

ieee_exceptions: IEEE exceptions module

July 31, 2012

1 Name

`ieee_exceptions` — Intrinsic module providing access to IEEE exceptions

2 Usage

USE,INTRINSIC :: IEEE_EXCEPTIONS

This module provides types, parameters and procedures for using IEEE exception flags. Its contents conform to technical report ISO/IEC TR 15580:1998(E).

3 Synopsis

Derived Types

IEEE_FLAG_TYPE, IEEE_STATUS_TYPE.

Parameters

IEEE_ALL, IEEE_DIVIDE_BY_ZERO, IEEE_INEXACT, IEEE_INVALID, IEEE_OVERFLOW,
IEEE_UNDERFLOW, IEEE_USUAL.

Procedures

IEEE_GET_FLAG, IEEE_GET_HALTING_MODE, IEEE_GET_STATUS, IEEE_SET_FLAG,
IEEE_SET_HALTING_MODE, IEEE_SET_STATUS, IEEE_SUPPORT_FLAG,
IEEE_SUPPORT_HALTING.

4 Derived-Type Description

```
TYPE IEEE_FLAG_TYPE
  PRIVATE
  ...
END TYPE
```

This type is provided for the purpose of specifying IEEE exception flags; a value of this type indicates each particular flag.

```
TYPE IEEE_STATUS_TYPE
  PRIVATE
  ...
END TYPE
```

This type is provided to allow the user to save (and restore) the current floating-point status. The floating-point status includes the state (quiet or signalling) of each IEEE exception flag, and the current rounding mode.

5 Parameter Description

```
TYPE(IEEE_FLAG_TYPE),PARAMETER :: IEEE_ALL = &  
    (/ IEEE_USUAL,IEEE_UNDERFLOW,IEEE_INEXACT /)
```

Array specifying all IEEE exception flags.

```
TYPE(IEEE_FLAG_TYPE),PARAMETER :: IEEE_DIVIDE_BY_ZERO
```

Specifies the division-by-zero exception flag.

```
TYPE(IEEE_FLAG_TYPE),PARAMETER :: IEEE_INEXACT
```

Specifies the inexact exception flag.

```
TYPE(IEEE_FLAG_TYPE),PARAMETER :: IEEE_INVALID
```

Specifies the invalid exception flag.

```
TYPE(IEEE_FLAG_TYPE),PARAMETER :: IEEE_OVERFLOW
```

Specifies the overflow exception flag.

```
TYPE(IEEE_FLAG_TYPE),PARAMETER :: IEEE_UNDERFLOW
```

Specifies the underflow exception flag.

```
TYPE(IEEE_FLAG_TYPE),PARAMETER :: IEEE_USUAL = &  
    (/ IEEE_OVERFLOW,IEEE_DIVIDE_BY_ZERO,IEEE_INVALID /)
```

Array specifying the most commonly desired subset of exception flags.

6 Procedure Description

```
ELEMENTAL SUBROUTINE IEEE_GET_FLAG(FLAG,FLAG_VALUE)
```

```
TYPE(IEEE_FLAG_TYPE),INTENT(IN) :: FLAG  
LOGICAL,INTENT(OUT) :: FLAG_VALUE
```

Sets FLAG_VALUE to .TRUE. if the specified IEEE exception flag is signalling, and to .FALSE. otherwise.

```
ELEMENTAL SUBROUTINE IEEE_GET_HALTING_MODE(FLAG,HALTING_MODE)
```

```
TYPE(IEEE_FLAG_TYPE),INTENT(IN) :: FLAG  
LOGICAL,INTENT(OUT) :: HALTING_MODE
```

Sets `HALTING_MODE` to `.TRUE.` if an occurrence of the specified exception will cause program termination, and to `.FALSE.` otherwise.

```
SUBROUTINE IEEE_GET_STATUS(STATUS_VALUE)

TYPE(IEEE_STATUS_TYPE),INTENT(OUT) :: STATUS_VALUE
```

Stores the current floating-point status in `STATUS_VALUE`.

```
ELEMENTAL SUBROUTINE IEEE_SET_FLAG(FLAG,FLAG_VALUE)

TYPE(IEEE_FLAG_TYPE),INTENT(IN) :: FLAG
LOGICAL,INTENT(IN) :: INTENT(IN) :: FLAG_VALUE
```

Sets the specified exception flag to signalling if `FLAG_VALUE` is `.TRUE.`, and to quiet otherwise.

```
ELEMENTAL SUBROUTINE IEEE_SET_HALTING_MODE(FLAG,HALTING_MODE)

TYPE(IEEE_FLAG_TYPE),INTENT(IN) :: FLAG
LOGICAL,INTENT(IN) :: HALTING_MODE
```

Sets the halting mode for exception `FLAG` to true if `HALTING_MODE` is `.TRUE.`, and to false otherwise. If the halting mode for an exception is true, an occurrence of that exception will cause program termination.

```
SUBROUTINE IEEE_SET_STATUS(STATUS_VALUE)

TYPE(IEEE_STATUS_TYPE),INTENT(IN) :: STATUS_VALUE
```

Restores the current floating-point status to that saved by a preceding invocation of `IEEE_GET_STATUS`.

```
LOGICAL FUNCTION IEEE_SUPPORT_FLAG(FLAG,X)

TYPE(IEEE_FLAG_TYPE),INTENT(IN) :: FLAG
REAL(any kind),OPTIONAL :: X
```

Returns `.TRUE.` if detection of the specified exception is supported; either for all real kinds (if `X` is absent), or for reals of the same kind as `X`.

```
LOGICAL FUNCTION IEEE_SUPPORT_HALTING(FLAG)

TYPE(IEEE_FLAG_TYPE),INTENT(IN) :: FLAG
```

Returns `.TRUE.` if `IEEE_SET_HALTING` can be used to alter the halting mode of the specified exception, for all real kinds for which `IEEE_SUPPORT_FLAG` returns `.TRUE.`.

7 See Also

`nagfor(1)`, `ieee_arithmetic(3)`, `ieee_features(3)`, `intro(3)`, `nag_modules(3)`.

8 Bugs

Please report any bugs found to ‘support@nag.co.uk’ or ‘support@nag.com’, along with any suggestions for improvements.