

Date: 29 June 1998
To: J3
From: Van Snyder
Subject: Rationale, Specs, Edits for B.6: Access to status error messages
References: 97-159, 98-172

The mechanisms proposed here are a lot simpler than any in 97-159.

Some of the mechanisms proposed in this paper depend on facilities proposed in paper 98-172 *Explicitly typed allocations - Rationale, Specs, Syntax, Edits*.

1 Rationale

Error status codes from input/output, ALLOCATE and DEALLOCATE statements are not standardized. This makes it difficult to cope gracefully with erroneous conditions in a portable way. MTE B.6 *Access to status error messages* is intended to remedy this problem.

2 Specs

Choose zero or more of the following. The simplest change to the standard is the first.

2.1 Intrinsic procedure

Define an intrinsic subroutine to provide access to error messages that would have been displayed had IOSTAT= or STAT= specifiers been absent from input/output, ALLOCATE and DEALLOCATE statements. The procedure can write the message to a file or to the usual message place, or return it in a variable. Exploit deferred-length allocation (see 97-172) to provide just the right size of output variable.

2.2 ERMSG= specifier – output from statement

Add a specifier to input/output, ALLOCATE and DEALLOCATE statements to indicate where the text of error messages is to be stored. Exploit deferred-length allocation (see 97-172) to provide just the right size of output variable. The spelling ERMSG= has been proposed by others, but spelling is negotiable.

2.3 ERMSG= specifier – input to statement

Add a specifier to input/output, ALLOCATE and DEALLOCATE statements to indicate that error messages, if any, are to be displayed even if IOSTAT= or STAT= are present. E.g. ERMSG='YES' or ERMSG='NO'. Spelling is negotiable.

3 Edits

3.1 Intrinsic procedure

Edits refer to 98-007r2. Page and line numbers are displayed in the margin. A page and line number followed by + indicates that immediately following text is to be inserted after the indicated line.

STATUS_ERROR_MSG (STAT [,	Access status error messages.	[253:5+]
UNIT, ACTION, MESSAGE])		
13.14.104+ STATUS_ERROR_MSG (STAT [, UNIT, ACTION, MESSAGE])		[296:40+]
Description. Access error message corresponding to error status returned by IOSTAT= or STAT= specifiers in input/output, ALLOCATE and DEALLOCATE statements. Display the message or write it to a specified unit. Carry out the default action that would have occurred had IOSTAT= or STAT= specifiers been absent from input/output, ALLOCATE or DEALLOCATE statements.		
Class. Subroutine		
Arguments.		
STAT	shall be scalar and of type default integer. It is an INTENT(IN) argument. If it does not have a value returned by an IOSTAT= specifier in an input/output statement, or a STAT= specifier in an ALLOCATE or DEALLOCATE statement, the UNIT and ACTION arguments are considered to be absent, and if MESSAGE is present and has the ALLOCATABLE or POINTER attribute, it is allocated with zero elements, else if MESSAGE is present its first element is assigned the value blank.	
UNIT (optional)	shall be scalar and of type default integer. It is an INTENT(IN) argument. If UNIT is present and has a value that is the unit number of a unit opened for sequential formatted output the message is written on the specified unit using FORMAT(A). If UNIT is present and has a value that is not a unit number of a unit opened for sequential formatted output, or if attempting to display the message on unit UNIT fails, the message is displayed where it would have been had the IOSTAT= or STAT= specifier been absent from the input/output, ALLOCATE or DEALLOCATE statement.	
ACTION (optional)	shall be scalar and of type default logical. It is an INTENT(IN) argument. ACTION is ignored if UNIT is absent. If UNIT is present, and ACTION is present and has the value true, the action that would have occurred had the IOSTAT= or STAT= specifier been absent from the input/output, ALLOCATE or DEALLOCATE statement takes place after the message is output.	

MESSAGE (optional) shall be of type default character and rank zero or one. It is an INTENT(OUT) argument. If it has the ALLOCATABLE or POINTER attribute it is allocated with a number of elements equal to the number of lines in the message. If it has deferred length it is allocated with a length equal to the length of the longest line of the message. If attempting to allocate MESSAGE fails and UNIT is absent the effect is as though MESSAGE were absent and UNIT were present with the value zero. If MESSAGE is scalar it is assigned the first line of the message, else if the message has k or more lines the k 'th element of MESSAGE is assigned the k 'th line of the message.

