4.3.1.9 MRAPI_MUTEX_UNLOCK

NAME

mrapi_mutex_unlock

SYNOPSIS

```
void mrapi_mutex_unlock(
  MRAPI_IN mrapi_mutex_hndl_t mutex,
  MRAPI_IN mrapi_key_t* lock_key,
  MRAPI_OUT mrapi_status_t* status
);
```

DESCRIPTION

This function unlocks a mutex. If the mutex is recursive, then the `lock_key` parameter passed in must match the `lock_key` that was returned by the corresponding call to lock the mutex, and the set of recursive locks must be released using `lock_keys` in the reverse order that they were obtained. 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.

RETURN VALUE

On success, `*status` is set to `MRAPI_SUCCESS`. On error, `*status` is set to the appropriate error defined below.

ERRORS

`MRAPI_ERR_MUTEX_INVALID`	Argument is not a valid mutex handle.
`MRAPI_ERR_MUTEX_NOTLOCKED`	Mutex is not locked.
`MRAPI_ERR_MUTEX_KEY`	`lock_key` is invalid for this mutex.
`MRAPI_ERR_MUTEX_LOCKORDER`	The unlock call does not match the lock order for this recursive mutex.
`MRAPI_ERR_PARAMETER`	Invalid `lock_key` parameter.
`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_ERR_NODE_NOTINIT`	The calling node is not intialized.

NOTE

SEE ALSO