MPI_INIT()

```
int MPI_Init(int *argc, char ***argv)

MPI_INIT(IERROR)
    INTEGER IERROR

void MPI::Init(int& argc, char**& argv)

void MPI::Init()
```

This routine must be called before any other MPI routine. It must be called at most once; subsequent calls are erroneous (see MPI_INITIALIZED).

All MPI programs must contain a call to MPI_INIT; this routine must be called before any other MPI routine (apart from MPI_GET_VERSION, MPI_INITIALIZED, and MPI_FINALIZED) is called. The version for ISO C accepts the argc and argv that are provided by the arguments to `main`:

```
int main(int argc, char** argv)
{
    MPI_Init(&argc, &argv);

    /* parse arguments */
    /* main program    */

    MPI_Finalize();     /* see below */
}
```

The Fortran version takes only IERROR.

Conforming implementations of MPI are required to allow applications to pass NULL for both the argc and argv arguments of main in C and C++. In C++, there is an alternative binding for MPI::Init that does not have these arguments at all.

> *Rationale.* In some applications, libraries may be making the call to MPI_Init, and may not have access to argc and argv from main. It is anticipated that applications requiring special information about the environment or information supplied by `mpiexec` can get that information from environment variables. (*End of rationale.*)

MPI_FINALIZE()

```
int MPI_Finalize(void)

MPI_FINALIZE(IERROR)
    INTEGER IERROR

void MPI::Finalize()
```

This routine cleans up all MPI state. Each process must call MPI_FINALIZE before it exits. Unless there has been a call to MPI_ABORT, each process must ensure that all