[L2/98-406 and L2/98-405]

The Chair welcome Neil Soiffer from Wolfram Research and Barbara Beeton from the American Mathematical Society.

L2/98-405*, Proposal to encode additional … symbols*
Beeton: Many symbols are already in Unicode, there some others that we were asked to exclude and about a third remain and are in the proposal.
Freytag: Epsilon in 1x1, why is this not the same as U+220A?
Sargent: It means element contained in a set.
Freytag: Is there both a "small element of", and a "large element of"
Beeton: I am still researching this.
Freytag: It could be we made a mistake in that the wrong glyph is used for "small element of" in the book. Epsilons include U+220A and U+220D Greek U+03B5, IP U+2208, 4 epsilon in Math, large U+2208, small U+220A straight, left right
Whistler: So we don't to encode new characters where we have a simple mistake.

Mansour: Four of the listed epsilon shapes can occur in everyday Greek. The important point is that these have a particular semantic in math, and you need to preserve the look recognizable to a mathematician.
Becker: Is there only one that can represent “small member of”? If 2 or 3 of these can represent “member of,” that will affect our decision.
Freytag: That’s what we need to determine.
Davis: Mathematicians are using these in a rather glyphic fashion. Also in a generative fashion. They are being invented all the time. We have to look at whether they will be generating lots of new symbols and we should look at whether they are taking advantage of Unicode’s generative capabilities. Page 13 has a number of symbols that perhaps could be dealt with by a diacritic-like approach.
Freytag: There are many more negations than are shown here so you must have assumed some productiveness of Unicode.
Beeton: Yes
Freytag: So you are comfortable with using productive forms of encoding. Have you tried to eliminate those that can be generated?
Beeton: Not diligently, fully.
Whistler: On "negated element of", this could be U+2209.
Beeton: Could be the two sizes. This could be large vs small.
Whistler: Row 7 with vertical is different from row 6.
Beeton: I don't know yet.
Whistler: We had, when we designed this, tried to unify negation with all of the operators.
Becker: A test for that is to know whether they can occur in the same equation with different meaning.
Freytag: Don Hughes indicated there is an element of taste in this, in that someone might use a double line rather than a single just because they feel it makes their point better. We need to establish which distinctions are semantic.
Becker: There are pretty large systematic variations of these that exist.
Soiffer : Less, than, Equal, Greater than, often take on different semantics depending on context. So often the same symbol will have different meanings and authors will choose variants to emphasize certain aspects of the operator.
Freytag: Yes, some of these are free variants,
Soiffer: The meanings of these symbols also vary over time, and they may diverge.
Davis: We should not predict divergence, since they may converge.
Page 17 row 7: If we find multiple forms of a symbol in the same document then we can know that they are different symbols, rather than in separate documents which could just be author's style.
Freytag: With all these variations, can we triage them into buckets of: straight equals, slant equals, etc.? Also we need to make sure they are not already coded in.
McGowan: This won't happen quickly. It needs greater scrutiny. We need the sources for these symbols, and documentation as to why it cannot be unified. Page 22 col 1.row 5, 6,7, 8: Can they be unified with CJK variants?
Beeton: I hesitated to unify with CJK. For example, if there were halfwidth and fullwidth versions, how to choose.
McGowan: Yes, those have other specific semantics.
Mansour: Even if shapes match, unifying with CJK block can have strange consequences.
Freytag: We have already unified CJK symbols with other mathematical operators.
Mansour: CJK standards contain everything from scientific and math, but others have special typographic properties.
Freytag: Yes, but word processors can look at how the symbols are used, in ideographic text or math equations and determine which look to give them.
Soiffer: Yes, but CJK users will use these symbols in their text when writing about math, so you can’t always decide which way to display.
Davis: It might be useful to look at the symbols in page 17 and see which can use combining marks, there are dots above, below, regular negations and vertical bar, whole batches can already be encoded. Second, we can look for common uses, for example several have x-below, and it might be better to create a combining “x-below” that will allow production of many characters, beyond what is here. We need the analysis to say what could be done.
Freytag: However, page 17 col 0 row 13, 14, it may not make sense to not introduce a combining plus.

