To: X3J3
From: David Muxworthy
Subject: Changes in Language Evolution Edits
Date: August 19, 1994

Revision history: 94-284r1: correction pointed out in discussion of 94-264 and 94-284 at meeting 130 is incorporated; references are improved.

References:		
	94-227	Language Evolution Edits (balloted May 27-June 27 1994)
	94-255	Results of Letter Ballot on Requirement B4.2
	94-264	Language Evolution (revised edits)
	94-276	B4.2 letter ballot - response to comments

This document notes the differences between the B4.2 language evolution edits circulated for letter ballot (X3J3 94-227) and the edits proposed after revision following the comments (X3J3 94-264).

General edit: Sections B.2.1 to B.2.7 in the original paper are renumbered to B.2.2 to B.2.8 in the revision, along with relevant pointers to these sections. These changes are not detailed individually.

D1. Edits relating to the deletion of real DO variables

Old: *In section 8.1.4.4.1, third line, [102:22]:*
Change "scalar-numeric-expr" to "scalar-int-expr", three times.

New: In section 8.1.4.4.1, second line, [102:22]:
Change "scalar-numeric-expr" to "scalar-int-expr", three times.

New: Delete the third constraint which follows rule R918 [123:31-32].

D4. Edits relating to deletion of ASSIGN, assigned GO TO, assigned formats

Old: *In section 2.1, rule R216 [9:29-30]:*
Delete the lines "or assign-stmt" and "or assigned-go-to-stmt".
In section 8.1.4.1.2, rule R829, first constraint [101:20-21]: Add "or" before "an arithmetic-if-stmt" and delete "or an assigned-go-to-stmt".
In section 8.1.4.1.2, rule R833, first constraint [101:33-34]: Add "or" before "an arithmetic-if-stmt" and delete "or an assigned-go-to-stmt".
In section 14.7.6 [251:20-21]:
Delete the section numbered (2) and renumber accordingly.

New: In section 2.1, rule R216 [9:29-30]:
Delete the lines "or assign-stmt" and "or assigned-goto-stmt".
In section 7.1.7, second paragraph [79:42]:
Delete the sentence "An integer operand must be defined with an integer value rather than a statement label value."
In section 8.1.4.1.2, rule R829, first constraint [101:20-21]: Add "or" before "an arithmetic-if-stmt" and delete ", or an assigned-goto-stmt".
In section 8.1.4.1.2, rule R833, first constraint [101:33-34]: Add "or" before "an arithmetic-if-stmt" and delete ", or an assigned-goto-stmt".
In section 14.7.6 [251:20-21 and 252:17-18]:
Delete the sections numbered (2) and (11) and renumber accordingly.

D5. Edits relating to the deletion of the cH edit descriptor

Old: *In the last paragraph, delete the words, "except for the characters following the H in the H edit descriptor and". [137:30]*

New: In the last paragraph, delete the words, "the characters following the H in the H edit descriptor and". [137:30]

O1. Edits related to making computed GO TO obsolescent

New: General edit for the revised rule R216: lines should be sorted firstly by font (normal, then obsolescent), secondly by alphabetical order.

O2. Edits related to making statement functions obsolescent

Old: In section 12.1.2.2.1, item numbered (11), the text "or in a stmt-function-stmt," [164:15-16];

New: In section 12.1.2.2.1, item numbered (11), the text ", or in a stmt-function-stmt," [164:15-16];

Old: In section 14.1.3, the entire first paragraph [245:16-18];

New: In section 14.1.3, the entire first paragraph [245:16-18], which should also be moved to the end of the section;

O4. Edits related to making obsolescent assumed length character functions

Old: [In obsolescent font:] In section 12.3.1.1, paragraph (2)(d) [167:3-4];

New: In section 12.3.1.1, paragraph (2)(d) [167:3-4] replace "neither assumed or constant" by "not constant or not assumed", with "or not assumed" in obsolescent font;

O5. Edits related to making fixed form source obsolescent

Old: Section C.4.2 [265:37-40];

New: Section C.4.2, last sentence [265:39-40].

O6. Edits related to making obsolescent assumed size arrays

Old: In section 12.4.1.4, the text "or assumed-size array" [174:16]; also the last sentence [174:18:20]; [also change "assumed size" to "assumed-size" as this is the only occurrence of the former.]

New: In section 12.4.1.4, the text "or assumed-size array" [174:16]; also the last sentence [174:18-20];

O7. Edits related to making obsolescent CHARACTER*char-length

New: [In obsolescent font:] The first two constraints following rule R508 [42:22-25].

Other related edits

New: In the Introduction, section on Execution control [xvi], replace "STOP, and PAUSE" by "and STOP".

In B.1.4 text:

Old: Further, "assigned-go-to-stmt" should be added to the lists of prohibited statements in the first constraints to rules R829 and R833 in section 8.4.1.1. For completeness, "assigned-stmt" and "assigned-go-to-stmt" ...

New: Further, "assigned-goto-stmt" should be added to the lists of prohibited statements in the first constraints to rules R829 and R833 in section 8.4.1.1. For completeness, "assign-stmt" and "assigned-goto-stmt"...

Old: In section 14.7.5, the sentence in section (10???), "When a numeric

New: In section 14.7.5, the sentence in section (11), "When a numeric

In B.2 text:

Old: The obsolescent features are those features of Fortran 90 that are redundant and for which better methods are available in Fortran 90.

Section 1.6.2 describes the nature of the obsolescent features. The obsolescent features in this International Standard are:

(1) Computed GO TO statement - see B.2.1
(2) Statement functions - see B.2.2
(3) DATA statements amongst executable statements - see B.2.3
(4) Assumed length character functions - see B.2.4
(5) Fixed form source - see B.2.5
(6) Assumed size arrays - see B.2.6
(7) CHARACTER form of CHARACTER declaration - see B.2.7*

New: The obsolescent features are those features of Fortran 90 that were redundant and for which better methods were available in Fortran 90. Section 1.6.2 describes the nature of the obsolescent features. The obsolescent features in this International Standard are:

(1) Arithmetic IF - use the IF statement (8.1.2.4) or IF construct (8.1.2)
(2) Shared DO termination and termination on a statement other than END DO or CONTINUE - use an END DO or a CONTINUE statement for each DO statement
(3) Alternate return - see B.2.1
(4) Computed GO TO statement - see B.2.2
(5) Statement functions - see B.2.3
(6) DATA statements amongst executable statements - see B.2.4
(7) Assumed length character functions - see B.2.5
(8) Fixed form source - see B.2.6
(9) Assumed size arrays - see B.2.7
(10) CHARACTER* form of CHARACTER declaration - see B.2.8

B.2.1 Alternate return

An alternate return introduces labels into an argument list to allow the called procedure to direct the execution of the caller upon return. The same effect can be achieved with a return code that is used in a computed GO TO statement or CASE construct on return. This avoids an irregularity in the syntax and semantics of argument association. For example,

```
CALL SUBR_NAME (X, Y, Z, *100, *200, *300)
```

may be replaced by

```
CALL SUBR_NAME (X, Y, Z, RETURN_CODE)
SELECT CASE (RETURN_CODE)
  CASE (1)
    ...
  CASE (2)
    ...
  CASE (3)
    ...
  CASE DEFAULT
    ...
END SELECT
```

Subsequent sections (B.2.2 to B.2.8) have been renumbered from B.2.1 to B.2.7 respectively.