



**packetvideo™**

OSCL API

Build Version: OPENCORE\_20090517

May 17, 2009

# Contents

|   |            |
|---|------------|
| <b>1 oscl Module Index</b>                        | <b>1</b>   |
| 1.1 oscl Modules . . . . .                        | 1          |
| <b>2 oscl Hierarchical Index</b>                  | <b>2</b>   |
| 2.1 oscl Class Hierarchy . . . . .                | 2          |
| <b>3 oscl Data Structure Index</b>                | <b>8</b>   |
| 3.1 oscl Data Structures . . . . .                | 8          |
| <b>4 oscl File Index</b>                          | <b>14</b>  |
| 4.1 oscl File List . . . . .                      | 14         |
| <b>5 oscl Module Documentation</b>                | <b>19</b>  |
| 5.1 OSCL config . . . . .                         | 19         |
| 5.2 OSCL Base . . . . .                           | 23         |
| 5.3 OSCL Memory . . . . .                         | 44         |
| 5.4 OSCL Util . . . . .                           | 60         |
| 5.5 OSCL Error . . . . .                          | 81         |
| 5.6 OSCL IO . . . . .                             | 91         |
| 5.7 OSCL Proc . . . . .                           | 99         |
| 5.8 OSCL Init . . . . .                           | 103        |
| <b>6 oscl Data Structure Documentation</b>        | <b>104</b> |
| 6.1 _OsclBasicAllocator Class Reference . . . . . | 104        |
| 6.2 _OsclHeapBase Class Reference . . . . .       | 106        |
| 6.3 AcceptParam Class Reference . . . . .         | 108        |
| 6.4 allocator Class Reference . . . . .           | 109        |
| 6.5 AllPassFilter Class Reference . . . . .       | 110        |
| 6.6 BindParam Class Reference . . . . .           | 112        |
| 6.7 BufferFragment Class Reference . . . . .      | 113        |

|      |  |     |
|------|--|-----|
| 6.8  | BufferMgr Class Reference . . . . .  | 114 |
| 6.9  | BufferState Class Reference . . . . .  | 115 |
| 6.10 | BufFragGroup< ChainClass, max_frags > Class Template Reference . . . . .             | 116 |
| 6.11 | BufFragStatusClass Class Reference . . . . .   | 119 |
| 6.12 | CallbackTimer< Alloc > Class Template Reference . . . . .                            | 120 |
| 6.13 | CallbackTimerObserver Class Reference . . . . .                                      | 122 |
| 6.14 | CFastRep Class Reference . . . . .   | 123 |
| 6.15 | CHheapRep Class Reference . . . . .  | 125 |
| 6.16 | ConnectParam Class Reference . . . . .   | 127 |
| 6.17 | CStackRep Class Reference . . . . .  | 128 |
| 6.18 | DNSRequestParam Class Reference . . . . .  | 129 |
| 6.19 | GetHostByNameParam Class Reference . . . . .   | 131 |
| 6.20 | HeapBase Class Reference . . . . .   | 132 |
| 6.21 | internalLeave Class Reference . . . . .  | 134 |
| 6.22 | LinkedListElement< LLClass > Class Template Reference . . . . .                      | 135 |
| 6.23 | ListenParam Class Reference . . . . .  | 136 |
| 6.24 | MediaData< ChainClass, max_frags, local_bufsize > Class Template Reference . . . . . | 137 |
| 6.25 | MediaStatusClass Class Reference . . . . .   | 140 |
| 6.26 | MemAllocator< T > Class Template Reference . . . . .                                 | 141 |
| 6.27 | MM_AllocBlockFence Struct Reference . . . . .  | 142 |
| 6.28 | MM_AllocBlockHdr Struct Reference . . . . .  | 143 |
| 6.29 | MM_AllocInfo Struct Reference . . . . .  | 144 |
| 6.30 | MM_AllocNode Struct Reference . . . . .  | 146 |
| 6.31 | MM_AllocQueryInfo Struct Reference . . . . .   | 147 |
| 6.32 | MM_Audit_Imp Class Reference . . . . .   | 148 |
| 6.33 | MM_AuditOverheadStats Struct Reference . . . . .                                     | 156 |
| 6.34 | MM_FailInsertParam Struct Reference . . . . .  | 157 |
| 6.35 | MM_Stats_CB Struct Reference . . . . .   | 158 |
| 6.36 | MM_Stats_t Struct Reference . . . . .  | 159 |
| 6.37 | NTPTTime Class Reference . . . . .   | 161 |
| 6.38 | Oscl_Alloc Class Reference . . . . .   | 165 |
| 6.39 | Oscl_Dealloc Class Reference . . . . .   | 166 |
| 6.40 | Oscl_DefAlloc Class Reference . . . . .  | 167 |
| 6.41 | Oscl_DefAllocWithRefCounter< DefAlloc > Class Template Reference . . . . .           | 168 |
| 6.42 | OSCL_FastString Class Reference . . . . .  | 170 |
| 6.43 | Oscl_File Class Reference . . . . .  | 174 |

|  |     |
|--|-----|
| 6.44 Oscl_File::OsclCacheObserver Class Reference . . . . .                                    | 182 |
| 6.45 Oscl_File::OsclFixedCacheParam Class Reference . . . . .                                  | 183 |
| 6.46 Oscl_FileFind Class Reference . . . . .   | 184 |
| 6.47 Oscl_FileServer Class Reference . . . . .   | 188 |
| 6.48 oscl_fsstat Struct Reference . . . . .  | 190 |
| 6.49 OSCL_HeapString< Alloc > Class Template Reference . . . . .                               | 191 |
| 6.50 OSCL_HeapStringA Class Reference . . . . .  | 193 |
| 6.51 Oscl_Int64_Utils Class Reference . . . . .  | 197 |
| 6.52 Oscl_Less< T > Struct Template Reference . . . . .  | 199 |
| 6.53 Oscl_Linked_List< LLClass, Alloc > Class Template Reference . . . . .                     | 200 |
| 6.54 Oscl_Linked_List_Base Class Reference . . . . .   | 204 |
| 6.55 Oscl_Map< Key, T, Alloc, Compare > Class Template Reference . . . . .                     | 208 |
| 6.56 Oscl_Map< Key, T, Alloc, Compare >::value_compare Class Reference . . . . .               | 215 |
| 6.57 Oscl_MTLinked_List< LLClass, Alloc, TheLock > Class Template Reference . . . . .          | 217 |
| 6.58 Oscl_Opaque_Type_Alloc Class Reference . . . . .  | 221 |
| 6.59 Oscl_Opaque_Type_Alloc_LL Class Reference . . . . .                                       | 222 |
| 6.60 Oscl_Opaque_Type_Compare Class Reference . . . . .  | 224 |
| 6.61 Oscl_Pair< T1, T2 > Struct Template Reference . . . . .                                   | 226 |
| 6.62 Oscl_Queue< T, Alloc > Class Template Reference . . . . .                                 | 227 |
| 6.63 Oscl_Queue_Base Class Reference . . . . .   | 230 |
| 6.64 Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > Class Template Reference . . . . . | 233 |
| 6.65 Oscl_Rb_Tree_Base Class Reference . . . . .   | 237 |
| 6.66 Oscl_Rb_Tree_Const_Iterator< Value > Struct Template Reference . . . . .                  | 238 |
| 6.67 Oscl_Rb_Tree_Iterator< Value > Struct Template Reference . . . . .                        | 241 |
| 6.68 Oscl_Rb_Tree_Node< Value > Struct Template Reference . . . . .                            | 244 |
| 6.69 Oscl_Rb_Tree_Node_Base Struct Reference . . . . .   | 245 |
| 6.70 Oscl_Select1st< V, U > Struct Template Reference . . . . .                                | 247 |
| 6.71 OSCL_StackString< MaxBufSize > Class Template Reference . . . . .                         | 248 |
| 6.72 oscl_stat_buf Struct Reference . . . . .  | 250 |
| 6.73 OSCL_String Class Reference . . . . .   | 251 |
| 6.74 Oscl_Tag< Alloc > Struct Template Reference . . . . .                                     | 256 |
| 6.75 Oscl_Tag_Base Struct Reference . . . . .  | 258 |
| 6.76 Oscl_TagTree< T, Alloc > Class Template Reference . . . . .                               | 260 |
| 6.77 Oscl_TagTree< T, Alloc >::const_iterator Struct Reference . . . . .                       | 264 |
| 6.78 Oscl_TagTree< T, Alloc >::iterator Struct Reference . . . . .                             | 267 |
| 6.79 Oscl_TagTree< T, Alloc >::Node Struct Reference . . . . .                                 | 270 |