One of my colleagues thinks he wouldn't use the MESSAGE argument. If J3 agrees, the procedure and its description would be simplified by removing it. *J3 note*

3.2 ERMSG= specifier – output from statement

R623 <i>allocate-stmt</i>	is ALLOCATE(<i>allocation-list</i> ■ ■ [, <i>alloc-opt-list</i>])	[90:13]
---------------------------	--	---------

R623a <i>alloc-opt</i>	is STAT = <i>stat-variable</i> or ERMSG = <i>default-char-variable</i>	
------------------------	---	--

[J3 note: This is the same in principle as the description in section 9.8.2 below. Perhaps they could be combined.] [91:9+]

The *default-char-variable* specified by an ERMSG= specifier shall be a rank zero or rank one default character variable. If the ERMSG= specifier is present, successful execution of an ALLOCATE statement causes the *default-char-variable* to be allocated with zero elements if it has the ALLOCATABLE or POINTER attribute, else it becomes defined by assigning blank to it. If it is an array only the first element is assigned. If an error condition occurs, the *default-char-variable* is allocated if it has the ALLOCATABLE or POINTER attribute. The number of elements is equal to the number of lines of a processor-dependent message. If the length is deferred the length is allocated as the number of characters in the longest line of the message. If the *default-char-variable* is a scalar, the first line of the message is assigned to it, else if the message has k or more lines, the k 'th element is assigned the k 'th line of the message.

[Editor: replace “the STAT=” by “a STAT= or ERMSG=”] [91:10-11]

R632 <i>deallocate-stmt</i>	is DEALLOCATE (<i>allocate-object-list</i> ■ ■ [, <i>alloc-opt-list</i>])	[93:27]
-----------------------------	--	---------

[J3 note: This is the same in principle as the description for ALLOCATE above. Perhaps they could be combined.] [93:43+]

The *default-char-variable* specified by an ERMSG= specifier shall be a rank zero or rank one default character variable. If the ERMSG= specifier is present, successful execution of a DEALLOCATE statement causes the *default-char-variable* to be allocated with zero elements if it has the ALLOCATABLE or POINTER attribute, else it becomes defined by assigning blank to it. If it is an array only the first element is assigned. If an error condition occurs, the *default-char-variable* is allocated if it has the ALLOCATABLE or POINTER attribute. The number of elements is equal to the number of lines of a processor-dependent message. If the length is deferred the length is allocated as the number of characters in the longest line of the message. If the *default-char-variable* is a scalar, the first line of the message is assigned to it, else if the

message has k or more lines, the k 'th element is assigned the k 'th line of the message.

[Editor: Replace “the STAT=” by “a STAT= or ERMSG=”]	[94:2]
or ERMSG= <i>default-char-variable</i>	[152:18+]
or ERMSG= <i>default-char-variable</i>	[155:26+]
or ERMSG= <i>default-char-variable</i>	[156:30+]
[Editor: Make the following changes, then move all of section 9.4.3 to a new section 9.8.6 (renumber existing 9.8 and 9.9). See edit instructions below for page [180:42+]. This paragraph applies to all input/output statements, not just the act of data transfer (which is the title of section 9.4).]	[163:10-37]
[Editor: Change “IOSTAT=” to “IOSTAT=, ERMSG=”]	[163:34]
[Editor: Change “IOSTAT= specifier” to “IOSTAT=, ERMSG=”]	[163:37]
or ERMSG= <i>default-char-variable</i>	[174:14+]
or ERMSG= <i>default-char-variable</i>	[176:6+]
[Editor: new section]	[180:42+]

9.8 Abnormal termination of input/output statements

[Editor: Move sections 9.4.1.4 – 9.4.1.7 and 9.4.3 here, calling them 9.8.1, 9.8.3, 9.8.4, 9.8.5, and 9.8.6, respectively. Change the title of 9.8.1 to **IOSTAT= specifier**. Insert the following section between 9.8.1 and 9.8.3:]

9.8.2 ERMSG= specifier

The *default-char-variable* specified by an ERMSG= specifier shall be a rank zero or rank one default character variable. If the ERMSG= specifier is present, and neither an error condition, an end-of-file condition nor an end-of-record condition occurs, the *default-char-variable* is allocated with zero elements if it has the ALLOCATABLE or POINTER attribute, else it becomes defined by assigning blank to it. If it is an array only the first element is assigned. If an error condition, an end-of-file condition or an end-of-record condition occurs, the *default-char-variable* is allocated if it has the ALLOCATABLE or POINTER attribute. The number of elements is equal to the number of lines of a processor-dependent message. If the length is deferred the length is allocated as the number of characters in the longest line of the message. If the *default-char-variable* is a scalar, the first line of the message is assigned to it, else if the message has k or more lines, the k 'th element is assigned the k 'th line of the message.