U+2250 has various forms of equal signs and we have not excluded single dot above, below. Identify which are suspected of being graphic variants and which we are sure are typographical variants. We should err on the side of inclusiveness, because mathematicians can use these.
Mittelstein: I had one professor who had unique choices different from all the other professors. We have to remember that we need to unify these.
Freytag: Yes, but this committee is not the one to decide which are due to an individual and which are not.
Davis: Is there a standard symbol for "nan"?
Soiffer: There isn't one yet, I have seen stylized "n"-"a"-"n", or "zero over zero", but there is no standard here yet. For dots, there is a standard size and placement, but for slashes and underlines, there are different sizes.
Freytag: For combining characters, the assumption is that the rendering system looks at lots of factors in coming up with better rendering. Reality is that it looks lousy if you just use simple overlaying approach.
Davis: Opentype, Adobe True Type will use mappings from the coding to specific glyphs.

Whistler: The other side of the problem is recognizing when things are the same, when people are just trying to make the characters look right. And stylize them.
Davis: Yes, they should compare the same.
Mittelstein: We discussed this morning "searchable". We can ask whether people would search the internet for one symbol and not the other.
Freytag: We need to be sure that the mathematicians will agree to use the proposed set and not object due to this or that symbol being missing. We have done a poor job of math until now. We need to be sure to correct this, and to get endorsement from the mathematical community. We are just asking if you could use overlays to keep the numerousity down.
Davis: You don't need to invent combining characters in some cases, because we already have macron and 2 dots above.
Moore: page 25 col 8 What are the differences between the squares, filled unfilled?
Beeton: I don't know yet. I need to research this further.
Sargent: We have the source information on a lot of these.
Davis: The black squares are not the right glyphs.
Beeton: Yes, we didn't know the right glyphs for these.
Soiffer The filled squares from Mathematica are dingbats.
Soiffer asked whether anyone objected to the doublestruck letters on page 4, which represent well known constants. They are usually drawn with Latin letters. They deserve distinctive drawing, but are not always done this way, perhaps due to limitations of typesetting at the time.
Davis: We are reluctant to create distinctive forms on our own. For example we have not created an abbreviation dot. Without it, the dot indicating an abbreviation can be confused with other things, but we did not create one.
Soiffer: Like "i" as used in an index and "i" as square root of -1.
Davis: The danger in separating out semantics is you get visual ambiguity.
Soiffer: These are very common units.
Davis: But do we separate all units. Is "St." used for "street" and "saint" different? The question is if they get encoded for general interchange then there is the problem of visual ambiguity.
Soiffer: If you don't encode separately then you always have ambiguity.
Beeton: We get more requests for people to enter into symbolic systems. Without these we cannot do this. These are important distinctions.
McGowan: What they are asking for is the character that means square root of -1, not an "i". Do we want to go down the road of defining all of these symbols?
Davis: The problem doing this is it is like asking for some kind of markup of plain text. This "c" means speed of light, this "c" means something else. It is more of a markup issue than an encoding issue.
Roberts: The Koreans encoded various Hanja characters for pronunciation,
Freytag: We encoded U+2044 fraction slash specifically because it has special properties. "d" used as differential operator versus as a symbol could be worth adding. However, "c", or Avogadro's number, these are not germane because a human reader cannot distinguish them. In looking at an equation with c in it, you assume it is the speed of light and not a variable, but you may not be sure.
Freytag: In electrical engineering, the i becomes a j, there it becomes useful to have an encoding for square root of -1. Is there anything in the way "e" is used ?

