4.6.2 MCAPI_WAIT

NAME

`mcapi_wait` – waits for a non-blocking operation to complete.

SYNOPSIS

```
#include <mcapi.h>

mcapi_boolean_t mcapi_wait(
     MCAPI_IN mcapi_request_t* request,
     MCAPI_OUT size_t* size,
     MCAPI_IN mcapi_timeout_t timeout,
     MCAPI_OUT mcapi_status_t* mcapi_status
);
```

DESCRIPTION

Wait until a non-blocking operation has completed. It is a blocking function and returns when the operation has completed, has been canceled, or a timeout has occurred. `request` is the identifier for the non-blocking operation. The call only waits for the completion of an operation (all buffers referenced in the operation have been filled or consumed and can now be safely accessed by the application) and doesn't affect any messages/packets/scalars. The `size` parameter is set to the number of bytes that were either sent or received by the non-blocking transaction that completed (size is irrelevant for non-blocking connect and close calls). The `mcapi_wait()` call will return if the request is cancelled by a call to `mcapi_cancel()`, and the returned `mcapi_status` will indicate that the request was cancelled. The units for `timeout` are implementation defined. If a timeout occurs the returned status will indicate that the timeout occurred. A value of 0 for the `timeout` parameter indicates that the function will return without blocking with failure or success and a value of `MCAPI_INFINITE` for the `timeout` parameter indicates no timeout is requested, i.e. the function will block until it is unblocked because of failure or success. In regards to timeouts `mcapi_wait` is non destructive, i.e. the request is not cleared in the case of a timeout and can be waited upon multiple times. Multiple threads attempting to wait on the same request will result in an error.

RETURN VALUE

On success, `MCAPI_TRUE` is returned and `*mcapi_status` is set to `MCAPI_SUCCESS`. On error or timeout `MCAPI_FALSE` is returned and `*mcapi_status` is set to the appropriate error/status defined below for parameter errors for the `mcapi_wait()` call or an error code for the corresponding non-blocking routine associated with the request structure.

ERRORS

`MCAPI_ERR_NODE_NOTINIT`	The node is not initialized.
`MCAPI_ERR_REQUEST_INVALID`	Argument is not a valid request handle.
`MCAPI_ERR_REQUEST_CANCELLED`	The request was canceled, by another thread (during the waiting).