before other MPI routines may be called. To provide for this, MPI includes an initialization routine MPI_INIT.

MPI_INIT()

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

```
MPI_INIT(IERROR)
    INTEGER IERROR
```

ticket150.

{`void MPI::Init(int& argc, char**& argv)` *(binding deprecated, see Section 15.2)* } ticket150.

{`void MPI::Init()` *(binding deprecated, see Section 15.2)* } ticket150.

ticket150.

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

ticket146.

[All MPI programs must contain a call to MPI_INIT; this routine must be called before any other MPI routine (apart from MPI-2.1 round-two - begin of modification `MPI_INITIALIZED`) MPI_GET_VERSION, MPI_INITIALIZED, and MPI_FINALIZED) MPI-2.1 round-two - end of modification is called. The version for MPI-2.1 Correction due to Reviews to MPI-2.1 draft Feb.23, 2008 ANSI C ISO C MPI-2.1 End of review based correction accepts the argc and argv that are provided by the arguments to `main`:] All MPI programs must contain exactly one call to an MPI initialization routine: MPI_INIT or MPI_INIT_THREAD. Subsequent calls to any initialization routines are erroneous. The only MPI functions that may be invoked before the MPI initialization routines are called are MPI_GET_VERSION, MPI_INITIALIZED, and MPI_FINALIZED. The version for ISO C accepts the argc and argv that are provided by the arguments to `main` or NULL: 146 ticket146.

```
int main([ticket60.]int argc, [ticket60.]char **argv)
[ticket60.][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()