3.3 ERMSG= specifier – input to statement

R623 <i>allocate-stmt</i>	is ALLOCATE(<i>allocation-list</i> ■ ■ [, <i>alloc-opt-list</i>])	[90:13]
R623a <i>alloc-opt</i>	is STAT = <i>stat-variable</i> or ERMSG = <i>scalar-default-char-expr</i>	
Constraint: ERMSG= shall not appear if IOSTAT= is absent.		[90:23+]
If the ERMSG= specifier is present the <i>scalar-default-char-expr</i> shall evaluate to YES or NO. If an error condition occurs during execution of an ALLOCATE statement and the <i>scalar-default-char-expr</i> evaluates to YES, a processor-defined error message shall be displayed. If the STAT= specifier is present and the ERMSG= specifier is absent, or the <i>scalar-default-char-expr</i> evaluates to NO, an error message shall not be displayed.		[91:9+]

R632 <i>deallocate-stmt</i>	is DEALLOCATE(<i>allocation-list</i> ■ ■ [, <i>alloc-opt-list</i>])	[93:27]
Constraint: ERMSG= shall not appear if STAT= is absent.		[93:28+]
If the ERMSG= specifier is present the <i>scalar-default-char-expr</i> shall evaluate to YES or NO. If an error condition occurs during execution of a DEALLOCATE statement and the <i>scalar-default-char-expr</i> evaluates to YES, a processor-defined error message shall be displayed. If the STAT= specifier is present and the ERMSG= specifier is absent, or the <i>scalar-default-char-expr</i> evaluates to NO, an error message shall not be displayed.		[93:43+]
or ERMSG= <i>default-char-expr</i>		[152:18+]
Constraint: ERMSG= shall not appear if IOSTAT= is absent.		[152:37+]
or ERMSG= <i>default-char-expr</i>		[155:26+]
Constraint: ERMSG= shall not appear if IOSTAT= is absent.		[155:34+]
or ERMSG= <i>default-char-expr</i>		[156:30+]
Constraint: ERMSG= shall not appear if IOSTAT= is absent.		[156:39+]
[Editor: Move all of section 9.4.3 to a new section 9.8.6 (renumber existing 9.8 and 9.9). See edit instructions below for page [180:42+]. This paragraph applies to all input/output statements, not just the act of data transfer (which is the title of section 9.4).]		[163:10-37]
or ERMSG= <i>default-char-expr</i>		[174:14+]
Constraint: ERMSG= shall not appear if IOSTAT= is absent.		[174:15+]
or ERMSG= <i>default-char-expr</i>		[176:6+]
Constraint: ERMSG= shall not appear if IOSTAT= is absent.		[176:36+]
[Editor: new section]		[180:42+]

9.8 Abnormal termination of input/output statements

[Editor: Move sections 9.4.1.4 – 9.4.1.7 and 9.4.3 here, calling them 9.8.1, 9.8.3, 9.8.4, 9.8.5, and 9.8.6, respectively. Change the title of 9.8.1 to **IOSTAT= specifier**. Insert the following section between 9.8.1 and 9.8.3:]

9.8.2 ERMSG= specifier

The *default-char-expr* may be a scalar or a rank one array. Its elements shall evaluate to YES, NO, EOF, EOR, ERR or blank. If it is an array and an element evaluates to YES or NO, its extent shall be one or other elements shall evaluate to blank. If the IOSTAT= specifier is present and the ERMSG= specifier is absent, or *default-char-expr* has no elements that evaluate to YES, EOF, EOR or ERR, messages shall not be displayed. If the IOSTAT= specifier and the ERMSG= specifier are both absent, or an element of *default-char-expr* evaluates to YES, processor-dependent messages shall be displayed when the IOSTAT= specifier is (or would be) defined with a nonzero value. Otherwise, a message shall be displayed for end-of-file if and only if an element of *default-char-expr* evaluates to EOF, a message shall be displayed for end-of-record if and only if an element of *default-char-expr* evaluates to EOR, and a message shall be displayed for an error condition if and only if an element of *default-char-expr* evaluates to ERR.

4 A “Quality of Implementation” issue in sections 3.1 and 3.2

It would be useful if the system could make the MESSAGE argument in section 3.1 or the *default-char-variable* in section 3.2 point to the error message, or allocate a buffer and copy the message to it, as it chooses. It must do the “right thing” when MESSAGE or *default-char-variable* is deallocated.