Date: 4 March 1999
To: J3
From: Van Snyder
Subject: Specifications, syntax and edits for M.23: Access to status error messages
References: 97-159, 98-172 98-173r1 98-208, 98-213

1 Background

Specifications and three possible syntaxes to access status error messages were proposed in paper 98-173r1. Informal discussions suggested the "intrinsic procedure" approach is the preferred approach.

Paper 98-213, proposing the intrinsic procedure approach, was presented and discussed at meeting 147. Several issues were raised but not resolved, and the paper was withdrawn after three unsuccessful revisions. Subsequent e-mail discussion revealed that the intrinsic procedure approach does not support access to error messages produced by user-defined derived-type input/output procedures. The approach of using an ERRMSG= specifier in input/output, allocate and deallocate statements does work for this circumstance. This paper proposes specifications, syntax and edits for using an ERRMSG= specifier in input/output, allocate and deallocate statements to provide access to status messages.

2 Specifications and Syntax

Define an ERRMSG=*errmsg-variable* specifier for use in input/output, allocate and deallocate statements. The *errmsg-variable* shall be a default character scalar. The `errmsg` argument of the derived-type input/output procedures shall have the same characteristics as the *errmsg-variable*.

3 Edits

Edits refer to 99-007. Page and line numbers are displayed in the margin. Absent other instructions, a page and line number or line number range implies all of the indicated text is to be replaced by immediately following text, while a page and line number followed by + indicates that immediately following text is to be inserted after the indicated line. Remarks for the editor are noted in the margin, or appear between [and] in the text.

■ [, *alloc-opt-list*]) 104:19

R623a *alloc-opt* **is** STAT = *stat-variable*
or ERRMSG = *errmsg-variable*

R623b *errmsg-variable* **is** *variable*

Constraint: The *errmsg-variable* shall be a default character scalar. 104:34+
Constraint: No *alloc-opt* specifier shall appear more than once in a given *alloc-opt-list*.

[Editor: Change "The *stat-variable* shall not" to "Neither the *stat-variable* nor the *errmsg-variable* shall".] 106:20

[Editor: Change "it" to "they"] 106:21

The ERRMSG= specifier is described in [new section] 6.4.1.4. 106:41+

[Editor: Insert a new section] 108:52+

6.4.1.4 ERRMSG= specifier

If an error condition occurs during execution of an ALLOCATE or DEALLOCATE statement the processor shall assign a message explaining the condition to the *errmsg-variable*

If no error condition occurs during execution of an ALLOCATE or DEALLOCATE statement, the processor shall not change the value of the *errmsg-variable*.

R632 *deallocate-stmt* **is** DEALLOCATE (*allocate-object-list* ■ 109:15
■ [, *alloc-opt-list*])

[Editor: Change "The *stat-variable* shall not" to "Neither the *stat-variable* nor the *errmsg-variable* shall". Change "it" to "they".] 109:20

The ERRMSG= specifier is described in [new section] 6.4.1.4. 109:31+

---

or ERRMSG=*errmsg-variable* 173:37+

Constraint: The *errmsg-variable* shall be a default character scalar. 174:6+

The IOSTAT=, ERR=, and ERRMSG= specifiers are described in [new section] 9.9. 174:16

---

or ERRMSG=*errmsg-variable* 178:9+

Constraint: The *errmsg-variable* shall be a default character scalar. 178:13+

The IOSTAT=, ERR=, and ERRMSG= specifiers are described in [new section] 9.9. 178:20

---

or ERRMSG=*errmsg-variable* 179:16+

Constraint: The *errmsg-variable* shall be a default character scalar. 179:24+

[Delete – moved into section 9.10] 180:16-22

The IOSTAT=, ERR=, EOR=, END=, and ERRMSG= specifiers are described in [new section] 9.9. 180:27+

[Editor: Insert into the list:] 182:17+

($3\frac{1}{2}$) If the input/output statement also contains an ERRMSG= specifier, the *errmsg-variable* becomes defined as specified in [new section] 9.9.2.

[Editor: Insert into the list:] 182:28+

($3\frac{1}{2}$) If the input/output statement also contains an ERRMSG= specifier, the *errmsg-variable* becomes defined as specified in [new section] 9.9.2.

[Editor: Insert into the list:] 182:39+

($4\frac{1}{2}$) If the input/output statement also contains an ERRMSG= specifier, the *errmsg-variable* becomes defined as specified in [new section] 9.9.2.

If an error condition occurs during execution of an input/output statement that contains an IOSTAT= or ERR= specifier, or an end-of-file condition occurs during execution of a READ or WAIT statement that contains an END= specifier, or an end-of-record condition occurs during execution of a WAIT or nonadvancing READ statement, execution continues as specified in [new section] 9.9. 187:37+

If the value true is assigned to the `err` argument, the derived-type input/output procedure shall assign a message explaining the condition to the `errmsg` argument. 195:25

If the value false is assigned to the `err` argument, the derived-type input/output procedure shall not change the value of the `errmsg` argument.

or ERRMSG=*errmsg-variable* 198:1+

Constraint: The *errmsg-variable* shall be a default character scalar. 198:9+

The IOSTAT=, ERR=, EOR=, END=, and ERRMSG= specifiers are described in [new section] 9.9. 198:13

or ERRMSG=*errmsg-variable* 199:9+

Constraint: The *errmsg-variable* shall be a default character scalar. 199:10+

The IOSTAT=, ERR=, and ERRMSG= specifiers are described in [new section] 9.9. 199:17

or ERRMSG=*errmsg-variable* 201:6+

Constraint: The *errmsg-variable* shall be a default character scalar. 201:38+

The IOSTAT=, ERR=, and ERRMSG= specifiers are described in [new section] 9.9. 201:47

[Editor: Delete (moved into 9.10 at 207:34+).] 207:2-4

[Editor: new section] 207:17+

9.9 Error processing

Error processing takes place if an error occurs during execution of input/output statements that have IOSTAT= or ERR= specifiers, an end-of-file condition occurs during execution of READ or WAIT statements that have IOSTAT= or END= specifiers, or an end-of-record condition occurs during execution of WAIT or nonadvancing read statements that have EOR= specifiers.

[Editor: Move 9.5.1.5 here, change its title to **9.9.1 IOSTAT= specifier**.]

9.9.2 ERRMSG= specifier

If an error, end-of-file or end-of-record condition occurs during execution of an input/output statement the processor shall assign a message explaining the condition to the *errmsg-variable*.

If no error condition, end-of-file condition, or end-of-record condition occurs during execution of an input/output statement, the processor shall not change the value of the *errmsg-variable*.

[Editor: Move 9.5.1.6 – 9.5.1.8 here, changing their numbers to 9.9.3 – 5, and their titles to "ERR= specifier", "END= specifier" and "EOR= specifier" respectively.]

A variable that may become defined or undefined as a result of its use in a specifier in an input/output statement, or any associated entity, shall not appear in another specifier in the same statement, nor shall it appear in or be associated with any entity in a data transfer list, *namelist-group-object-list*, or *do-variable* of an *io-implied-do*, in the same data transfer statement. If it is an array element reference, its subscript values shall not be affected by the data transfer, the *io-implied-do* processing, or the definition or evaluation of any other specifier in the *io-control-spec-list*, in the same statement. 207:34+ *Overkill?*

[Editor: Delete] 279:12-21

[Editor: Delete] 283:27-28

[Editor: Delete] 329:1-10