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

ISO/IEC JTC1/SC22
N 1506

NOVEMBER 1993

TITLE:	WG11's Response to comments received on the CD registreation ballot for Language-independent procedure calling
SOURCE:	Secretariat ISO/IEC JTC1/SC22
WORK ITEM:	JTC1.22.16
STATUS:	New
CROSS REFERENCE:	N1289, N1347
DOCUMENT TYPE:	Response to comments
ACTION:	For information to SC22 Member Bodies.

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

Response to Comments on LIPC CD Registration Ballot (SC22/N1347)
November 4, 1993

- New Zealand

  1. Accepted: Clause 7.1 in the LID will be included in clause 4.1 of the LIPC.
  2. Accepted: Text for the development of "spec" will be clarified. P and I will also be defined in clause 6.13.
  3. Accepted: The absence of termination declarations is indeed confusing. Termination declarations will be added to clause 6.14.
  4. Accepted: Example will be rewritten.
  5. Accepted: Every standard normatively referenced in the LIPC should be included in clause 2. The paragraph in clause 7.1.5.1.3 has been rewritten to correct an error.
  6. Accepted: Clause 7.1.10 is being deleted.

- United Kingdom

  1. Accepted: Copy syntax and "Description" paragraphs from LID. Point to LID for complete formal definition. E.g.

```
7.1.5.1.1 Integer
Integer is the mathematical datatype comprising the exact
integral values.  Syntax:
integer-type = "integer"  .
The interpretation of the syntax is formally defined in DIS 11404.

Special cases:

7.1.5.1.7 Procedure-type
Copy and edit ALL of LID 8.3.3 to make the terminology match LIPC
clause 6.  (Ignore the Notes, unless you think one of them is important.)
The definition of "Values", for example, should begin:  The values of a
procedure-type are procedure closures, as defined in 6.6 ...

7.1.5.2.4 Pointer
"Pointer is a type-generator which generates a primitive datatype
each of whose values constitutes a means of reference to values of another
datatype, designated the \element\ datatype.  The values of a pointer
datatype are boxes, as defined in 6.8."

Retain the definition of "restricted" and strike the rest of the
existing text (from RPC).  Insert the text on aliasing. (see US-69).

7.1.6 Procedure declaration
"A procedure-declaration associates one name with a procedure-type
(see 7.1.5.1.7), as part of the interface-type association (see 6.12).
The procedure type is ... <copy LID Value syntax for procedure-type>"
Strike the rest of the current text.
```

Editorial NOTE: All of Clause 7 is in 7.1. Eliminate the 7.1 heading and move all subclauses up one level.

2. Accepted: It is SC22/WG11's intent to have aligned IDN between the LIPC, LID, and RPC standard at the final IS stage.

3. Accepted: "Global IDN" will be defined.

4. Accepted: An informative annex with example binding to the LIPC will be included. It is not likely that the example bindings will be completed in time for the first CD.

5. Accepted: A number a areas in the text have been identified as being too informal. This text will be rewritten in a more formal style. Also, explanatory text will be included in notes as identified.

- United States

1. Rejected: The proposed change does not help improve the clarity of the document.

2. Accepted: "-- >" and "- >" have been replaced by a proper right arrow.

3. Accepted: Definition of "program text" has been added to clause 3.

4. Accepted: Definitions for "partial procedure closure" and "complete procedure closure" have been added to clause 3.

5. Accepted: Definition of "name" has been added to clause 3.

6. Accepted: Occurrences of "run" have been replace by appropriate variant of "execute" throughout the document.

7. Accepted: The last sentence of clause 6.1 has been changed as requested.

8. Accepted: The title of clause 6.2 has been changed to "Boxes and global state".

9. Accepted: The two paragraphs following the first note in clause 6.2 has been replaced as with the suggested text.

10. Accepted: The first note has been removed from clause 6.2.

11. Accepted: A note has been added to the end of clause 6.2 as suggested.

12. Accepted: The last sentence of clause 6.2 has been removed.

13. Accepted: Clause 6.3 has been rewritten.

14. Accepted: Clause 6.3 has been rewritten.

15. Accepted: Clause 6.3 has been rewritten.

16. Accepted: Clause 6.3 has been rewritten.

17. Accepted: A new note has been added to the end of the new clause 6.3.

18. Accepted: "Note. How references to values in a program text in a particular language are expressed is defined by the rules of the language, including its scoping rules. For example, the means of reference may be an identifier and the same identifier may relate to two different references in different program contexts (because of scoping rules). The identifier would thus correspond to two different "symbols" in the sense of this subclause."

19. Accepted: "A procedure image is the abstraction of a procedure text. Implicit in a procedure image is the procedure-type, the global, local and parameter symbols used within the procedure text, and the algorithm to be executed by the language processor."

20. Accepted: A note has been added after the operation definitions as suggested in clause 6.4.

21. Rejected: Argument symbols have been renamed to parameter symbols. The notion of distinguishing between input and output parameter symbols does not improve clarity.

