4.3.2.7 MRAPI_SEM_LOCK

NAME

mrapi_sem_lock

SYNOPSIS

```
void mrapi_sem_lock(
  MRAPI_IN mrapi_sem_hndl_t sem,
  MRAPI_IN mrapi_timeout_t timeout,
  MRAPI_OUT mrapi_status_t* status
);
```

DESCRIPTION

This function attempts to obtain a single lock on the semaphore and will block until a lock is available or the timeout is reached (if timeout is non-zero). If the request can’t be satisfied for some other reason, this function will return the appropriate error code below. An application may make this call as many times as needed to obtain multiple locks, up to the limit specified by the `shared_lock_limit` parameter used when the semaphore was created.

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 lock is called on semaphore that no longer exists, an `MRAPI_ERR_SEM_DELETED` error code will be returned. When extended error checking is disabled, the `MRAPI_ERR_SEM_INVALID` error will be returned and the lock will fail.

ERRORS

`MRAPI_ERR_SEM_INVALID`	Argument is not a valid semaphore handle.
`MRAPI_ERR_SEM_DELETED`	If the semaphore has been deleted then if `MRAPI_ERROR_EXT` attribute is set, MRAPI will return `MRAPI_ERR_SEM_DELETED` otherwise MRAPI will just return `MRAPI_ERR_SEM_INVALID`.
`MRAPI_TIMEOUT`	Timeout was reached.
`MRAPI_ERR_NODE_NOTINIT`	The calling node is not intialized.

NOTE

SEE ALSO