4.2.2 MCAPI_ENDPOINT_GET_I

NAME

`mcapi_endpoint_get_i` – Obtain the endpoint associated with a given tuple.

SYNOPSIS

```
#include <mcapi.h>


void  mcapi_endpoint_get_i(
     MCAPI_IN mcapi_domain_t domain_id,
     MCAPI_IN mcapi_node_t node_id,
     MCAPI_IN mcapi_port_t port_id,
     MCAPI_OUT mcapi_endpoint_t* endpoint,
     MCAPI_OUT mcapi_request_t* request,
     MCAPI_OUT mcapi_status_t* mcapi_status
);
```

DESCRIPTION

`mcapi_endpoint_get_i()` allows other nodes (“third parties”) to get the endpoint identifier for the endpoint associated with a global tuple name <`domain_id, node_id, port_id`>. This function is non-blocking and will return immediately. An endpoint can receive messages from a multitude of other endpoints. Message type endpoints can therefore have multiple outstanding endpoint gets from other nodes. Channel connected endpoints on the other hand has a one to one relationship. Channel type endpoints can therefore only have one outstanding get. A second get on a channel type endpoint will result in an error.

RETURN VALUE

On success, `*mcapi_status` is set to `MCAPI_SUCCESS` if completed and `MCAPI_PENDING` if not yet completed. On error, `*mcapi_status` is set to the appropriate error defined below.

ERRORS

`MCAPI_ERR_PORT_INVALID`	The parameter is not a valid port. This error also covers endpoints without ports.
`MCAPI_ERR_NODE_INVALID`	The parameter is not a valid node.
`MCAPI_ERR_NODE_NOTINIT`	The node is not initialized.
`MCAPI_ERR_DOMAIN_INVALID`	The parameter is not a valid domain.
`MCAPI_ERR_REQUEST_LIMIT`	No more request handles available.
`MCAPI_ERR_ENDP_GET_LIMIT`	The endpoint get reference count is too high. Use service function to check this status.