

# **f90\_unix\_errno: Unix Error Code Module**

March 15, 2019

## **1 Name**

**f90\_unix\_errno** — Module of Unix error codes

## **2 Usage**

### **USE F90\_UNIX\_ERRNO**

This module contains part of a Fortran API to functions detailed in ISO/IEC 9945-1:1990 Portable Operating System Interface (POSIX) - Part 1: System Application Program Interface (API) [C Language].

The facilities in this module are from Section 2.4 Error Numbers.

## **3 Synopsis**

### **Kind Parameter**

`ERROR_KIND.`

### **Error Parameters**

`E2BIG`, `EACCES`, `EAGAIN`, `EBADF`, `EBUSY`, `ECHILD`, `EDEADLK`, `EDOM`, `EEXIST`, `EFAULT`, `EFBIG`, `EINTR`, `EINVAL`, `EIO`, `EISDIR`, `EMFILE`, `EMLINK`, `ENAMETOOLONG`, `ENFILE`, `ENODEV`, `ENOENT`, `ENOEXEC`, `ENOLCK`, `ENOMEM`, `ENOSPC`, `ENOSYS`, `ENOTDIR`, `ENOTEMPTY`, `ENOTTY`, `ENXIO`, `EPERM`, `EPIPE`, `ERANGE`, `EROFS`, `ESPIPE`, `ESRCH`, `EXDEV`.

## **4 Error Handling**

Many procedures provided by the `F90_UNIX_*` modules have an optional `ERRNO` argument, declared as

```
INTEGER(error_kind),OPTIONAL,INTENT(OUT)
```

If this argument is provided it receives the error status from the procedure; zero indicates successful completion, otherwise it will be a non-zero error code, usually one of the ones listed in this module.

If the `ERRNO` argument is omitted and an error condition is raised, the program will be terminated with an informative error message.

If a procedure has no `ERRNO` argument it indicates that no error condition is possible - the procedure always succeeds.

## **5 Parameter Description**

All parameters are of type `INTEGER` with kind `ERROR_KIND`. The following table lists the error message typically associated with each error code; for full details see ISO/IEC 9945-1:1990, either section 2.4 or the appropriate section for the function raising the error.

E2BIG	Arg list too long
EACCES	Permission denied
EAGAIN	Resource temporarily unavailable
EBADF	Bad file descriptor
EBUSY	Resource busy
ECHILD	No child process
EDEADLK	Resource deadlock avoided
EDOM	Domain error
EEXIST	File exists
EFAULT	Bad address
EFBIG	File too large
EINTR	Interrupted function call
EINVAL	Invalid argument
EIO	Input/Output error
EISDIR	Is a directory
EMFILE	Too many open files
EMLINK	Too many links
ENAMETOOLONG	Filename too long
ENFILE	Too many open files in system
ENODEV	No such device
ENOENT	No such file or directory
ENOEXEC	Exec format error
ENOLCK	No locks available
ENOMEM	Not enough space
ENOSPC	No space left on device
ENOSYS	Function not implemented
ENOTDIR	Not a directory
ENOTEMPTY	Directory not empty
ENOTTY	Inappropriate I/O control operation
ENXIO	No such device or address
EPERM	Operation not permitted
EPIPE	Broken pipe
ERANGE	Result too large
EROFS	Read-only file system
ESPIPE	Invalid seek
ESRCH	No such process
EXDEV	Improper link

## 6 See Also

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

## 7 Bugs

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