Sargent: e as charge and e as used in logarithms.
Becker: e as a number has very distinct semantics, analogous to Cyrillic yeh, which we didn't remove because it looked like e. It was required because it may need to be searched separate from an English e. The distinct semantics are a good argument for distinct encoding.
Davis: We considered decimal point which sometimes appears as comma or dot. It is a disaster. I send a symbol meaning "i", and I send it to you it appears as a "j" and therefore is given a completely different meaning. There are many cases where we have been forced to encode some of these things due to legacy encodings, we should think twice before creating them.
Soiffer: MathML is a standard that encodes them separately. Historically all computation systems have distinguished between them. It will be important to future systems. In Mathematica, because it is not good to have things that are visually same, we do actually use a different glyph. Yes, it is a slippery slope, but these are so common we feel they are important.
Freytag: In summary- we have a request from the user community- we should take that into consideration. We should consider the size of the set. I don't hear a request for "pi".
Soiffer: Mathematica doesn't, some others do. Probability also uses pi, and so there is some confusion possibility.
Freytag: So mathematical system users, do use e, i, d D, imaginary unit, natural exponent, differential d upper and lower (not Eulers constant).
Whistler: What are these others?
Beeton: They came from different sources.
Freytag: We are not ready for a decision, if you go back and collect more evidence as we have discussed.
Soiffer: What about nan, and true/false and other potential concepts?
Davis: We don't encode concepts, but if there was a symbol for nan … We do need to be careful about not setting a precedent, so that other disciplines don't then ask for special symbols.
Freytag: It would be helpful to document these things with samples and why the needs of mathematics are different from CAD/CAM and other groups. And we should distinguish those that are structurally different such as differential d. If you could take into account the arguments you have heard here, and explain why this is still the best solution.
Becker: How big is the scope in mathematics? How many more symbols such as pi are there? Then we can see how big the danger is and the scope.
Aliprand: This has been very productive. I look forward to a reorganized and reanalyzed proposal.
McGowan: It would be nice to post this information as it comes in on the web,
Freytag: It would be good to have the proposal and review again in February, while this discussion is still fresh. If it waits for July then the discussion may be forgotten.

Action item 78-36 for Beeton: Investigate "member of set" variants.

Action item 78-37 for Beeton: Indicate composability of characters in revised proposal.

Action item 78-38 for Beeton: Identify possible typographic variants in revised proposal.

Action item 78-39 for Whistler: Convert latest version of math file to Excel spreadsheet, post to Working Group section, and send notice to "unicore" when spreadsheet is available

Action item 78-40 for Aliprand: Put math on agenda for February meeting as an "if time available" item

Three additional characters
Soiffer asked if he could propose 3 characters which came up in MathML.

f(x+y) is interpreted as function call. But a(x+y) is interpreted as multiplication. So MathML added 2 operators "invisible times" and "apply". The third character is "invisible comma". When you have the symbol "m12", you don't know if you have the 12th element or the coordinates 1,2.
Freytag: These seem very necessary. They are analogous to the bidirectional character RLE.
Roberts: I like invisible comma, as it helps with search. The others don't really help me in text processing.
Freytag: If you had invisible times, do you still need "apply"?
Soiffer: Juxtaposition.

McGowan: It seems like you need markup language.
Freytag: They are needed for screen readers, to know that there is an implied symbol there.
Jenkins: Having an Invisible Comma you could use the Invisible Space.
Suignard: Then you can't do math in Thai.
Freytag: Yes, we don't want to overload symbols too much.
Soiffer: Possibly we could use invisible space.
Freytag: We can consider using invisible space, but it is more clear to have the specific meanings.
Soiffer: I would also like to propose hinting for breaks: good place to break, bad place to break, no break. I would also like to propose a new line with indenting .
Davis: Use ZWS, ZWNBS (zero width nobreak space), for new line there is PS, and for indenting there is LS.
McGowan: The more of these I see, the more clear it becomes that we need markup, The more fancy requirements need to be satisfied by markup.
Freytag: PS implies a gap, so I am not so sure about it. And we don't have an equivalent for no bad break, so we need to consider this.
McGowan: For indenting newline, you could use new line and tab together.
Davis: Are these break codes are to guide an algorithmic line break?
Soiffer Yes.
Whistler: Would the suggestions (ZWS, ZWNBS) work for you?
Soiffer: I would have to do recoding, but yes.

