Date:	29 November 1998
To:	J3
From:	Van Snyder
Subject:	Specifications, syntax and edits to access standard unit numbers (unresolved issue 63)

1 Background

There is at present no way to access the unit numbers equivalent to "asterisk" units in READ and WRITE statements.

Because it is not possible to access these unit numbers, it is not possible to specify or change any default modes for "asterisk" units in OPEN statments. In addition to default behavior specifiers present in Fortran 95, the impossibility to access these unit numbers affects cultural adaptability (DECIMAL= specifier), facilities to support interval arithmetic (ROUND= specifier) and stream input/output (ACCESS= specifier).

In many applications it is desirable to be able to write either to a file or to "standard output," depending on user input or other criteria. To be certain of portability, these applications need a test to decide whether to write to a specified unit, or to the "asterisk" unit, with redundant output statements that are identical except for the unit specifiers. These applications would be simplified if it were possible to access a unit number having an effect equivalent to "standard output." Similar considerations apply for the choice whether to read from "standard input" or from a file.

2 Specifications

There are at least five ways to access standard input/output units.

In all cases, we propose that three standard input/output units should be available, identified here as INPUT, OUTPUT, and ERROR. The INPUT and OUTPUT units are the processor-dependent pre-connected units identified by an asterisk in READ and WRITE statements, respectively. If the processor has a unit that is pre-connected for sequential formatted output to the "standard error" stream, the ERROR unit is that unit; otherwise ERROR has the same value as OUTPUT.

Any of these unit numbers are allowed to be negative. No other negative unit numbers are allowed.

2.1 Intrinsic Subroutine

Provide an intrinsic subroutine STANDARD_IO_UNITS having INTENT(OUT) arguments OUTPUT, ERROR, and INPUT.

This option has not received support in informal discussions and is not further described here.

2.2 Three Intrinsic Functions

Provide separate intrinsic functions, e.g. STANDARD_OUTPUT_UNIT, STANDARD_ERROR_UNIT, and STANDARD_INPUT_UNIT, to access each standard input/output unit value.

The advantage of an intrinsic function, as compared to a subroutine, is that an intrinsic function can be used in a specification, e.g.

```
  INTEGER, PARAMETER :: MY_UNIT = STANDARD_OUTPUT_UNIT()
```

The disadvantage of this approach is that three intrinsic procedure names are introduced.

This option has not received support in informal discussions and is not further described here.

2.3 One Intrinsic Function

Provide a single intrinsic function that accesses different standard input/output unit values, depending on its argument, e.g. STANDARD_IO_UNIT (WHAT_UNIT), in which the actual argument associated with WHAT_UNIT is a default character expression with one of the values INPUT, OUTPUT, or ERROR.

This approach has the advantage of being usable in a specification expression, and in addition introduces only one intrinsic procedure name.

2.4 Intrinsic Module

Provide an intrinsic module, e.g. SYSTEM, that makes public three parameters of default integer type, e.g. STANDARD_OUTPUT_UNIT, STANDARD_ERROR_UNIT, and STANDARD_INPUT_UNIT.

This approach has the advantage of being usable in a specification expression, and in addition introduces no intrinsic procedure names. If J3 had chosen to export a parameter from an intrinsic module to identify the KIND parameters for DEFAULT, ASCII, and ISO-10646 characters, we would have have at least six parameters to export from the intrinsic module SYSTEM.

2.5 INQUIRE statement

An additional specifier could be added to the INQUIRE statement, and used like inquire-by-file to get these unit numbers. This one seems a bit messy to describe, and is a source of potential controversy – which specifiers can be added? It's not further described here.

3 Syntax

The syntaxes are all obvious. Names are negotiable.

4 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.

4.1 One Intrinsic Function

If the value of a scalar integer expression that identifies an external file unit is negative, it shall be the same as the unit argument of a currently active user-defined derived-type input/output procedure. 171:29

[Editor: Delete] 172:3-6

[Editor: Change “*” to “an asterisk unit” for consistency with usage elsewhere in section 9.] 194:38

- If the parent data transfer statement is a WRITE statement with an asterisk unit or a PRINT statement, the unit argument shall have the same value as the result of the STANDARD_IO_UNIT intrinsic function with the OUTPUT argument. 194:40
- If the parent data transfer statement is a READ statement with an asterisk unit or a READ statement without a control list, the unit argument shall have the same value as the result of the STANDARD_IO_UNIT intrinsic function with the INPUT argument.
- Otherwise (the parent data transfer statement accesses an internal file), the unit argument shall have a processor-dependent negative value.

