



ISO / IEC JTC1 / SC22
Programming languages, their environments and system software interfaces
Secretariat: CANADA (SCC)

ISO/IEC JTC1/SC22
N 1572

MARCH 1994

TITLE: Summary of Voting and comments received on CD 13886:
Language independent procedure calling

SOURCE: Secretariat ISO/IEC JTC1/SC22

WORK ITEM: JTC1.22.16

STATUS: New

CROSS REFERENCE: N1507

DOCUMENT TYPE: Summary of Voting - CD

ACTION: For information to SC22 Member Bodies.
See attached.

Address reply to: ISO/IEC JTC1/SC22 Secretariat
J.L. Côté
Treasury Board Secretariat
300 Laurier Ave. West, 10th Floor, Ottawa, Ontario, Canada K1A 0R5
Tel.: (613)957-2496 Telex: 053-3336 Fax: (613)957-8700

SUMMARY OF VOTING ON:

Letter Ballot Reference No: SC22 N1507
Circulated by :JTC1/SC22
Circulation Date :1993-11-26
Closing Date :1994-03-10

SUBJECT: CD 13886: Language independent procedure calling

The following responses have been received:

'P' Members supporting proposal, without comments	: 06
'P' Members supporting proposal, with comments	: 03
'P' Members not supporting proposal,	: 01
'P' Members abstaining	: 03
'P' Members not voting	: 10

Secretariat Action:

The Secretariat of JTC1/SC22 will forward the comments to WG11 for recommendation on further processing of CD 13886.

ISO/IEC JTC1/SC22 LETTER BALLOT SUMMARY

PROJECT NO: JTC1.22.16

SUBJECT: CD 13886: Language-independent procedure calling

Reference Document No: N1507
Circulation Date: 1993-11-26

Ballot Document No: N1507
Closing Date: 1994-03-10

Circulated To: SC22 P, L

Circulated By: Secretariat

SUMMARY OF VOTING AND COMMENTS RECEIVED

	Approve	Disapprove	Abstain	Comments	Not Voting
'P' Members					
Australia	()	()	()	()	
Austria	()	()	()	()	
Belgium	(x)	()	()	()	
Brazil	()	()	()	()	
Bulgaria	()	()	()	()	
Canada	()	()	(x)	()	
China	()	()	()	()	
Denmark	(x)	()	()	()	
Finland	()	()	()	()	
France	()	()	()	()	
Germany	()	()	(x)	()	
Greece	()	()	()	()	
Italy	()	()	(x)	()	
Japan	(x)	()	()	()	
Netherlands	(x)	()	()	()	
New Zealand	(x)	()	()	(x)	
Slovenia	()	()	()	(x)	
Sweden	(x)	()	()	()	
Switzerland	(x)	()	()	()	
UK	()	(x)	()	(x)	
Ukraine	(x)	()	()	(x)	
USA	(x)	()	()	()	
'O' Members					
Argentina	()	()	()	()	()
Cuba	()	()	()	()	()
Czech/Slovak Re	(x)	()	()	()	()
Hungary	()	()	()	()	()
Iceland	()	()	()	()	()
India	()	()	()	()	()
Poland	(x)	()	()	()	()
Portugal	()	()	()	()	()
Romania	()	()	()	()	()
Singapore	()	()	()	()	()
Turkey	()	()	()	()	()
Thailand	()	()	()	()	()

NNI comment, accompanying the YES vote on CD 13886 (SC22/N1507):

In section 5.2.1. of CD 13886, the following parameter passing methods are described:

- Call by Value Sent on Initiation
- Call by Value Sent on Request
- Call by Value Returned on Termination
- Call by Value Returned when Available

The first 2 are about input parameters, the last two are about output parameters. The first and the third methods are clearly each others counterparts, and likewise the second and fourth. However, the latter only superficially: the description of 'Call by Value Returned when Available' explicitly allows that a value can be returned more than once (see the note), while this is not mentioned in the description of 'Call by Value Sent on Request'. The 3rd note even suggests that this is NOT allowed.