|  |     |
|--|-----|
| 6.80 Oscl_TAlloc< T, Alloc > Class Template Reference . . . . .                  | 272 |
| 6.81 Oscl_TAlloc< T, Alloc >::rebind< U, V > Struct Template Reference . . . . . | 275 |
| 6.82 Oscl_Vector< T, Alloc > Class Template Reference . . . . .                  | 276 |
| 6.83 Oscl_Vector_Base Class Reference . . . . .                                  | 281 |
| 6.84 OSCL_wFastString Class Reference . . . . .                                  | 285 |
| 6.85 OSCL_wHeapString< Alloc > Class Template Reference . . . . .                | 288 |
| 6.86 OSCL_wHeapStringA Class Reference . . . . .                                 | 290 |
| 6.87 OSCL_wStackString< MaxBufSize > Class Template Reference . . . . .          | 293 |
| 6.88 OSCL_wString Class Reference . . . . .                                      | 295 |
| 6.89 OsclAcceptMethod Class Reference . . . . .                                  | 299 |
| 6.90 OsclAcceptRequest Class Reference . . . . .                                 | 300 |
| 6.91 OsclActiveObject Class Reference . . . . .                                  | 301 |
| 6.92 OsclAllocDestructDealloc Class Reference . . . . .                          | 305 |
| 6.93 OsclAOStatus Class Reference . . . . .                                      | 306 |
| 6.94 OsclAsyncFile Class Reference . . . . .                                     | 307 |
| 6.95 OsclAsyncFileBuffer Class Reference . . . . .                               | 310 |
| 6.96 OsclAuditCB Class Reference . . . . .                                       | 312 |
| 6.97 OsclBindMethod Class Reference . . . . .                                    | 313 |
| 6.98 OsclBindRequest Class Reference . . . . .                                   | 314 |
| 6.99 OsclBinIStream Class Reference . . . . .                                    | 315 |
| 6.100 OsclBinIStreamBigEndian Class Reference . . . . .                          | 317 |
| 6.101 OsclBinIStreamLittleEndian Class Reference . . . . .                       | 320 |
| 6.102 OsclBinOStream Class Reference . . . . .                                   | 322 |
| 6.103 OsclBinOStreamBigEndian Class Reference . . . . .                          | 323 |
| 6.104 OsclBinOStreamLittleEndian Class Reference . . . . .                       | 325 |
| 6.105 OsclBinStream Class Reference . . . . .                                    | 327 |
| 6.106 OsclBuf Class Reference . . . . .  | 331 |
| 6.107 OsclCompareLess< T > Class Template Reference . . . . .                    | 333 |
| 6.108 OsclComponentRegistry Class Reference . . . . .                            | 334 |
| 6.109 OsclComponentRegistryData Class Reference . . . . .                        | 336 |
| 6.110 OsclComponentRegistryElement Class Reference . . . . .                     | 337 |
| 6.111 OsclConnectMethod Class Reference . . . . .                                | 339 |
| 6.112 OsclConnectRequest Class Reference . . . . .                               | 340 |
| 6.113 OsclDestructDealloc Class Reference . . . . .                              | 341 |
| 6.114 OsclDNS Class Reference . . . . .  | 342 |
| 6.115 OsclDNSI Class Reference . . . . .   | 344 |

