

## 1 SUMMARY

HSL\_MP01 is a simple package that encapsulates all necessary include files for the use of MPI. It is intended to be used in place of an include 'mpif.h' directive, thus allowing other library MPI packages to be written in free source form.

**ATTRIBUTES** — **Version:** 1.1.0 (16 January 2023) **Types:** None. **Calls:** None. **Date:** July 2008. **Origin:** J. D. Hogg, Rutherford Appleton Laboratory. **Language:** Fortran 95. **Remark:** The development of HSL\_MP01 was supported by EPSRC grant EP/F006535/1.

## 2 HOW TO USE THE PACKAGE

### 2.1 Use of package

Access to the package requires a USE statement

```
USE HSL_MP01
```

This package supplies access to the constants of the version of MPI in use.

## 3 GENERAL INFORMATION

**Workspace:** None required. MPI implementations may manage their own memory.

**Other routines called directly:** None.

**Input/output:** None.

**Restrictions:** None.

**Portability:** Fortran 95.

## 4 METHOD

Inclusion of the Fortran 77 file mpif.h from some MPI implementations requires the file including it to be in fixed source form, this is undesirable for modern codes. This module aims to be the only part of the library where this is necessary, and other codes that require MPI merely need to use this one.

**Note:** Some newer implementations of MPI offer a Fortran 90 module file in addition to the include file mpif.h, however this is not sufficiently widely supported to justify its use in portable codes.

## 5 EXAMPLE OF USE

We give a basic example of a “hello world” code that uses this module

```
program hsl_mp01s
    use hsl_mp01
    implicit none

    integer :: i, p, np, ercode, recvд_p
    integer :: st(MPI_STATUS_SIZE)

    call MPI_INIT(ercode)

    call MPI_COMM_RANK(MPI_COMM_WORLD, p, ercode)
    call MPI_COMM_SIZE(MPI_COMM_WORLD, np, ercode)

    if( p .ne. 0 ) call MPI_Send(p,1,MPI_INTEGER,0,0,MPI_COMM_WORLD,ercode)

    if( p == 0 ) then
        print "(2(a,i4))", "Hello from processor ", p, " of ", np
        do i = 1, np-1
            call MPI_Recv(recvд_p,1,MPI_INTEGER,i,0,MPI_COMM_WORLD,st,ercode)
            print "(2(a,i4))", "Hello from processor ", recvд_p, " of ", np
        end do
    end if

    call MPI_FINALIZE(ercode)
end program hsl_mp01s
```

Sample output on 4 processors:

```
Hello from processor      0 of      4
Hello from processor      1 of      4
Hello from processor      2 of      4
Hello from processor      3 of      4
```