

*Advice to users.* Under circumstances of store exhaustion an attempt to put a name of any length could fail, therefore the value of `MPI_MAX_OBJECT_NAME` should be viewed only as a strict upper bound on the name length, not a guarantee that setting names of less than this length will always succeed. (*End of advice to users.*)

*Advice to implementors.* Implementations which pre-allocate a fixed size space for a name should use the length of that allocation as the value of `MPI_MAX_OBJECT_NAME`. Implementations which allocate space for the name from the heap should still define `MPI_MAX_OBJECT_NAME` to be a relatively small value, since the user has to allocate space for a string of up to this size when calling `MPI_COMM_GET_NAME`. (*End of advice to implementors.*)

`MPI_COMM_GET_NAME` (`comm`, `comm_name`, `resultlen`)

IN	<code>comm</code>	communicator whose name is to be returned (handle)
OUT	<code>comm_name</code>	the name previously stored on the communicator, or an empty string if no such name exists (string)
OUT	<code>resultlen</code>	length of returned name (integer)

`int MPI_Comm_get_name(MPI_Comm comm, char *comm_name, int *resultlen)`

`MPI_COMM_GET_NAME(COMM, COMM_NAME, RESULTLEN, IERROR)`

INTEGER COMM, RESULTLEN, IERROR

CHARACTER\*(\*) COMM\_NAME

`void MPI::Comm::Get_name(char* comm_name, int& resultlen) const`

`MPI_COMM_GET_NAME` returns the last name which has previously been associated with the given communicator. The name may be set and got from any language. The same name will be returned independent of the language used. `name` should be allocated so that it can hold a resulting string of length `MPI_MAX_OBJECT_NAME` characters.

`MPI_COMM_GET_NAME` returns a copy of the set name in `name`. In C, a null character is additionally stored at `name[resultlen]`. **resultlen cannot be larger than MPI\_MAX\_OBJECT\_NAME-1.** In Fortran, `name` is padded on the right with blank characters. **resultlen cannot be larger than MPI\_MAX\_OBJECT\_NAME.**

If the user has not associated a name with a communicator, or an error occurs, `MPI_COMM_GET_NAME` will return an empty string (all spaces in Fortran, "" in C and C++). The three predefined communicators will have predefined names associated with them. Thus, the names of `MPI_COMM_WORLD`, `MPI_COMM_SELF`, and the communicator returned by `MPI_COMM_GET_PARENT` (if not `MPI_COMM_NULL`) will have the default of `MPI_COMM_WORLD`, `MPI_COMM_SELF`, and `MPI_COMM_PARENT`. The fact that the system may have chosen to give a default name to a communicator does not prevent the user from setting a name on the same communicator; doing this removes the old name and assigns the new one.

*Rationale.* We provide separate functions for setting and getting the name of a communicator, rather than simply providing a predefined attribute key for the following reasons: