Date:	5 August 1998
To:	J3
From:	Van Snyder
Subject:	Edits for B.6: Access to status error messages
References:	97-159, 98-172 98-173r1 98-208

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.

The procedure proposed here is simpler than the one proposed in 98-173r1 – the ACTION argument has been removed. Some J3 members informally expressed a preference to remove the UNIT argument, but informal discussion with potential users suggest the most common usage would be with UNIT=0 and MESSAGE absent, which would have the effect of displaying the error message in the usual place, but at an instant under the program's control, and with termination under the program's control.

This paper depends on facilities proposed in papers 98-172 *Explicitly typed allocations - Rationale, Specs, Syntax, Edits* and 98-208 *Edits for explicitly typed allocations.*

2 Specifications

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 unit or to the usual message destination, or return it in a variable. Exploit deferred-length allocation (see 98-172 and 98-208) to provide exactly the right length of output variable.

3 Edits

Edits refer to 98-007r3. 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.

STATUS_ERROR_MSG (STAT [, UNIT, MESSAGE])	Access status error messages.	[267:9+]

13.14.107+ STATUS_ERROR_MSG (STAT [, UNIT, MESSAGE]) [311:35+]

Description. Access error message corresponding to error status returned by IOSTAT= or STAT= specifiers in input/output, ALLOCATE and DEALLOCATE statements. Optionally display the message or write it to a specified unit. Optionally return it in an argument.

Class. Subroutine

Arguments.

STAT	shall be scalar and of type default integer. It is an INTENT(IN) argument. If it has a nonzero value returned by an IOSTAT= specifier in an input/output statement, or a STAT= specifier in an ALLOCATE or DEALLOCATE statement, then the message that would have been displayed had IOSTAT= or STAT= been absent will optionally be output as specified by the UNIT argument, or copied to the MESSAGE argument. If it has the value zero or a nonzero value not returned by an IOSTAT= specifier in an input/output statement, or a STAT= specifier in an ALLOCATE or DEALLOCATE statement, then the UNIT argument is considered to be absent, and if MESSAGE is present and has the ALLOCATABLE or POINTER attribute, then 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 then each line of the message is written on the specified unit using list-directed output. 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 then the message is displayed where it would have been displayed had the IOSTAT= or STAT= specifier been absent from the input/output, ALLOCATE or DEALLOCATE statement.
MESSAGE (optional)	shall be of type default character and rank zero or one. It is an INTENT(OUT) argument. If it is has the ALLOCATABLE attribute and is associated then it is deallocated. If it is has the ALLOCATABLE or POINTER attribute then it is allocated with a number of elements equal to the number of lines in the message. If it has deferred length then 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 then the effect is as though MESSAGE were absent and UNIT were present with the value zero. If MESSAGE is scalar then it is assigned the first line of the message, else if the message has k or more lines then 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. Then, there would be no reason for UNIT to be optional. *J3 note*