Chapter 13

Error Handling

CMMD provides two functions for handling errors:

- **CMMD_get_errno** returns the value of the CMMD internal error variable, in cases where a previous function call has returned **CMMD_ERRVAL.**
- **CMMD_error** displays a user-supplied error messages and terminates the program.

13.1 Retrieving an Error Value

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.

FUNCTION **CMMD_get_errno**

C SYNTAX

int **CMMD_get_errno(**void**)**

FORTRAN SYNTAX

INTEGER FUNCTION **CMMD_get_errno()**

RETURNS

CMMD_get_errno returns the current value of the current value of the internal variable 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.

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

13.2 Printing Error Messages

The function **CMMD_error** can be used to signal errors from user programs. Its arguments are similar to those of **printf**, that is:

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

CMMD_error displays the error message determined by the *format* string and the *args*, and then signals an error, terminating the program. It does not return. (In other words, **CMMD_error** is the equivalent of calling **abort** in a UNIX program.)