4.3.1.7 MRAPI_MUTEX_LOCK

NAME

mrapi_mutex_lock

SYNOPSIS

```
void mrapi_mutex_lock (
  MRAPI_IN mrapi_mutex_hndl_t mutex,
  MRAPI_OUT mrapi_key_t* lock_key,
  MRAPI_IN mrapi_timeout_t timeout,
  MRAPI_OUT mrapi_status_t* status
);
```

DESCRIPTION

This function attempts to lock a mutex and will block if another node has a lock on the mutex. When it obtains the lock, it sets up a unique key for that lock and that key is to be passed back on the call to unlock. This key allows us to support recursive locking. The `lock_key` is only valid if status indicates success. Whether or not a mutex can be locked recursively is controlled via the `MRAPI_MUTEX_RECURSIVE` attribute, and the default is `MRAPI_FALSE`.

RETURN VALUE

On success, `*status` is set to `MRAPI_SUCCESS`. On error, `*status` is set to the appropriate error defined below. When extended error checking is enabled, if this function is called on a mutex that no longer exists, an `MRAPI_ERR_MUTEX_DELETED` error code will be returned. When extended error checking is disabled, the `MRAPI_ERR_MUTEX_INVALID` error will be returned.

ERRORS

`MRAPI_ERR_MUTEX_INVALID`	Argument is not a valid mutex handle.
`MRAPI_ERR_MUTEX_LOCKED`	Mutex is already locked by another node or mutex is already locked by this node and is not a recursive mutex.
`MRAPI_ERR_MUTEX_DELETED`	If the mutex has been deleted then if `MRAPI_ERROR_EXT` attribute is set, MRAPI will return `MRAPI_ERR_MUTEX_DELETED` otherwise MRAPI will just return `MRAPI_ERR_MUTEX_INVALID`.
`MRAPI_TIMEOUT`	Timeout was reached.
`MRAPI_ERR_PARAMETER`	Invalid lock_key or timeout parameter.
`MRAPI_ERR_NODE_NOTINIT`	The calling node is not intialized.

NOTE

SEE ALSO