|   |     |
|---|-----|
| 6.116OsclDNSIBase Class Reference . . . . .                           | 346 |
| 6.117OsclDNSMethod Class Reference . . . . .                          | 349 |
| 6.118OsclDNSObserver Class Reference . . . . .                        | 352 |
| 6.119OsclDNSRequest Class Reference . . . . .                         | 353 |
| 6.120OsclDNSRequestAO Class Reference . . . . .                       | 354 |
| 6.121OsclDoubleLink Class Reference . . . . .                         | 357 |
| 6.122OsclDoubleList< T > Class Template Reference . . . . .           | 358 |
| 6.123OsclDoubleListBase Class Reference . . . . .                     | 359 |
| 6.124OsclDoubleRunner< T > Class Template Reference . . . . .         | 361 |
| 6.125OsclError Class Reference . . . . .                              | 363 |
| 6.126OsclErrorAllocator Class Reference . . . . .                     | 365 |
| 6.127OsclErrorTrap Class Reference . . . . .                          | 367 |
| 6.128OsclErrorTrapImp Class Reference . . . . .                       | 368 |
| 6.129OsclException< LeaveCode > Class Template Reference . . . . .    | 370 |
| 6.130OsclExclusiveArrayPtr< T > Class Template Reference . . . . .    | 371 |
| 6.131OsclExclusivePtr< T > Class Template Reference . . . . .         | 374 |
| 6.132OsclExclusivePtrA< T, Alloc > Class Template Reference . . . . . | 377 |
| 6.133OsclExecScheduler Class Reference . . . . .                      | 380 |
| 6.134OsclExecSchedulerBase Class Reference . . . . .                  | 382 |
| 6.135OsclExecSchedulerCommonBase Class Reference . . . . .            | 383 |
| 6.136OsclFileCache Class Reference . . . . .                          | 392 |
| 6.137OsclFileCacheBuffer Class Reference . . . . .                    | 394 |
| 6.138OsclFileHandle Class Reference . . . . .                         | 396 |
| 6.139OsclFileStats Class Reference . . . . .                          | 397 |
| 6.140OsclFileStatsItem Class Reference . . . . .                      | 398 |
| 6.141OsclGetHostByNameMethod Class Reference . . . . .                | 399 |
| 6.142OsclGetHostByNameRequest Class Reference . . . . .               | 400 |
| 6.143OsclInit Class Reference . . . . .                               | 401 |
| 6.144OsclInteger64Transport Struct Reference . . . . .                | 402 |
| 6.145OsclIPSocketI Class Reference . . . . .                          | 403 |
| 6.146OsclJump Class Reference . . . . .                               | 406 |
| 6.147OsclListenMethod Class Reference . . . . .                       | 407 |
| 6.148OsclListenRequest Class Reference . . . . .                      | 408 |
| 6.149OsclLockBase Class Reference . . . . .                           | 409 |
| 6.150OsclMem Class Reference . . . . .                                | 410 |
| 6.151OsclMemAllocator Class Reference . . . . .                       | 411 |

|   |     |
|---|-----|
| 6.152OsclMemAllocDestructDealloc< T > Class Template Reference . . . . .                      | 412 |
| 6.153OsclMemAudit Class Reference . . . . .   | 414 |
| 6.154OSCLMemAutoPtr< T, _Allocator > Class Template Reference . . . . .                       | 420 |
| 6.155OsclMemBasicAllocator Class Reference . . . . .  | 424 |
| 6.156OsclMemBasicAllocDestructDealloc< T > Class Template Reference . . . . .                 | 425 |
| 6.157OsclMemGlobalAuditObject Class Reference . . . . .                                       | 426 |
| 6.158OsclMemoryFragment Struct Reference . . . . .  | 427 |
| 6.159OsclMemPoolAllocator Class Reference . . . . .   | 428 |
| 6.160OsclMemPoolFixedChunkAllocator Class Reference . . . . .                                 | 429 |
| 6.161OsclMemPoolFixedChunkAllocatorObserver Class Reference . . . . .                         | 433 |
| 6.162OsclMemPoolResizableAllocator Class Reference . . . . .                                  | 434 |
| 6.163OsclMemPoolResizableAllocator::MemPoolBlockInfo Struct Reference . . . . .               | 440 |
| 6.164OsclMemPoolResizableAllocator::MemPoolBufferInfo Struct Reference . . . . .              | 441 |
| 6.165OsclMemPoolResizableAllocatorMemoryObserver Class Reference . . . . .                    | 442 |
| 6.166OsclMemPoolResizableAllocatorObserver Class Reference . . . . .                          | 443 |
| 6.167OsclMemStatsNode Class Reference . . . . .   | 444 |
| 6.168OsclMutex Class Reference . . . . .  | 445 |
| 6.169OsclNameString< __len > Class Template Reference . . . . .                               | 447 |
| 6.170OsclNativeFile Class Reference . . . . .   | 448 |
| 6.171OsclNativeFileParams Class Reference . . . . .   | 451 |
| 6.172OsclNetworkAddress Class Reference . . . . .   | 452 |
| 6.173OsclNullLock Class Reference . . . . .   | 453 |
| 6.174OsclPriorityLink Class Reference . . . . .   | 454 |
| 6.175OsclPriorityList< T > Class Template Reference . . . . .                                 | 455 |
| 6.176OsclPriorityQueue< Qelem, Alloc, Container, Compare > Class Template Reference . . . . . | 456 |
| 6.177OsclPriorityQueueBase Class Reference . . . . .  | 460 |
| 6.178OsclProcStatus Class Reference . . . . .   | 461 |
| 6.179OsclPtr Class Reference . . . . .  | 463 |
| 6.180OsclPtrC Class Reference . . . . .   | 465 |
| 6.181OsclRand Class Reference . . . . .   | 467 |
| 6.182OsclReadyAlloc Class Reference . . . . .   | 468 |
| 6.183OsclReadyCompare Class Reference . . . . .   | 469 |
| 6.184OsclReadyQ Class Reference . . . . .   | 470 |
| 6.185OsclRecvFromMethod Class Reference . . . . .   | 472 |
| 6.186OsclRecvFromRequest Class Reference . . . . .  | 474 |
| 6.187OsclRecvMethod Class Reference . . . . .   | 476 |

