4.3.4 MCAPI_MSG_RECV

NAME

`mcapi_msg_recv` – receives a (connectionless) message from a receive endpoint.

SYNOPSIS

```
#include <mcapi.h>

void mcapi_msg_recv(
     MCAPI_IN mcapi_endpoint_t  receive_endpoint,
     MCAPI_OUT void* buffer,
     MCAPI_IN size_t buffer_size,
     MCAPI_OUT size_t* received_size,
     MCAPI_OUT mcapi_status_t* mcapi_status
);
```

DESCRIPTION

Receives a (connectionless) message from a receive endpoint. It is a blocking function, and returns once a message is available and the received data filled into the buffer. `receive_endpoint` is a local endpoint identifying the receive endpoint. `buffer` is the application provided buffer, and `buffer_size` is the buffer size in bytes. The `received_size` parameter is filled with the actual size of the received message. This method will block if there is insufficient memory space available. When sufficient space becomes available, the function will complete.

RETURN VALUE

On success, `*mcapi_status` is set to `MCAPI_SUCCESS`. On error, `*mcapi_status` is set to the appropriate error defined below.

ERRORS

`MCAPI_ERR_NODE_NOTINIT`	The node is not initialized.
`MCAPI_ERR_ENDP_INVALID`	Argument is not a valid local endpoint descriptor.
`MCAPI_ERR_MSG_TRUNCATED`	The message size exceeds the buffer_size.
`MCAPI_ERR_TRANSMISSION`	Transmission failure. This error code is optional, and if supported by an implementation, its functionality shall be described.
`MCAPI_ERR_PARAMETER`	Incorrect `buffer` and/or `received_size` parameter.
`MCAPI_TIMEOUT`	The operation timed out. Implementations can optionally support timeout for this function. The timeout value is set with endpoint attributes.