22. Rejected: Argument symbols have been renamed to parameter symbols. The notion of distinguishing between input and output parameter symbols does not improve clarity.

23. Accepted: The last paragraph of clause 6.4 has been changed to a note as suggested.

24. Accepted: In clause 6.5, x- > y has been changed as suggested.

25. Accepted: "dom" and "rng" have been expanded.

26. Accepted: Last sentence of the last paragraph in clause 6.6 has been removed.

27. Accepted: Last paragraph of clause 6.6 has been changed to a note.

28. Rejected: No new clause is necessary

29. Accepted: "activated" and its variants have been changed to "invoked" in clause 6.6.

30. Accepted: Definition of complete procedure closures added to clause 6.6.

31. Accepted: Paragraph two in clause 6.6 has been changed as suggested.

32. Accepted: First paragraph in clause 6.7 has been changed as suggested.

33. Accepted: I and A have been expanded throughout clause 6.7

34. Accepted: Sentence before bullets in clause 6.7 has been modified. Bullets have been removed from the end of clause 6.7.

35. Accepted: Last sentence of clause 6.7 has been changed to a note.

36. Accepted: Fourth sentence of first paragraph in clause 6.7 has been changed as suggested.

37. Accepted: Additional sentence has been added to clause 6.7 as suggested.

38. Rejected: Previous changes dealing with input parameters were not accepted.

39. Accepted: Sections 6.8, 6.9, and 6.10 have had relevant parts extracted and these sections have been combined into one new clause entitled "Boxes, pointers, values, and datatypes."

40. Accepted: Sections 6.8, 6.9, and 6.10 have had relevant parts extracted and these sections have been combined into one new clause entitled "Boxes, pointers, values, and datatypes."

41. Accepted: Sections 6.8, 6.9, and 6.10 have had relevant parts extracted and these sections have been combined into one new clause entitled "Boxes, pointers, values, and datatypes."

42. Accepted: Sections 6.8, 6.9, and 6.10 have had relevant parts extracted and these sections have been combined into one new clause entitled "Boxes, pointers, values, and datatypes."

43. Accepted: Sections 6.8, 6.9, and 6.10 have had relevant parts extracted and these sections have been combined into one new clause entitled "Boxes, pointers, values, and datatypes."

44. Accepted: First paragraph of clause 6.11 has been rewritten with a slight modification to the original suggestion. The definitions of interface closure has also been updated in clause 3.

45. Accepted: First paragraph of clause 6.12 has been rewritten with a slight modification to the original suggestion. The definitions of interface closure has also been updated in clause 3.

46. Accepted: Wording of the example in clause 6.12 has been modified.

47. Rejected: No change is necessary.

48. Accepted: Suggested wording changes in clause 6.14 have been made.

49. Accepted: The suggested word has been added to clause 6.14.

50. Accepted: The suggested line in clause 6.14 has been moved.

51. Accepted: Suggested wording changes in clause 6.14 have been made.

52. Accepted: Based on this comment, many additional change to the wording in clause 6.14 were made.

53. Accepted: Clause 6.7 "Basic procedure invocation" has been moved immediately before clause 6.14.

54. Accepted: Clause 6.15 has been broken into subclauses.

55. Accepted: Clause 6.15 has been modified to eliminate the If clauses.

56. Accepted: In clause 6.15 "union" has been changed.

57. Rejected: Proposed wording is not clear.

58. Accepted: The first sentence of the referenced paragraph in clause 6.15 contains normative information and will not be deleted. The rest of the paragraph has been removed.

59. Accepted: Last paragraph within clause 6.15 has been changed as suggested.

60. Accepted: Last paragraph of clause 6.15 has been changed as suggested.

61. Accepted: "3. J is in the range of the invocation association of <I,A> and B can be constructed from accessible values."

62. Accepted: Suggested change has been applied to the text.

63. Accepted: Certain sections have been moved in order to help with the organization and flow of the document. Further refinements may be necessary.

64. Accepted: See response to US-63

65. Rejected: No change has been made for the first part. Instead of deleting the last paragraph of clause 8.2, it has been made a note.

66. Rejected: No change is necessary.

67. Rejected: No change is necessary.

68. Accepted: Clause 8.5 has been merged into clause 8.

69. Accepted: "A pointer value is said to be 'statically aliased' within a procedure closure <P,A> if there is more than one Box which contains it among the direct associates of range(A).

    "A pointer value is said to be 'dynamically aliased' at a procedure invocation if there is more than one Box which contains it among the generalized associates of the invocation association at initiation."

    "Note: 'static aliasing' is a property of the closure, while 'dynamic aliasing' is a property of the invocation. The above definitions make the assumption that a formal parameter becomes a Box in the invocation association containing the actual parameter value. Since this is not actually required the notion 'Box' must be extended to include the instantiation of the formal/actual parameter bindings for the purposes of the above definition only."