

Standard intrinsic module ISO_FORTRAN_ENV

July 31, 2012

1 Name

`iso_fortran_env` — standard intrinsic module

2 Usage

```
USE, INTRINSIC :: ISO_FORTRAN_ENV
```

(The `INTRINSIC ::` part is optional.)

3 Synopsis

ISO_FORTRAN_ENV provides named constants describing the Fortran environment.

Storage units are described by `CHARACTER_STORAGE_SIZE` and `NUMERIC_STORAGE_SIZE`.

I/O units are described by `ERROR_UNIT`, `INPUT_UNIT` and `OUTPUT_UNIT`.

The `RECL=` specifier for the `INQUIRE` and `OPEN` statements is described by `FILE_STORAGE_SIZE`.

`IOSTAT=` return values are described by `IOSTAT_END` and `IOSTAT_EOR`. These and many other `IOSTAT=` return values are provided by the intrinsic module `F90_IOSTAT` (except that it provides `IOSTAT_EOF` instead of `IOSTAT_END`).

4 Parameter Descriptions

```
INTEGER, PARAMETER :: character_storage_size = 8
```

The size of a character storage unit in bits.

```
INTEGER, PARAMETER :: error_unit = 0
```

The standard error reporting unit number.

```
INTEGER, PARAMETER :: file_storage_size = 8
```

The size of a file storage unit (used by `RECL=` in `OPEN` and `INQUIRE`) in bits.

```
INTEGER, PARAMETER :: input_unit = 5
```

The standard input unit number. This is the one used by `READ` with an asterisk (`'*`') unit.

```
INTEGER, PARAMETER :: iostat_end = -1
```

The IOSTAT= return value for end of file.

```
INTEGER,PARAMETER :: iostat_eor = -2
```

The IOSTAT= return value for end of record.

```
INTEGER,PARAMETER :: numeric_storage_size = BIT_SIZE(0)
```

The size of a numeric storage unit in bits.

```
INTEGER,PARAMETER :: output_unit = 6
```

The standard output unit number. This is the one used by PRINT, and by WRITE with an asterisk ('*') unit.

5 Files

The source code for this module may be found in the NAG Fortran runtime library directory (usually /usr/local/lib/NAG_Fortran).

6 See Also

`f90_iostat(3)`, `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.

8 Author

Malcolm Cohen, Nihon Numerical Algorithms Group KK, Tokyo, Japan.