|   |     |
|---|-----|
| 6.188OsclRecvRequest Class Reference . . . . .                                      | 477 |
| 6.189OsclRefCounter Class Reference . . . . .                                       | 478 |
| 6.190OsclRefCounterDA Class Reference . . . . .                                     | 480 |
| 6.191OsclRefCounterMemFrag Class Reference . . . . .                                | 482 |
| 6.192OsclRefCounterMTDA< LockType > Class Template Reference . . . . .              | 484 |
| 6.193OsclRefCounterMTSA< DeallocType, LockType > Class Template Reference . . . . . | 486 |
| 6.194OsclRefCounterSA< DeallocType > Class Template Reference . . . . .             | 488 |
| 6.195OsclRegistryAccessClient Class Reference . . . . .                             | 490 |
| 6.196OsclRegistryAccessClientImpl Class Reference . . . . .                         | 492 |
| 6.197OsclRegistryAccessClientTlsImpl Class Reference . . . . .                      | 493 |
| 6.198OsclRegistryAccessElement Class Reference . . . . .                            | 494 |
| 6.199OsclRegistryClient Class Reference . . . . .                                   | 495 |
| 6.200OsclRegistryClientImpl Class Reference . . . . .                               | 497 |
| 6.201OsclRegistryClientTlsImpl Class Reference . . . . .                            | 499 |
| 6.202OsclRegistryServTlsImpl Class Reference . . . . .                              | 500 |
| 6.203OsclScheduler Class Reference . . . . .  | 502 |
| 6.204OsclSchedulerObserver Class Reference . . . . .                                | 503 |
| 6.205OsclScopedLock< LockClass > Class Template Reference . . . . .                 | 504 |
| 6.206OsclSelect Class Reference . . . . .   | 505 |
| 6.207OsclSemaphore Class Reference . . . . .  | 507 |
| 6.208OsclSendMethod Class Reference . . . . .                                       | 509 |
| 6.209OsclSendRequest Class Reference . . . . .                                      | 510 |
| 6.210OsclSendToMethod Class Reference . . . . .                                     | 511 |
| 6.211OsclSendToRequest Class Reference . . . . .                                    | 512 |
| 6.212OsclSharedPtr< TheClass > Class Template Reference . . . . .                   | 513 |
| 6.213OsclShutdownMethod Class Reference . . . . .                                   | 516 |
| 6.214OsclShutdownRequest Class Reference . . . . .                                  | 517 |
| 6.215OsclSingleton< T, ID, Registry > Class Template Reference . . . . .            | 518 |
| 6.216OsclSingletonRegistry Class Reference . . . . .                                | 520 |
| 6.217OsclSocketI Class Reference . . . . .  | 521 |
| 6.218OsclSocketIBase Class Reference . . . . .                                      | 526 |
| 6.219OsclSocketMethod Class Reference . . . . .                                     | 531 |
| 6.220OsclSocketObserver Class Reference . . . . .                                   | 534 |
| 6.221OsclSocketRequest Class Reference . . . . .                                    | 535 |
| 6.222OsclSocketRequestAO Class Reference . . . . .                                  | 536 |
| 6.223OsclSocketServ Class Reference . . . . .                                       | 540 |