[Editor: Remove “, the “*” unit or no unit”] 194:43

[Editor: Insert a new section 13.11 and renumber subsequent sections.] 278:23+

13.11 Access standard input/output units

The result value of the intrinsic function STANDARD_IO_UNIT is the processor-dependent preconnected external unit identified by an asterisk in a READ or WRITE statement, or a processor-dependent external unit preconnected for sequential formatted output for the purpose of error reporting, if such a unit exists.

[Editor: Insert a new section 13.12.22 and renumber subsequent sections.] 283:15+

13.12.22 Access standard input/output units

STANDARD_IO_UNIT(WHAT_UNIT)	The result value is the processor-dependent preconnected external unit identified by an asterisk in a READ or WRITE statement, or a processor-dependent external unit preconnected for sequential formatted output for the purpose of error reporting, if such a unit exists.

[Editor: Insert a new section 13.15.110 and renumber subsequent sections.] 329:21+

13.15.110 STANDARD_IO_UNIT(WHAT_UNIT)

Description. Return the processor-dependent preconnected external unit identified by an asterisk in a READ or WRITE statement, or a processor-dependent external unit preconnected for sequential formatted output for the purpose of error reporting, if such a unit exists.

Class. Inquiry function.

Argument.

WHAT_UNIT	shall be of type default character. It shall evaluate to INPUT, OUTPUT, or ERROR. Trailing blanks are ignored. The value is without regard to case.

Result Characteristics. Scalar default integer.

Result Value. The result depends on the value of the argument in the following way. If the argument is:

Case (INPUT): The processor-dependent preconnected external unit identified by an asterisk in a READ statement.

Case (OUTPUT): The processor-dependent preconnected external unit identified by an asterisk in a WRITE statement.

Case (ERROR): If an external unit is preconnected for sequential formatted output for the purpose of error reporting, the result value identifies that unit. All other characteristics of this connection are initially the default characteristics that would result from omitting their specifiers in an OPEN statement. If no such unit exists, the result value is the same as for the OUTPUT case.

4.2 Intrinsic Module

If the value of a scalar integer expression that identifies an external file unit is negative, it shall be the same as the `unit` argument of a currently active user-defined derived-type input/output procedure. 171:29

[Editor: Delete] 172:3-6

[Editor: Change "*" to "an asterisk unit" for consistency with usage elsewhere in section 9.] 194:38

- If the parent data transfer statement is a WRITE statement with an asterisk unit or a PRINT statement, the `unit` argument shall have the same value as the OUTPUT_UNIT parameter of the SYSTEM intrinsic module (16). 194:40
- If the parent data transfer statement is a READ statement with an asterisk unit or a READ statement without a control list, the `unit` argument shall have the same value as the INPUT_UNIT parameter of the SYSTEM intrinsic module (16).
- Otherwise (the parent data transfer statement accesses an internal file), the `unit` argument shall have a processor-dependent negative value.

[Editor: Remove ", the "*" unit or no unit"] 194:43+

[Editor: Insert a new section 16, renumber C interoperability to be section 17.] 376+

Section 16: Intrinsic module for standard system-dependent constants

The intrinsic module SYSTEM provides public parameters giving processor-dependent values.

16.1 Standard input/output units

Three parameters giving processor-dependent values of preconnected units shall be provided by the processor.

16.1.1 INPUT_UNIT

The value of the default integer parameter INPUT_UNIT identifies the processor-dependent preconnected external unit identified by an asterisk in a READ statement.

16.1.2 OUTPUT_UNIT

The value of the default integer parameter OUTPUT_UNIT identifies the processor-dependent preconnected external unit identified by an asterisk in a WRITE statement.

16.1.3 ERROR_UNIT

If an external unit is preconnected for sequential formatted output for the purpose of error reporting, the value of the default integer parameter ERROR_UNIT identifies that unit. All other characteristics of this connection are initially the default characteristics that would result from omitting their specifiers in an OPEN statement. If no such unit exists, the value of ERROR_UNIT is the same as OUTPUT_UNIT.

Some of the characteristics of preconnected standard input/output units may be changed by an OPEN statement (9.4.4). Note 16.1