L2 98-406*, Mathematical variant tags*
Mansour: There is an assumption that math has to be representable in plain text.
Sargent: However, there is an important point about being searchable in plain text.
Texin: What is the mapping?
Sargent: Used for math.
Freytag: Needed for math.
Soiffer: We tried to do this, and what is there is completely inadequate. We ended up adding 5-600 characters to do math. And then AMS wanted 800 or so.
Mansour: If we needed 5-600 symbols to make it complete, what would be wrong with that?
Sargent: This proposal is extensible. Also this does let you search for the underlying character.
Soiffer: You have to check if you search for the underlying character that you don't get a false match and check the character after.
Winkler: What hinders you from implementing this anyway as a higher level protocol?
Freytag: What we have today is not usable as it is incomplete.
Moore: I support this, it is a good goal to support math better in plain text. It is not a good idea to replicate all of these characters, and this is open ended and productive and will cut down on our future need to add characters.
McGowan: I am supportive, but would like to see an implementation that demonstrates this. Also Sargent once brought in a proposal for superscript and subscript operators which perhaps should be considered.
Beeton: There are other symbols, such as what do you put a radical over, length of a line, …
Whistler: You call these things math variant tags. This is missing a description whether these are variant markers, this character plus this equals that. This what we are doing with Mongolian and we need an encyclopedia of these things. Even if they are predictable, we need this mapping. Or is it open ended or are there only a certain set of characters that I can apply this to? For example can I math-bold a Devanagari character? Which characters do these operators apply to?
Sargent: We will specify that.
Whistler: Are all variant tags formally combining? Do they all have the combining property and be included in annex B in 10646? How do we express that in our code charts? We have not yet had an invisible format control that was an MC.

Freytag: We did have in mind a specific set of characters that this would apply to, such as uppercase characters. For the combining marks, the variance selectors are shown as dashed boxes (format controls), if we are going to have a number of them then we can make them selectors, so they are not necessarily combining characters.
Whistler: But they would have to have the combining property.

Becker: You can't say that they cannot be applied to other characters, only that the result is unspecified.
Freytag: We would say that it is only applied to some list of characters, and exclude the ability to apply to other characters.
Becker: You have a table of what can happen to any character, but most of the table is null.

Mittelstein: It looks like markup to me. I thought Unicode was encoding characters not markup. There is a concept of language tagging, which also is a form of markup. I am concerned that we have the feeling that for some specific needs we might like to use markup, we should plan very carefully. This is suffix tagging, and we need to make sure it works with other kinds of markups that we use. What happens when we mix a prefix with a postfix?
Mansour: There are mathematicians working in many scripts that use bold and italic and so on and it may be counter productive to limit what they apply to. Since they apply to one character at a time that is limiting, but it won't be surprising if someone generates these on an entire paragraph to get the style they want. I agree we should make sure this is consistent with other things we want to add later on.
McGowan: We need to make explicit whether this is open ended or not, as Ken said.
Whistler: We have to tradeoff the cost of adding some math symbols vs. the issues that this architecture brings with it. We have to clarify we would do this rather than a plane 14 tagging approach. Limiting them does not prevent our making them useful for something else later, it is still under standards control. It is a standard interoperable way of working.
Roberts: I am not happy with limiting A-Z, once you do this, the cat is out of the bag.
Mittelstein: Do we need to have them combine? For example, we could have a bold-italic or a bold-script.
Long: To do a search, you can have the character generate a sequence that can be searched for, so it does not need to be a search for a single character.
Whistler: I wanted to come out in favor of single tags, since it discourages people from using these for other purposes, and a fixed set of things that this can apply to.
Yang: I think its good for math, but it should be extended to other purposes.
Sargent: I will fill in the list and the table and have it for the next meeting.
Freytag: There is also the super and subscript operator and insure that there aren't any other operators that we have left out from consideration.

Action item 78-41 for Sargent:
Create two lists and append to math proposal (revision of L2/98-406):
a. table of character and variant definitions for all math "alphabets;"
b. list of canonical and compatibility equivalents for existing math characters
for February meeting

Action item 78-42 for Aliprand:
Put math variant tags on agenda for February meeting as an "if time available" item.

CJK Radicals Supplement

[Document L2/98-415]

Jenkins: These characters violate the Unihan encoding rules, and explicitly have duplicate glyphs, for example, the 3 vs. 4 stroke forms of "grass" radical. There is no single font which has both forms of grass radical in it.

Kobayashi: The purpose of the radical supplement is different than for other ideographs. It is for searching and indexing. The IRG explicitly did not want to unify the different forms for that reason.. The issues John raised are all understood. I don't mind that L2 disagrees with the PDAM, but UTC should not disagree.
Hiura agreed with this statement.
Jenkins: KangXi radicals are universally understood as a standard set of radicals and they cannot be extended. No justification provided in the PDAM.
Whistler: There was never any justifying document explaining why IRG has gone from the small set of radicals to this large set of radicals.
McGowan: Yes. Suddenly this PDAM showed up with this extra set of radicals.
Whistler: What is the justification for the deunifications that took place?