|   |     |
|---|-----|
| 6.224OsclSocketServI Class Reference . . . . .                      | 542 |
| 6.225OsclSocketServIBase Class Reference . . . . .                  | 544 |
| 6.226OsclSocketServRequestList Class Reference . . . . .            | 546 |
| 6.227OsclSocketServRequestQElem Class Reference . . . . .           | 548 |
| 6.228OsclTCPSocket Class Reference . . . . .                        | 549 |
| 6.229OsclTCPSocketI Class Reference . . . . .                       | 555 |
| 6.230OsclThread Class Reference . . . . .                           | 558 |
| 6.231OsclThreadLock Class Reference . . . . .                       | 562 |
| 6.232OsclTickCount Class Reference . . . . .                        | 563 |
| 6.233OsclTimer< Alloc > Class Template Reference . . . . .          | 565 |
| 6.234OsclTimerCompare Class Reference . . . . .                     | 568 |
| 6.235OsclTimerObject Class Reference . . . . .                      | 569 |
| 6.236OsclTimerObserver Class Reference . . . . .                    | 573 |
| 6.237OsclTimerQ Class Reference . . . . .                           | 574 |
| 6.238OsclTLS< T, ID, Registry > Class Template Reference . . . . .  | 575 |
| 6.239OsclTLSE< T, ID, Registry > Class Template Reference . . . . . | 577 |
| 6.240OsclTLSRegistry Class Reference . . . . .                      | 579 |
| 6.241OsclTLSRegistryEx Class Reference . . . . .                    | 580 |
| 6.242OsclTrapItem Class Reference . . . . .                         | 581 |
| 6.243OsclTrapStack Class Reference . . . . .                        | 582 |
| 6.244OsclTrapStackItem Class Reference . . . . .                    | 583 |
| 6.245OsclUDPSocket Class Reference . . . . .                        | 584 |
| 6.246OsclUDPSocketI Class Reference . . . . .                       | 589 |
| 6.247OsclUuid Struct Reference . . . . .                            | 591 |
| 6.248PVActiveBase Class Reference . . . . .                         | 593 |
| 6.249PVActiveStats Class Reference . . . . .                        | 597 |
| 6.250PVLogger Class Reference . . . . .                             | 598 |
| 6.251PVLoggerAppender Class Reference . . . . .                     | 604 |
| 6.252PVLoggerFilter Class Reference . . . . .                       | 605 |
| 6.253PVLoggerLayout Class Reference . . . . .                       | 607 |
| 6.254PVLoggerRegistry Class Reference . . . . .                     | 609 |
| 6.255PVSchedulerStopper Class Reference . . . . .                   | 612 |
| 6.256PVSockBufRecv Class Reference . . . . .                        | 613 |
| 6.257PVSockBufSend Class Reference . . . . .                        | 614 |
| 6.258PVThreadContext Class Reference . . . . .                      | 615 |
| 6.259RecvFromParam Class Reference . . . . .                        | 617 |

|  |            |
|--|------------|
| 6.260RecvParam Class Reference . . . . .             | 619        |
| 6.261SendParam Class Reference . . . . .             | 620        |
| 6.262SendToParam Class Reference . . . . .           | 621        |
| 6.263ShutdownParam Class Reference . . . . .         | 622        |
| 6.264SocketRequestParam Class Reference . . . . .    | 623        |
| 6.265StrCSumPtrLen Struct Reference . . . . .        | 625        |
| 6.266StrPtrLen Struct Reference . . . . .            | 628        |
| 6.267TimeValue Class Reference . . . . .             | 630        |
| 6.268TLSStorageOps Class Reference . . . . .         | 636        |
| 6.269TReadyQueLink Class Reference . . . . .         | 637        |
| 6.270WStrPtrLen Struct Reference . . . . .           | 638        |
| <b>7 oscl File Documentation . . . . .</b>           | <b>640</b> |
| 7.1 oscl_aostatus.h File Reference . . . . .         | 640        |
| 7.2 oscl_assert.h File Reference . . . . .           | 641        |
| 7.3 oscl_base.h File Reference . . . . .             | 642        |
| 7.4 oscl_base_alloc.h File Reference . . . . .       | 643        |
| 7.5 oscl_base_macros.h File Reference . . . . .      | 644        |
| 7.6 oscl_bin_stream.h File Reference . . . . .       | 645        |
| 7.7 oscl_byte_order.h File Reference . . . . .       | 646        |
| 7.8 oscl_defalloc.h File Reference . . . . .         | 647        |
| 7.9 oscl_dll.h File Reference . . . . .              | 648        |
| 7.10 oscl_dns.h File Reference . . . . .             | 649        |
| 7.11 oscl_dns_gethostname.h File Reference . . . . . | 650        |
| 7.12 oscl_dns_imp.h File Reference . . . . .         | 651        |
| 7.13 oscl_dns_imp_base.h File Reference . . . . .    | 652        |
| 7.14 oscl_dns_imp_pv.h File Reference . . . . .      | 653        |
| 7.15 oscl_dns_method.h File Reference . . . . .      | 654        |
| 7.16 oscl_dns_param.h File Reference . . . . .       | 655        |
| 7.17 oscl_dns_request.h File Reference . . . . .     | 656        |
| 7.18 oscl_dns_tuneables.h File Reference . . . . .   | 657        |
| 7.19 oscl_double_list.h File Reference . . . . .     | 658        |
| 7.20 oscl_errno.h File Reference . . . . .           | 659        |
| 7.21 oscl_error.h File Reference . . . . .           | 660        |
| 7.22 oscl_error_allocator.h File Reference . . . . . | 661        |
| 7.23 oscl_error_codes.h File Reference . . . . .     | 662        |
| 7.24 oscl_error_imp.h File Reference . . . . .       | 663        |

