

Doc. No.: **SC22/WG11 N371**
Date: 28 January 1994
Reply to: Ed Barkmeyer
National Institute of
Standards & Technology
Bldg 220, Room A127
Gaithersburg, MD 20899 USA
+1 (301) 975-3528

Disposition of Ballot Comments on ISO/IEC CD 11404.2, Language-independent datatypes

On the ballot for progression of (2nd) CD 11404 to DIS (SC22 N1305), comments were received from 5 National Bodies. The comments are contained in SC22 N1354 and SC22 1458.

Liaison comments from SC2 (Character Sets), contained in SC22/WG11 N358, and Liaison agreements with SC21/WG8/RPC, contained in SC22/WG11 N353 and SC22/WG11 N360, resulted in additional modifications to the document.

The following is the disposition of all formal comments on CD 11404.2:

SC22 N1354, comments from New Zealand:

New Zealand voted YES with comments. The Working Group did not accept either of the New Zealand comments, although some editorial modifications to CD 11404 resulted.

(1). Datatypes should support Undefined as a value.

Resolution: Not accepted. See Resolved Issues 17 and 18. Undefined is not a "value" of a datatype: it has none of the properties of any datatype to which it is assigned. It is rather a part of the behavior of some entities which possess the notion "datatype", and its meaning depends very much on the situation and the entity rather than the datatype. In fact, some processing entities, e.g. SQL, have more than one "undefined" value, in order to model the "reason" why no value is available.

An example of *one* method of supporting Undefined has been added to Choice (8.3.1).

(2). Abstract Syntax.

Resolution: Unresolved. The Working Group did not understand this comment and no member of the national body was present to explain it. In particular, what is meant by "abstract syntax" in this regard?

The WG postulated that "abstract syntax" refers to definition of a production language which yields "symbols" for all the values of a "type" by their relationship to each other and some fundamental values. The WG previously determined, in response to a liaison statement from WG17 (Prolog), that this was unnecessary for the simple types, which are well understood, very difficult for Real and Complex, and probably impossible for types like Procedure and Pointer.

The Working Group does not, in fact, expect that "provable equivalence" will actually exist in many language mappings to/from LI Datatypes.

SC22 N1354, comments from Sweden:

Sweden voted NO with comments, but did not specify which of these comments were required to change the vote to YES. The Working Group accepted all of the Swedish comments essentially verbatim. Although no representative from Sweden was present at the editing meeting, Sweden has subsequently confirmed that the NO vote is resolved, and is changed to YES.

1. Fix reference to ISO 8601.

Resolution: Accepted. Clause 2 corrected.

2. Align character names with ISO/IEC 10646-1.

Resolution: Accepted. Clause 4.1 corrected. A number of additional corrections to clause 4.1 resulted from discussion of this comment and the SC2 liaison contribution.

- (3). Correct references to ISO/IEC 10646-1.

Resolution: Accepted. All occurrences corrected.

- (4). Correct reference to Latin alphabet.

Resolution: Accepted. Text in 8.1.4 corrected.

- (5). Correct value-syntax for Time types to align with ISO 8601.

Resolution: Accepted. Syntax and text in 8.1.6 modified.

SC22 N1354, comments from United Kingdom:

The United Kingdom voted YES with comments. The Working Group accepted all of the UK comments.

1. Modify definition(s) of datatype.

Resolution: Accepted in principle. Clauses 4.11 and 6.1 are now aligned and the definition reworded in the spirit of the proposed text.

2. Add a Note to 5.2 for conformance of programming language standards.

Resolution: Accepted.

3. Clarify that Character does not refer to codes.

Resolution: Accepted in principle. The Example in 8.1.4 was reworded, but the text was from the SC2 liaison.

4. Produce an additional mapping annex.

Resolution: Accepted. UK was asked to provide the text of the annex for the DIS.

5. Create SC22 Ad Hoc meeting to discuss LID with PL standards.

Resolution: No change to the document. Convenor accepts the recommendation as an action item for the Convenor's report.

SC22 N1354, comments from the United States:

The United States voted NO with comments. Comments 1 and 2 were identified as required in order to change the US vote to YES. The Working Group accepted Comment 2. After considerable discussion, the US withdrew Comment 1. Most, but not all, of the other US comments were accepted. The US delegate confirmed that the US vote is changed to YES.

1. Merge Sequence and Array.

Resolution: Withdrawn. Discussion of this comment made it clear that the real issue was whether the “aggregate properties” of aggregate datatypes had any normative force. The decision was made that they have none. A Note to that effect was added to Clause 11, and the “property” lists for aggregate type-generators were separated, in order to clarify the distinction in what must be supported.

2. Redefine Table to support SQL.

Resolution: Accepted. Text of 8.4.6 replaced. [Editor retained some text from WD7 8.4.6 and copied additional text from Record (8.4.1) to complete the type definition, as the U.S. text was incomplete.]

3. Clarify multiple character repertoires.

Resolution: Accepted.

4. Make Ordinal a defined-type.

Resolution: Rejected. Ordinal is conceptually different from Integer in important ways - they have nearly no common characterizing operations. The issue of ambiguous value syntax is irrelevant, since the IDN ensures that the type of a value must be known before the value syntax is encountered.

5. Make Bit a defined-type.

Resolution: Accepted. Clause 8.1.7 modified to 10.1.2.

6. Modify characterizing operations for Choice.

Resolution: Accepted, with minor editorial changes.

7. Add a Note on variables to Pointer.

Resolution: Accepted.

8. Delete Switch.

Resolution: Accepted in principle. Clause 10.1.1 deleted. The datatype Switch is defined as an Example under State (8.1.2).

