Chapter 16

Error Handling

CMMD provides two functions for handling CMMD-defined errors, and one function for handling user-defined errors:

- `CMMD_get_errno` returns the value of the CMMD internal error variable, in cases where a previous function call has returned `CMMD_ERRVAL`. The value returned by `CMMD_get_errno` can then be passed to the function `CMMD_error_string`, for cases in which the user prefers to receive an explanatory character string rather than an integer.
- `CMMD_error_string` takes a value returned by `CMMD_get_errno` as input and returns a system-supplied character string describing the error.
- `CMMD_error` displays a user-supplied error message and terminates the program.

This chapter provides the reference pages for these three functions.

Copyright © 1995 Thinking Machines Corporation

CMMD_get_errno

Returns the current error set by CMMD, after some function has returned CMMD_ERRVAL.

SYNOPSIS

C Syntax

```
int CMMD_get_errno(void)
```

Fortran Syntax

```
INTEGER FUNCTION CMMD_get_errno()
```

RETURNS

CMMD_get_errno returns the current value of the internal variable that is set when a function returns CMMD_ERRVAL.

Possible values are as follows:

CMMD_ERR_BAD_PROTOCOL
: A channel ID or MCB provided to a CMMD function referred to an rport allocated to the wrong protocol.

CMMD_ERR_BAD_STATE
: A user command attempted a state change on a channel or message when the channel or message was not in the required preceding state.

CMMD_ERR_ID_OUT_OF_RANGE
: A channel ID or MCB provided as argument to a CMMD function was out of the allowable range

CMMD_ERR_NO_CHANNELS
: CMMD has run out of channels to allocate.

CMMD_ERR_NOT_ALLOCATED
: A channel ID or MCB provided as an argument to a CMMD function referred to an rport that has not been allocated.

Version 3.3, April 1995
Copyright © 1995 Thinking Machines Corporation

`CMMD_ERR_Q`	An internal error occurred. You should never see this error value.

DESCRIPTION

Some CMMD functions may return `CMMD_ERRVAL` if an error occurs. When this happens, CMMD sets an internal variable to a particular value. The function `CMMD_get_errno` returns the value of this internal variable. The application has the choice of using the value returned by `CMMD_get_errno` or of passing that value to another function, `CMMD_error_string`. `CMMD_error_string` takes a value from `CMMD_get_errno` as input and returns a character string describing the error in question.

Copyright © 1995 Thinking Machines Corporation

CMMD_error_string

Returns the error string that matches a given error value.

SYNOPSIS

C Syntax

```
char *CMMD_error_string(int errno)
```

Fortran Syntax

```
EXTERNAL CMMD_error_string
    CHARACTER*(512) CMMD_error_string
```

RETURNS

Given a value returned by `CMMD_get_errno`, `CMMD_error_string` returns a character string describing the error in question.

Version 3.3, April 1995
Copyright © 1995 Thinking Machines Corporation

CMMD_error

Signals an error, prints a user-supplied error message.

SYNOPSIS

C Syntax

```
void CMMD_error(const char *format, args ...)
```

Fortran Syntax

```
SUBROUTINE CMMD_error(string)
    CHARACTER* string
```

ARGUMENTS

For the C syntax, the arguments are similar to arguments for `printf`.
For Fortran, the only allowable argument is a character string.

DESCRIPTION

`CMMD_error` signals errors from user programs. It displays the error message specified by its argument(s), and then signals an error, terminating the program. It does not return. (Thus, calling `CMMD_error` is equivalent to calling `abort` in a UNIX program.)

Normally, a call to `CMMD_error` will cause the system to dump CMOST core and error files before the program exits. The dumping of these files may be disabled by setting the environment variable `CM_NO_PN_CORE`.

Copyright © 1995 Thinking Machines Corporation