We believe that this is not the intention; see also note 2 with a perfect example of a multiple passing of the parameter.

Therefore, 'Call by Value Sent on Request' should permit multiple requests for the value of the same parameter. The notes in 5.2.1.2 and 5.2.1.4 need to be corrected.

New Zealand Comments on ISO/IEC JTC1/SC22/N1507

Section 1, last 'bullet'

Typographic error – "... a parameter."

Section 2

Reference to DIS 11404 should be corrected.

Section 3.1

This definition is surely incorrect. We suggest that it would better read as "*A value that is bound to a formal parameter symbol during the execution of a procedure.*" – or something equivalent referring to the binding involved.

Section 3.2

The use of the word *via* is not understood in this definition, since the general definition of that word has the sense of 'through' ie either *through the agency of something* or *through the location of something*. Neither of these seems relevant to this context. It is suggested, therefore, that the following wording might be close to the intended meaning – "*A value passed between client and server procedures during the period of time involved in the calling and elaboration of a server procedure*".

Section 3.8, 3.32 & 3.34

The definition given in 3.34 indicates that the global symbols are mapped – and so does the definition in 3.8. The need for the definition in 3.8 is therefore not understood. Given the definition in 3.34 then the definition in 3.32 is in error since to be a procedure closure (according to 3.34) the global symbols are mapped, so while the entity defined in 3.32 may well be a 'partial' something it cannot be any kind of procedure closure (because that defines the global mapping!). In none of these subsections is there any indication to what the global symbols are mapped (to be fully defined a mapping must surely have both domain and range!). Neither is there a definition of the term 'global symbol', which is surely necessary in a client/server context. Something which is global to the server may be visible to the client and hence be jointly global. However, an entity which is global to the client which is the program may not be accessible to the server although it is still global to every part of the client. May we please request that these terms be formally and unambiguously defined before further voting takes place on a revised version as DIS?

Section 3.10

The definition of this term is both confusing and possibly malformed. We understand that sequential execution of a program starts from an initial state and continues with a succession of states occurring one after the other such that each state is distinguishable from the state immediately before it by either the creation of a new component or the alteration of an existing state component. If this understanding conforms with the understanding of WG11 then we are worried by the use of the word *series* – which has no necessary temporal significance – and the absence of the word *program*, which we believe is necessary to indicate of what the execution sequence forms a part (it otherwise has an anonymous referent which is unwise in a formal definition).

Section 3.11

The word *identification* used here implies an action, which is surely not the case. We suggest that the definition could better be worded as "*The name symbol of a parameter used in the definition of a procedure – to which a value will be bound during execution*". Note that we have a slight preference for the term *instantiation* of a procedure rather than *execution*, but that this has been eschewed elsewhere in this document.

Section 3.12

It is suggested that this definition is backwards and would read better as the following – "*The*

Schweizerische Normen-Vereinigung
 Association Suisse de Normalisation
 Swiss Association for Standardization

Telefax

national (CH) 01 254 54 74
 international +41 1 254 54 74



Mühlebachstrasse 54
 CH-8008 Zurich

Telefon
 national (CH) 01 254 54 54
 international +41 1 254 54 54

Teletex
 228-47935131 = SNV

Telex
 755931 snv ch

Empfänger ISO/IEC JTC 1/SC 22 Secretariat
 Adresssee J.L. Côté
 Treasury Board Secretariat
 Telefax-Nr. 001 613 957 8700
 Betrifft ISO/IEC CD 13886
 Subject

Datum 1994-02-21
 Date

Seite 1 von 2
 Page 1 of 2

Absender P. Scheibli/fa
 Sender

Interface Definition Notation (IDN) defined in the Remote Procedure Calling standard (ISO/IEC 11578-2.2). The contents of Annex C to this international standard is a sub-set of the notation defined there."

Section 3.13

It had been thought from earlier definitions that the term *global state* referred to the actual state occurring during the execution of some program. The definition given here, however, refers to the term *boxes* – which are defined as models and therefore are abstract rather than real. The phrase referring to currently assigned values, however, only serves to confirm the earlier understanding. There is therefore a confusion of terms since a box is defined as being virtual while a state is supposedly real. This needs clarifying and we are unable to offer a suggestion.