9. Rename Cardinal NaturalNumber.

Resolution: Accepted. Clause 10.1.1 modified.

10. Add note to CharacterString.

Resolution: Accepted, with additions.

11. Remove Currency.

Resolution: Accepted, with editorial changes. Clause 10.1.6 deleted, note/example modified in 8.1.9.

12. Rename Interval to TimeInterval.

Resolution: Accepted. Clause 10.1.6 modified.

13. Delete DistinguishedName.

Resolution: Accepted. Clause 10.1.11 deleted.

14. Remove Cyclic-enumerated.

Resolution: Rejected. This type-generator is considered by the UK to be useful to applications of LI Datatypes.

SC22 N1458, comments from Canada:

Canada voted NO with three comments, all of which were required to change the Canadian vote to YES. The Working Group accepted Comment 1, but did not accept Comments 2 and 3. The resolution of Comment 2 was accepted by Canada, but the resolution of Comment 3 was NOT. Canada's vote remains NO.

1. Make CharacterString unordered and modify Annex A.

Resolution: Accepted. Clause 10.1.4 corrected. Annex A has been replaced by a text offered by the SC2 Liaison.

2. Specify that CharacterString ordering is to be handled by Operating System services (i.e. POSIX).

Resolution: Rejected. How "user-imposed" ordering of an unordered LI Datatype may be imposed is out of the scope of DIS 11404. (Canada accepts this resolution.)

3. Add a "ProcessedString" datatype implementing a SHARE recommendation.

Resolution: Not accepted. After extensive deliberations (including an exchange of views with the proposers of the datatype) WG11 has decided NOT to include the ProcessedString datatype in CD 11404. The reasons for this decision are as follows:

a) Incomplete specification of the requested datatype.

The following documents were mentioned as base documents for the required datatype:

- SHARE - Europe: White Paper on National Language Architecture (1990)
- CSA Standard Z243.4.1-1992, Canadian Alphanumeric Ordering Standard for Character Sets.

A detailed analysis of these documents showed the following:

1. The SHARE document describing the "functionality" is aimed at rendering foreign language characters into EBCDIC (and ISO 646) in such a way that transliterations can be supported, collating sequences can be defined, and orthographic conventions can be selectively used or ignored in string comparisons (and presumably other processing). It has examples, but does not *define* a complete form for *any* language. This document is a technical report, identifying the things which must be considered, and a consistent approach, in developing some elements of what have become POSIX LOCALE files.

2. The cited Canadian standard defines the complete information needed for the development of a ProcessedString datatype for French and English. Additional facilities, however, would be needed to accommodate other European languages, such as German, Swedish, Spanish and Dutch, while a significantly different set of facilities would be needed for Russian, Greek, Hebrew and Arabic. And it is not clear that the SHARE approach can be extended to Chinese and Japanese at all.

Thus, the cited references do not readily lead to the formulation of a datatype which extends to languages outside of Western Europe. The Working Group has therefore come to the conclusion that if LID is to contain such a datatype, it must define the components thereof in such a way as to be useful to more than a few National Bodies. But, extension to other languages (ideally to all languages covered by ISO 10646) requires additional work, which should be the responsibility of WG20.

b) Insufficient justification for inclusion.

Although it is recognized that “something like the ProcessedString datatype” could be useful for certain fields of application (like the POSIX Locale mechanism), this in itself is not enough justification for inclusion of the datatype in the LID standard.

Moreover, the existence of ISO 10646 (which postdates the SHARE document) guarantees that there is an international standard character set, and therefore a LI Character-String datatype, in which all of the information contained in the ProcessedString values can be exchanged. While there is value added in processing the string into “textual components”, it appears that that processing is dependent on the natural language and possibly the application.

The Canadian proposal seems to be asking for a combination of a number of character strings, with a certain semantic relationship between these strings. Such a datatype can easily be defined based on the concepts already available in LID. This should be done by the writers of the interface by which the information conforming to this datatype should be passed (this is similar to a language independent definition of the FILE structure from the C standard).

For the reasons mentioned above, the Working Group considers it inappropriate to include the requested datatype in the document at this moment. It is also recognized that when a complete proposal, based on consensus among the various technical expert groups involved, were forwarded to WG11, this could be included in a later revision of the standard.

Canada accepts the recommendation that further work should be done, but insists that the requested addition to CD 11404 is needed *now*, in order to support the cited Canadian national standard and to support development of language-independent interfaces utilizing features of POSIX. Canada *does not accept* the WG11 resolution! Canada’s vote remains NO!

SC22/WG11 N358, Comments from J.W. van Wingen, SC2 Liaison

4.1 Clarify Metacharacters.

Resolution: accepted. Table 4-1 replaced.

7.1 ‘’ is ambiguous.

Resolution: not accepted. Because apostrophes must enclose a SINGLE character, ‘’ unambiguously represents an APOSTROPHE character. A note is added to 8.1.4.

10.1.4 Allow “” within a string value to represent a single QUOTATION MARK.

Resolution: rejected. Representation of character-string values in the IDN is a rare occurrence, almost irrelevant to LI Datatypes. Few users would benefit from the syntactic extension, which is purely a convenience.

10.1.8 Octet should be ordered.

Resolution: accepted. Octet is redefined as a subtype of Integer.

(10.1.8) Add OctetString.

Resolution: accepted. OctetString added to clause 10.1.

Annex A. Replace annex A.

Resolution: accepted. Annex A replaced by the offered text. WG11 thanks the SC2 liaison for this contribution.