|  |     |
|--|-----|
| 7.25 oscl_error_imp_cppexceptions.h File Reference . . . . . | 664 |
| 7.26 oscl_error_imp_fatalerror.h File Reference . . . . .    | 665 |
| 7.27 oscl_error_imp_jumps.h File Reference . . . . .         | 666 |
| 7.28 oscl_error_trapcleanup.h File Reference . . . . .       | 668 |
| 7.29 oscl_exception.h File Reference . . . . .               | 669 |
| 7.30 oscl_exclusive_ptr.h File Reference . . . . .           | 670 |
| 7.31 oscl_file_async_read.h File Reference . . . . .         | 671 |
| 7.32 oscl_file_cache.h File Reference . . . . .              | 672 |
| 7.33 oscl_file_dir_utils.h File Reference . . . . .          | 673 |
| 7.34 oscl_file_find.h File Reference . . . . .               | 675 |
| 7.35 oscl_file_handle.h File Reference . . . . .             | 676 |
| 7.36 oscl_file_io.h File Reference . . . . .                 | 677 |
| 7.37 oscl_file_native.h File Reference . . . . .             | 678 |
| 7.38 oscl_file_server.h File Reference . . . . .             | 679 |
| 7.39 oscl_file_stats.h File Reference . . . . .              | 680 |
| 7.40 oscl_file_types.h File Reference . . . . .              | 681 |
| 7.41 oscl_heapbase.h File Reference . . . . .                | 682 |
| 7.42 oscl_init.h File Reference . . . . .                    | 683 |
| 7.43 oscl_int64_utils.h File Reference . . . . .             | 684 |
| 7.44 oscl_ip_socket.h File Reference . . . . .               | 685 |
| 7.45 oscl_linked_list.h File Reference . . . . .             | 686 |
| 7.46 oscl_lock_base.h File Reference . . . . .               | 687 |
| 7.47 oscl_map.h File Reference . . . . .                     | 688 |
| 7.48 oscl_math.h File Reference . . . . .                    | 689 |
| 7.49 oscl_media_data.h File Reference . . . . .              | 690 |
| 7.50 oscl_media_status.h File Reference . . . . .            | 691 |
| 7.51 oscl_mem.h File Reference . . . . .                     | 692 |
| 7.52 oscl_mem_align.h File Reference . . . . .               | 695 |
| 7.53 oscl_mem_audit.h File Reference . . . . .               | 696 |
| 7.54 oscl_mem_audit_internals.h File Reference . . . . .     | 698 |
| 7.55 oscl_mem_auto_ptr.h File Reference . . . . .            | 699 |
| 7.56 oscl_mem_basic_functions.h File Reference . . . . .     | 700 |
| 7.57 oscl_mem_inst.h File Reference . . . . .                | 701 |
| 7.58 oscl_mem_mempool.h File Reference . . . . .             | 702 |
| 7.59 oscl_mempool_allocator.h File Reference . . . . .       | 703 |
| 7.60 oscl_mutex.h File Reference . . . . .                   | 704 |

