

### 4.6.1 MCAPI\_TEST

#### NAME

`mcapi_test` – Tests if a non-blocking operation has completed.

#### SYNOPSIS

```
#include <mcapi.h>

mcapi_boolean_t mcapi_test(
    MCAPI_IN mcapi_request_t* request,
    MCAPI_OUT size_t* size,
    MCAPI_OUT mcapi_status_t* mcapi_status
);
```

#### DESCRIPTION

Checks if a non-blocking operation has completed. The function returns in a timely fashion. `request` is the identifier for the non-blocking operation. The call only checks the completion of an operation and doesn't affect any messages/packets/scalars and does not release the request. If the specified request completes and the pending operation was a send or receive operation, the `size` parameter is set to the number of bytes that were either sent or received by the non-blocking transaction. The request is released with either of `mcapi_wait()`, `mcapi_wait_any()`, or `mcapi_cancel()`.

#### RETURN VALUE

On success, `MCAPI_TRUE` is returned and `*mcapi_status` is set to `MCAPI_SUCCESS`. If the operation has not completed `MCAPI_FALSE` is returned and `*mcapi_status` is set to `MCAPI_PENDING`. On error `MCAPI_FALSE` is returned and `*mcapi_status` is set to the appropriate error defined below or an error code for the corresponding non-blocking routine associated with the request structure.

#### ERRORS

<code>MCAPI_ERR_NODE_NOTINIT</code>	The node is not initialized.
<code>MCAPI_ERR_REQUEST_INVALID</code>	Argument is not a valid request handle. This also applies if the request has been canceled.
<code>MCAPI_ERR_PARAMETER</code>	Incorrect size parameter.
<code>MCAPI_ERR_GENERAL</code>	Implementation specific error not covered by other status codes. Specifics must be documented.

#### SEE ALSO

```
mcapi_endpoint_t mcapi_endpoint_getmcapi_endpoint_get_i(),
mcapi_msg_send_i(), mcapi_msg_rcv_i(), mcapi_pktchan_connect_i(),
```