Section 3.14

Surely the important point about the definition of this term is that the definition must be provided by the implementer. It is suggested therefore that the definition should be reworded to read "*An entity which is indicated in this international standard to be implementation-defined may differ between processors, but the definition for any conforming processor shall be provided by the implementer of that processor.*"

Section 3.16 & 3.17

The phrase *on entry* in both of these sub-sections is not understood since it may ambiguously refer to several points of entry (the client, the server, the communications port, the interface, the protocol marshaller, etc). Could this please be clarified.

Section 3.18

Surely the interface closure is just the mapping from the domain of procedure names to the range of closures provided? Why, therefore, is this defined as being two 'collection's AND a mapping? This is quite confusing.

Section 3.19 & 3.24

Reworded along the lines suggested for 3.18 the definition in 3.24 seems to be what we would understand by the term – informally just some sort of 'record of procedures' of which a closure is the instantiation. The duplicate in 3.19 seems to be wrong.

Section 3.20

The parenthesised '(A)' on the second line seems to be a typographical error. The last sentence should be reworded as "... procedure image to values for ...".

Section 3.29

Typographic error - delete the space before the comma.

Section 3.30

The use of the term 'some scope' is not incorrect, but the term *scope* itself could well need definition since in a client/server situation where multiple independently generated name spaces are in existence, some clarification may be in order.

Section 3.43

Surely the form in which a language is submitted to a language processor need not bear any relation to what is conventionally thought of as text. The use of this term is therefore strongly discouraged (see later comments) and it could well be omitted from this section.

Section 3.46

This definition refers to program text. It is suggested therefore that it be reworded as – "*A programmer defined syntactic entity.*" This does not need to refer to text, nor to the (we believe) erroneous use of the term *box* (which is a model and as it is understood a concept in this standard and

not in a programming language standard necessarily. Providing this more generic definition avoids both potential traps.

Section 4.1

The term '*written in ... language*' appears several times. It is suggested that this be replaced by wording similar to "*generated for ... language processor*" – as may be appropriate to the textual constructs involved.

Section 5.1, last para

The word '*transmittable*' is not English. Elsewhere the term '*transmissible*' is (correctly) used.

Section 5.2, first para

This reads in part "... *an actual parameter may be an expression ...*". However this conflicts with the definition of the term actual parameter (in 3.1) as just 'a value'. The sentence should therefore be reworded to read "*Therefore an actual parameter is a value of ...*".

Section 5.3.1.2, 2nd 'bullet'

This ends "... *as in case one.*" However, nothing is earlier identified as being 'case one'. Does this mean "... *as in the above case.*"?

Section 6.2, para 1

Once again the term box is confirmed as being a model -- is it is virtual rather than real. May we suggest that the complete text be checked for text where something real is meant and where something virtual or abstract is meant and different terminology be employed to avoid confusing the reader? Incidentally the second word of the third sentence should be "*may*" and not "*many*".

Section 6.2, 2nd para

This ends "... *stipulated.*". The OED gives the word '*stipulated*' the sense of prior contract, but we are doubtful if this is intended here. Perhaps some other word should be used to better give the intended meaning.

Section 6.2, 4th para

If an execution sequence (given its revised definition suggested is strictly temporal), then why is the word '*point*' used – which is expressive of spatial location? May we suggest replacing it by "*instant*". Note that the second sentence is unnecessary sine the definition is given in 3.10.

Section 6.3, 2nd 'bullet'

This should read "... are created during ...".

Section 6.4, last para

The word '*spec*' in the second sentence should be in the italic font.

Section 6.9, last word

This should be "*sub-section*"?

Section 6.13.2 Note

Why is the word '*AND*' on the second line in upper case – it looks rather odd!

Section 6.15, p22, para 1

This should end as "... *different effects from calling C1.*".

Section 7.1.1

Is the definition of '*hyphen*' a typographic error – in our copy it appears to be a low-line.