Jenkins: We should ask the IRG to explain and justify what has been done. The bone radical deunification really bothers me.
Freytag asked when L2's vote is due on the ballot. Winkler replied that it was not until after the IRG meeting. Freytag suggested that L2 could vote no unless we get justification, and when we get it change our vote.
Hiura explained that the KangXi radicals are for searching and unification, but the IRG wants the supplement to allow searching for particular radicals.
Kobayashi: Radicals are used to characterize Han characters. There will be one radical for water. However in characters, to search them we may need variants: 3 dot "water" and full "water." For looking things up, you may need to look up water by its different radicals.

Jenkins agreed, but questioned the forms of the "grass" radical. When do you want to distinguish the 3 and 4 stroke form of Grass?
McGowan: The problem is that many of these are already encoded. "Sanzui" (??) is not a character, even though it is encoded, its only use is as a radical.
Kobayashi pointed out that we have both a-ring and angstrom. McGowan replied that this is due to source separation.
Kobayshi: Radicals are not Han-unified ideograms. So the conflict is due to the Han unification rule.

McGowan: If it is in main URO, then it is not a character.
Jenkins: Deunifying "bone" is dangerous and explicit typographical variants like "bone" are a problem. It is the worst case of three. U+2E46 is another one. Character form U+89D2. U+2E58 is a concern, like bone. U+2E5D is a concern. There are others.
Koboyashi: U+2E3F and U+2E3E are for human searching. Japanese and Chinese use "number of strokes" to search, and U+2E3E has 3 strokes and U+2E3F has 4 strokes. Japanese dictionaries will have 2 entries for these characters, that is, they will be listed twice. It is the same issue for U+2E4C-U+2E4E. We need each of these.
Jenkins: That is exactly the evidence we need from the IRG.
Roberts: If they have the same number of strokes then would you need them, like "bone"?
Yang: The bones have different stroke counts.
Jenkins: But we have unified these. We need IRG to say explicitly that these characters should not be unified because for radicals we need to distinguish stroke counts.
McGowan: We can look at Nelsons and see that they are multiple listed.

Action item 78-43 for Jenkins: Ask IRG for formal clarification re CJK Radical Supplement.

Action item 78-51 for Jenkins: Explain differences in unification rules for CJK Radical Supplement (for Chapter 10 of Version 3.0).

Additional Arabic characters

[Document L2 98/409]

Suignard: How does the shaping engine work?
Mansour: If it is not ambiguous, you use the appropriate character. If it is ambiguous than you would use the character I propose.
Becker: Why not just code the dotless Qaf?
Mansour: I want to preserve the ambiguity.
Whistler: It is analogous to looking at old German, and an ambiguous "m", "u", "i", would you want to encode an ambiguous character? I worry about including a class of characters that represent ambiguity.
McGowan: This is old literature in which modern versions have made choices about these characters, but in encoding the old ones, we shouldn't make the choice for them.
Mansour: Without having an ambiguous symbol, I can't indicate that I don't know which character this represents.
Whistler: We could say that the dotless feh is polyvalent, and in the context of ancient Arabic then you know it is ambiguous.

Mansour: Can we add a note to the book about how to handle the ambiguous character?
Whistler: Yes. It sounds like the more conservative solution is to add the 2 dotless forms (be and qaf) and not the ambiguous one. If it isn't sufficient and we find we need it, then we can add feh-qaf later.

