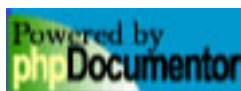


patSHMC



Contents

Package patSHMC Procedural Elements	2
patSHMC.php	2
Define patEXCLUSIVE_LOCK	2
Define patRELEASE_LOCK	2
Define patSHARED_LOCK	2
Package patSHMC Classes	3
Class patSHMC	3
Constructor patSHMC	3
Method detach	3
Method get_var	4
Method lock	4
Method put_var	4
Method remove	5
Method remove_var	5
Appendices	6
Appendix A - Class Trees	7
patSHMC	7
Appendix D - Todo List	8

Package patSHMC Procedural Elements

patSHMC.php

- **Package** patSHMC

patEXCLUSIVE_LOCK = 2 [*line 4*]
patRELEASE_LOCK = 3 [*line 5*]
patSHARED_LOCK = 1 [*line 3*]

Package patSHMC Classes

Class patSHMC

[line 16]

patSHMC - Shared Memory Connectivity This class can be used to easily handle shared memory including semaphore locking.

- **Package** patSHMC
- **TODO** Shared Locking (using a second semaphore), Error Handling
- **Author** Stephan Schmidt < schst@php-tools.de>
- **Version** 0.1

Constructor function patSHMC::patSHMC(\$shm_key, \$shm_size, \$sem_key, [\$perm = 438]) *[line 28]*

Function Parameters:

- *integer* **\$shm_key** unique key for the shared memory
- *integer* **\$shm_size** size of the shared memory in bytes
- *integer* **\$sem_key** unique key for the semaphore that is used for locking
- *integer* **\$perm** permissions

create a new shared memory connectivity

function patSHMC::detach() *[line 104]*

disconnect from the shared memory

- **Access** public

mixed function patSHMC::get_var(\$key) [*line 83*]

Function Parameters:

- *integer* **\$key** unique key for the variable

read a variable from the shared memory

- **Access** public

function patSHMC::lock(\$operation) [*line 50*]

Function Parameters:

- *integer* **\$operation** 2 = exclusive lock, 3 = free lock

lock or unlock the shared memory

- **Access** public

function patSHMC::put_var(\$key, \$value) [*line 71*]

Function Parameters:

- *integer* **\$key** unique key for the variable
- *mixed* **\$value** value of the variable

write a variable to the shared memory

- **Access** public

function patSHMC::remove() [*line 114*]
remove (clear) shared memory

- **Access** public

function patSHMC::remove_var(\$key) [*line 94*]
Function Parameters:

- *integer* **\$key** unique key for the variable

remove a variable from the shared memory

- **Access** public

Appendices

Appendix A - Class Trees

Package patSHMC

patSHMC

- [patSHMC](#)

Appendix D - Todo List

In Package patSHMC

In [patSHMC](#):

- Shared Locking (using a second semaphore), Error Handling

Index

C

constructor <code>patSHMC::patSHMC()</code>	3
<i>create a new shared memory connectivity</i>	

P

patSHMC::lock()	4
<i>lock or unlock the shared memory</i>	
patSHMC::put_var()	4
<i>write a variable to the shared memory</i>	
patSHMC::remove()	5
<i>remove (clear) shared memory</i>	
patSHMC::remove_var()	5
<i>remove a variable from the shared memory</i>	
patSHMC::get_var()	4
<i>read a variable from the shared memory</i>	
patSHMC::detach()	3
<i>disconnect from the shared memory</i>	
patEXCLUSIVE_LOCK	2
patRELEASE_LOCK	2
patSHARED_LOCK	2
patSHMC	3
<i>patSHMC - Shared Memory Connectivity</i>	
<i>This class can be used to easily handle shared memory including semaphore locking.</i>	
patSHMC.php	2