Section 7.1.1

Please correct the references throughout to ISO/IEC 10646-1:1993.

Section 7.8

Would it not be better to use the now standard name of 'low-line' rather than 'underline'?

Annex D

Is this intentionally still blank? An informative example could be useful to implementers.



VOTE ON COMMITTEE DRAFT	
Date of circulation 1993-11-24	Reference number 13886
Closing date for voting 1994-03-09	ISO/TC 1 / SC 22 N 1507

ISO/TC JTC 1 / SC 22 Title Programming Languages, their environments and system software interfaces Secretariat Canada, SCC

Circulated to P-members of the committee for voting on registration of the draft as a DIS, in accordance with 2.5.7 of part 1 of the ISO/IEC Directives

Please send this form, duly completed, to the Secretariat indicated above.
P-members of the technical committee or subcommittee concerned have an obligation to vote.

CD 13886 Title Information Technology - Programming Languages, their environments and system software interfaces - Language-independent procedure calling

- We agree to the circulation of the draft as a DIS in accordance with 2.6.1 of part 1 of the ISO/IEC Directives
- with comments (editorial or other) appended

We do not agree to the circulation of the draft as a DIS

The reasons for our disagreement are the following (use a separate page as annex, if necessary)

At least one sample binding is required. See comment-S2 in attached comments.

We abstain

P-member voting Date 94-02-05	Signature C M Jenkins
-------------------------------------	--------------------------

The UK votes NO vote CD 13886 Language-independent Procedure Calling with a number of comments. This vote will be changed to YES if the Annex noted in comment 52 is provided.

1. Section 3 should include definitions for "global data" (5.2.2) and "global symbol" and the definition of LID should refer to ISO/IEC 11404.
2. section 3: since other quite basic terms are defined, we would have expected to find 'synchronous' and 'asynchronous' also (used in e.g. 5.3.1.1).
3. 3.2 parameter is out of place: all other items are in alphabetical order.
4. In 3.34 add "procedure" before "image" for clarification purposes
5. In 4.1 in 1. and 2. delete the reference to 4.1.1 as this is not part of the model
6. In general throughout the document reference to "LIPC" should be to "this International Standard" except when specifically needed, e.g. in the first sentence of 4.
7. Add "of procedure calling" to the title of Clause 5 in order to emphasise the difference with clause 6. [It would also be helpful if a paragraph could be inserted describing the purpose of the clause, c.f. clause 6.]
8. In 5.1 para 4 refs to 6.4 should be 5.2 and 6.4.6 to 5.2.2, 6.4.7 to 5.2.3
9. In 5.2 first line replace "International...Datatypes" with "LID"
10. In 5.2 in the para starting "Where the..." add after ""write once only"" "as far as the server procedure is concerned" for clarification as to which procedure invocation has the restriction
11. 5.2 'Weak typing' is covered in a cursory manner out of keeping with the rest of the document.
12. In 5.2.1.3 in the 4th para "Note: (1)" replace "(hardware" with "(perhaps even the hardware" for clarification purposes
13. 5.2.1.3 "know" should be "known"
14. In 5.2.13 in the 5th para replace "as Specified" with "on Termination" as this seems to be an inadvertent omission from the previous editing of the document
15. In 5.2.2 line 1 delete "units" as this seems to add nothing to the term "data" and may cause confusion since it is not defined in the document. Also a definition of "shared execution environment context" might be helpful.
16. In 5.2.2 sentence 4 replace "from a procedure" by "used by a procedure from another" for clarification.
17. In 5.2.5 sentence 2 replace "protected parameter" with "parameter of a private type" as there is no definition of the term "protected parameter".
18. Should the word restricted in the next sentence be in capitals in order to distinguish it from other terms being used?
19. In 5.3.1 the term "should not be completely ignored" appears to be very loose indeed and weaken what is intended. One possibility for replacement is "must not be ignored". Strengthening this is very important.
20. In 5.3 the definition Normal Termination (5.3.1.3) should precede those of Cancellation and Abnormal, in order to aid understanding.