Non-spacing characters
Long: Can dot above and below be combined with existing characters?
Suignard: We don't have any that go across left-right and right left languages.
Mansour: It might have some value for Hebrew.
Suignard: Did you check Syriac?
Mansour: Would it be unusual to take them out of another script?
Whistler: No. That is why Hamza above, below and Madda were added to WG2. The Syriac encoding was done with the idea that combining marks from the Arabic block could be used.
Becker: Historically, we didn't encode these earlier because we were concerned for the interaction between the vowels. We added combining marks to other scripts and now we have the problem of having a number of them. We are ready now then to add these to Arabic.
Freytag: I would suggest that there are unifications with Syriac, that you were not aware of, so we ask you to come back with a revised proposal. You can also specify what notes you would like to have.
Aliprand: There is not a statement here about why the combining marks from other languages should not be used.
Mansour: I looked for precedents and I found examples both ways, I can argue either way.
e.g., Devanagari nukta, dot below. Also, they will behave differently with Arabic.
Suignard: It is annoying that "Syriac combining" has Syriac in the name.
Whistler: For positions 4, 5, 6, 7, 8 are already encoded in Syriac. There are already instances of dot above and below in Syriac.
Mansour: Looking at precedents there were cases where semantics made a difference and where they were ignored. It make sense to make these as 2 separate proposals.
Becker: Should we accept the first now?
Whistler: If we introduce decomposition, then we would have to block decomposition with respect to these characters. In accepting these we have to determine what to put in the database with respect to canonical decomposition.
Mittelstein: Why would you want to block decomposition?
Whistler: It could lead to decompositions in the future that we do not want to have.
Roberts: I thought we named the Syriac characters with Syriac in the name to prevent decomposition.
Whistler: When the Syriac authors examined Arabic, they determined that the rest of Arabic was already encoded and so they were encoded for Syriac, then Hamza above, below and Madda were added.
McGowan: We haven't seen new Arabic characters for a while, perhaps the decomposition adds more characters than simply adding the few.

Action item 78-44 for Mansour & Davis: Split Arabic script proposal into:
1.characters for ancient Arabic
2.Combining characters for Arabic. Take Syriac combining characters into account in this proposal.

Action item 78-45 for Aliprand: Put Arabic proposals on agenda for February meeting as an "if time available" item

Approval of the Minutes

McGowan move to accept the minutes of the joint meeting in July. Seconded by Suignard.
Kobayashi and Hiura pointed out that the discussion of their objections to interlinear proposal had not been included. McGowan withdrew the motion to accept the minutes.

Action item 78-46 for all: Send Aliprand personal notes re discussion of interlinear annotation for revision of Minutes.

Action item 78-47 for Aliprand: Revise Minutes on basis of personal notes.

Action item 78-48 for Aliprand: Put approval of Minutes from UTC/L2 joint meeting in July on agenda for February meeting.

Interlinear Annotation Characters

An Ad Hoc discussion on the interlinear annotation characters took place, and produced a problem statement on annotation characters.

Action item 78-49 for Hiura: Give Aliprand or Winkler copy of problem statement re annotation characters for L2 distribution.

Moved by McGowan seconded by Moore
[#78-M14]. Motion: That the UTC recommend withdrawal of the interlinear annotation characters from the PDAM for Amd. 30 pending further study.
11 for; 2 against; 1 abstention (SAP)
Motion approved.

Moved by Honomichl, seconded by Winkler
[#78-M15]. Motion: The UTC revokes acceptance of the interlinear annotation characters (part of Resolution M35.12) in the consent docket, L2/98-389.

Freytag proposed a friendly amendment to withdraw approval for all characters in PDAM for Amd. 30 pending further study. Whistler disagreed with this proposal because all the other characters have individually been looked at. Freytag responded that Soft space is a character that we had not seen before because it originated at the WG2 meeting. The Mongolian tugrik sign was a new Japanese proposal and had not been seen as a character proposal before.

Honomichl did not accept Freytag's amendment.

Vote on Motion #78-M15:
10 for; 2 against; 1 abstention (SAP)
Motion approved (by 2/3 majority, since this amends Motion #78-M1).

Moved by McGowan, seconded by Whistler
[#78-M16]. Motion: The UTC instructs the Unicode representative to NCITS/L2 to vote yes on the PDAM for Amd. 30, and support the US comments that are consistent with UTC motions on this PDAM.
7 for; 2 against; 3 abstentions (SAP, Microsoft, Justsystem); NCR, Oracle absent
Motion approved.

