f90_kind: KIND Number Parameters Module

March 8, 2024

1 Name

`f90_kind` — Module providing useful KIND number parameters

2 Usage

USE F90_KIND

3 Synopsis

Parameters

ASCII, BYTE, DOUBLE, INT16, INT32, INT64, INT8, JIS, LOGICAL64, QUAD, REAL128, REAL32, REAL64, REAL64x2, SINGLE, TWOBYTE, UCS2, UCS4, WORD.

4 Parameter Description

```
INTEGER,PARAMETER :: SINGLE
```

For `REAL` and `COMPLEX`, selects the default real or default complex kind; this is equivalent to leaving the `KIND` selector off entirely.

```
INTEGER,PARAMETER :: DOUBLE
```

Selects the double precision real kind; this is equivalent to declaring `REAL` entities using the `DOUBLE PRECISION` type specifier, to declaring `COMPLEX` entities using `COMPLEX(KIND(0d0))`, and to using the exponent letter `D` on literal constants.

```
INTEGER,PARAMETER :: QUAD
```

`REAL`/`COMPLEX` kind selector for real and complex types with approximately twice the precision of `DOUBLE`. This might not be available on some systems; on a system without this type, the value of this parameter will be negative.

```
INTEGER,PARAMETER :: REAL16
```

`REAL`/`COMPLEX` kind selector for real and complex types that are represented using 16-bit floating-point numbers.

```
INTEGER,PARAMETER :: REAL32
```

`REAL`/`COMPLEX` kind selector for real and complex types that are represented using 32-bit floating-point numbers.

```
INTEGER,PARAMETER :: REAL64
```

`REAL`/`COMPLEX` kind selector for real and complex types that are represented using 64-bit floating-point numbers.

```
INTEGER,PARAMETER :: REAL64x2
```

REAL/COMPLEX kind selector for real and complex types that are represented using "double-double" floating-point numbers. A double-double floating-point number consists of two 64-bit values, one of which is at least DIGITS(1._REAL64) smaller than the other; this has almost twice the precision of REAL64 (except when near zero), but a smaller exponent range.

This type is not available on all systems; on a system without this type, the value of this parameter is −1.

```
INTEGER,PARAMETER :: REAL128
```

REAL/COMPLEX kind selector for real and complex types that are represented using 128-bit floating-point numbers. This will select a "true 128-bit" floating-point type if one is available, and if not it will select a "double-double" floating-point type if that is available; if no 128-bit floating-point type is available the value of this parameter is -1.

```
INTEGER,PARAMETER :: INT8
```

INTEGER kind selector for integer types with at least 8 bits of precision.

```
INTEGER,PARAMETER :: INT16
```

INTEGER kind selector for integer types with at least 16 bits of precision.

```
INTEGER,PARAMETER :: INT32
```

INTEGER kind selector for integer types with at least 32 bits of precision.

```
INTEGER,PARAMETER :: INT64
```

INTEGER kind selector for integer types with at least 64 bits of precision.

```
INTEGER,PARAMETER :: BYTE
```

LOGICAL kind selector for logical types occupying only one byte of memory.

```
INTEGER,PARAMETER :: TWOBYTE
```

LOGICAL kind selector for logical types occupying the same space as INTEGER(INT16) entities.

```
INTEGER,PARAMETER :: WORD
```

LOGICAL kind selector for a 32-bit logical type.

```
INTEGER,PARAMETER :: LOGICAL64
```

LOGICAL kind selector for a 64-bit logical type.

```
INTEGER,PARAMETER :: ASCII
```

CHARACTER kind selector for the ASCII character set.

```
INTEGER,PARAMETER :: JIS
```

CHARACTER kind selector for the JIS X 0213:2004 character set.

```
INTEGER,PARAMETER :: UCS2
```

CHARACTER kind selector for the UCS-2 (Unicode) character set.

```
INTEGER,PARAMETER :: UCS4
```

CHARACTER kind selector for the UCS-4 (ISO 10646) character set.

5 Notes

The source code for this module can be found in the NAG Fortran runtime library directory (on Unix this is usually '`/usr/local/lib/NAG_Fortran`' or '`/opt/NAG_Fortran/lib`', and on Windows it is usually '`C:\Program Files\NAG\EFBuilderPro\NAG_Fortran\lib`').

6 See Also

nagfor(1), **nag_modules**(3).

7 Bugs

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