21. In 5.3.1.1 the heading should be "Cancellation Termination" and it should start "A procedure terminating by cancellation means the issuing of a command..." in order to be consistent with 5.3.1.2 and 5.3.1.3
22. In 5.3.1.2 sentence 3 replace "terminations" with "termination" as the mechanism is singular.
23. In 5.3.1.2 in the last but one sentence replace "the International" with "this International"
24. 5.4.2 "needs" should be "need"
25. One of our panel members comments: What difference would it make to the effect of the standard if the whole of section 6 were to be deleted? The reason for section 6 needs to be stated or the section omitted.
26. In 6.1 the term Value is defined. Unfortunately this may conflict with the usage of value in potential readers minds. A different term might avoid confusion.
27. In 6.2 in the 1st sentence replace "model of a variable or" with "generic term for a" and add to the end of the sentence ", for example in some contexts what would be a ""variable""." In some readers minds models by definition cannot exist and therefore confusion would be introduced by the next sentence.
28. In 6.2 in the 2nd sentence replace "many" by "may"
29. In 6.2 a notation for operation description is introduced, but not explained. A definition of it should be placed in clause 3.
30. In 6.2 the para starting "The operation Create" replace the last words "not stipulated" with "indeterminate" or "unspecified" as the original term is not normal standardese .
31. In 6.2 the term "global state" is defined. This does not appear to be a good term, when compared with the definition of "global data". An alternative is requested.
32. 6.2 "write: Box x value ->": is the 'x' a typo?
33. In 6.3 the first sub-para refers Global symbols to refer to values that exist prior to invocation. In some languages it is possible to create "variables" which are not known prior to entry to the procedure but are known after it has terminated, either to the client procedure or as persistent entities (for example a new database file) which are known to concurrent and subsequent processes. Does the definition of this and global entities allow for this possibility?
34. In 6.2 the second sub-para delete "at"
35. In 6.4 in the para "Gsym..." the term "spec" should be in italics as are the other operations
36. In 6.6 the 1st para replace "<image,association>" with "<I,A>", "where" with "where A is", "image's" with "image I's" and "others" with "other symbols" for clarification.
37. In 6.6 the 2nd para insert "image's" after "whose" and "by the association" after "mapped" for clarification
38. In 6.6 the 3rd para insert "image's" after "of the" and "by the association" after "mapped"
39. In 6.10 1st para insert "the operation" after "defines" for clarification.
40. In 6.10 should spec also be in italics as in 6.4?
41. In 6.10 last para replace "function" with "operation" and "to" with "and"
42. In 6.11 1st para insert "procedure" after "Basic"

43. Without having an electronic version of the document it is difficult to be certain, but it seems as if the definition of IAssoc and GIAssoc are not used elsewhere in the document. Should they not be deleted? If they remain should they not be described as "operations" rather than "functions" for consistency?
44. In 6.15 para 5 the terms "forward" and "backward" are used without any definitions.
45. In 6.16 the terms "congruent" and "sharing structure" are used, again without any definition.
46. In 6.16 operation 1. of TF is Assoc(V) intersect Boxes. Should this "intersect" be a mathematical symbol?
47. 6.16 We dislike using "complex" in its ordinary English sense in essentially a mathematical context - it is ambiguous at first sight. In fact much of 6.16 (especially the bit about the identity transformation) reads like a text book and we are not sure what it is adding to the standard.
48. In 7.1.1 the definition of "added-character" is unclear. Is it intended to mean those additional characters in 10646 which are present in the implementation character-set?
49. 7.6.1.11 "and additive" should read "the additive".
50. Heading A.1 in appendix B should be "B.1". This is a pretty important point but is baldly stated with no examples or description.
51. Appendix C & section 7: Some numbering of syntax rules would make cross-referencing, and perhaps consistency checking, much easier.
52. It was a condition of our previous vote that further progress of this document should not be allowed without the provision of at least one sample binding, preferably two. This needs to be provided in Annex D.
53. The names in some programming languages are case-sensitive and in others are not. How LIPC addresses this issue, presumably in some mapping, should be mentioned in order to avoid confusing some readers.