FRIDAY, DECEMBER 4

PRESENT: Apple Computer, Inc.; Compaq Computer Corporation; IBM Corporation; Justsystem Corporation; Microsoft Corporation; NCR Corporation; Novell, Inc.; Oracle Corporation; The Research Libraries Group, Inc.; SAP AG; Sun Microsystems, Inc.; Sybase, Inc.; Unisys Corporation; Xerox Corporation
(Total members represented:14)
Quorum = 10

NOT PRESENT (at time of roll-call): Booz, Allen, Hamilton, Inc.; Hewlett-Packard Company; Mathema Software, GmbH; Reuters, Ltd.; Silicon Graphics, Inc.
(Total not represented:5)

Script capital P

Sargent led the discussion. This symbol character is actually the script form of the lower case letter p, despite its name. It is not the same as the power set symbol. Sargent recommended we delete reference to power set function, because the glyph is wrong for that.
Freytag: Then we have a long term problem if you ever want a power set symbol
Sargent: It's not any symbol, it has a particular look
Freytag: Weierstrauss used a calligraphic lowercase p. If that is what we intend, we need to change the book to see it is a lowercase calligraphic p and not alias to Weierstrauss. Otherwise, it will look like a production error. So, 10646 has the right glyph in Annex P.
Whistler: what is the decomposition of this?
Freytag: It is not decomposable.
Whistler: It has one currently, as a script uppercase P. It should be treated as the Weierstrauss symbol, i.e., lowercase P.
We have consensus to remove the decomposition.

Action item 78-52 for Editorial Committee: Delete alternative name "= power set" from code chart entry for U+2118, SCRIPT CAPITAL P

Action item 78-53 for Whistler: Remove decomposition for U+2118 from UnicodeData

Recommendation to L2: Ask WG2 to add a note to Annex P, to say that glyph is intended to be lowercase P representing the Weierstrauss symbol, and the name is therefore contradictory.

Freytag: For German Pfennig, should we add a note to the book that the script capital M U+2133 was used for identifying the German Mark? It is just informative.

Action item 78-54 for Editorial Committee: Add alternative name to code chart entry for U+2133, SCRIPT CAPITAL M, to show that it can be used to represent the old German Mark.

Meeting adjourned.

ATTACHMENT 1

UTC #78 and L2 #175 Joint Meeting – Attendees

Tuesday, December 1, 1998

Joan Aliprand; RLG; Joan_Aliprand@notes.rlg.org
Julie Allen; Unicode, Inc.; julie@unicode.org
Joe Becker, Xerox, becker.osbunorth@xerox.com
Barbara Beeton, American Mathematical Society, bnb@ams.org
Rick Buhler, Novell, rbuhler@novell.com
Mark Davis; Unicode, mark@unicode.org
Martin Dürst, W3C, duerst@w3.org
Ed Hart, SHARE, edwin.hart@jhuapl.edu
Hideki Hiura; Sun Microsystems; hiura@eng.sun.com
Lloyd Honomichl; Novell; lloyd_honomichl@novell.com
John Jenkins, Apple, jenkins@apple.com
Wai-man Long, Compaq, longman@zk3.dec.com
Kamal Mansour, Monotype, Kamal.Mansour@monotype.com
Rick McGowan; Apple Computer; rick@unicode.org
Matthias Mittelstein, SAP, matthias.mittelstein@sap.ag.de
Lisa Moore; IBM; lisam@us.ibm.com
Julia Oesterle; Unicode, Inc.; julia@unicode.org
Gary Roberts; NCR; Gary.Roberts@SanDiego.CA.NCR.com
Murray Sargent III; Microsoft; murrays@microsoft.com
Michel Suignard; Microsoft; michelsu@microsoft.com
Tex Texin, Progress, texin@progress.com
Ken Whistler; Sybase; kenw@sybase.com
Arnold Winkler; Unisys; Arnold.Winkler@unisys.com
Jianping Yang, Oracle, jiyang@us.oracle.com
By proxy: Tatsuo L. Kobayashi; Justsystem; Tatsuo_Kobayashi@justsystem.co.jp
Apologies; V. S. Umamaheswaran; IBM; umavs@ca.ibm.com

Wednesday, December 2, 1998

As above with these differences:
Absent: John Jenkins
Additional attendees:
Tatsuo L. Kobayashi; Justsystem; Tatsuo_Kobayashi@justsystem.co.jp
Asmus Freytag, Unicode, asmusf@unicode.org
Patrick Reilly, IAC, patrick@iac.com

Thursday, December 3, 1998

Additionally:
Neil Soiffer, Wolfram Research, soiffer@wolfram.com

Friday, December 4, 1998

Additionally:
Peter Lofting, Apple, lofting@apple.com