|  |     |
|--|-----|
| 7.61 <a href="#">oscl_namestring.h File Reference</a>                | 705 |
| 7.62 <a href="#">oscl_opaque_type.h File Reference</a>               | 706 |
| 7.63 <a href="#">oscl_priqueue.h File Reference</a>                  | 707 |
| 7.64 <a href="#">oscl_proctstatus.h File Reference</a>               | 708 |
| 7.65 <a href="#">oscl_queue.h File Reference</a>                     | 709 |
| 7.66 <a href="#">oscl_rand.h File Reference</a>                      | 710 |
| 7.67 <a href="#">oscl_refcounter.h File Reference</a>                | 711 |
| 7.68 <a href="#">oscl_refcounter_memfrag.h File Reference</a>        | 712 |
| 7.69 <a href="#">oscl_registry_access_client.h File Reference</a>    | 713 |
| 7.70 <a href="#">oscl_registry_client.h File Reference</a>           | 714 |
| 7.71 <a href="#">oscl_registry_client_impl.h File Reference</a>      | 715 |
| 7.72 <a href="#">oscl_registry_serv_impl.h File Reference</a>        | 716 |
| 7.73 <a href="#">oscl_registry_serv_impl_global.h File Reference</a> | 717 |
| 7.74 <a href="#">oscl_registry_serv_impl_tls.h File Reference</a>    | 718 |
| 7.75 <a href="#">oscl_registry_types.h File Reference</a>            | 719 |
| 7.76 <a href="#">oscl_scheduler.h File Reference</a>                 | 720 |
| 7.77 <a href="#">oscl_scheduler_ao.h File Reference</a>              | 721 |
| 7.78 <a href="#">oscl_scheduler_aobase.h File Reference</a>          | 722 |
| 7.79 <a href="#">oscl_scheduler_readyq.h File Reference</a>          | 723 |
| 7.80 <a href="#">oscl_scheduler_threadcontext.h File Reference</a>   | 724 |
| 7.81 <a href="#">oscl_scheduler_tuneables.h File Reference</a>       | 725 |
| 7.82 <a href="#">oscl_scheduler_types.h File Reference</a>           | 726 |
| 7.83 <a href="#">oscl_semaphore.h File Reference</a>                 | 727 |
| 7.84 <a href="#">oscl_shared_ptr.h File Reference</a>                | 728 |
| 7.85 <a href="#">oscl_singleton.h File Reference</a>                 | 729 |
| 7.86 <a href="#">oscl_snprintf.h File Reference</a>                  | 731 |
| 7.87 <a href="#">oscl_socket.h File Reference</a>                    | 732 |
| 7.88 <a href="#">oscl_socket_accept.h File Reference</a>             | 733 |
| 7.89 <a href="#">oscl_socket_bind.h File Reference</a>               | 734 |
| 7.90 <a href="#">oscl_socket_connect.h File Reference</a>            | 735 |
| 7.91 <a href="#">oscl_socket_imp.h File Reference</a>                | 736 |
| 7.92 <a href="#">oscl_socket_imp_base.h File Reference</a>           | 737 |
| 7.93 <a href="#">oscl_socket_imp_pv.h File Reference</a>             | 738 |
| 7.94 <a href="#">oscl_socket_listen.h File Reference</a>             | 739 |
| 7.95 <a href="#">oscl_socket_method.h File Reference</a>             | 740 |
| 7.96 <a href="#">oscl_socket_recv.h File Reference</a>               | 741 |

|  |     |
|--|-----|
| 7.97 oscl_socket_recv_from.h File Reference . . . . .        | 742 |
| 7.98 oscl_socket_request.h File Reference . . . . .          | 743 |
| 7.99 oscl_socket_send.h File Reference . . . . .             | 744 |
| 7.100oscl_socket_send_to.h File Reference . . . . .          | 745 |
| 7.101oscl_socket_serv_imp.h File Reference . . . . .         | 746 |
| 7.102oscl_socket_serv_imp_base.h File Reference . . . . .    | 747 |
| 7.103oscl_socket_serv_imp_pv.h File Reference . . . . .      | 748 |
| 7.104oscl_socket_serv_imp_reqlist.h File Reference . . . . . | 749 |
| 7.105oscl_socket_shutdown.h File Reference . . . . .         | 750 |
| 7.106oscl_socket_stats.h File Reference . . . . .            | 751 |
| 7.107oscl_socket_tuneables.h File Reference . . . . .        | 753 |
| 7.108oscl_socket_types.h File Reference . . . . .            | 755 |
| 7.109oscl_stdstring.h File Reference . . . . .               | 757 |
| 7.110oscl_str_ptr_len.h File Reference . . . . .             | 759 |
| 7.111oscl_string.h File Reference . . . . .                  | 760 |
| 7.112oscl_string_containers.h File Reference . . . . .       | 761 |
| 7.113oscl_string_rep.h File Reference . . . . .              | 762 |
| 7.114oscl_string_uri.h File Reference . . . . .              | 763 |
| 7.115oscl_string_utf8.h File Reference . . . . .             | 764 |
| 7.116oscl_string_utils.h File Reference . . . . .            | 765 |
| 7.117oscl_string_xml.h File Reference . . . . .              | 766 |
| 7.118oscl_tagtree.h File Reference . . . . .                 | 767 |
| 7.119oscl_tcp_socket.h File Reference . . . . .              | 768 |
| 7.120oscl_thread.h File Reference . . . . .                  | 769 |
| 7.121oscl_tickcount.h File Reference . . . . .               | 771 |
| 7.122oscl_time.h File Reference . . . . .                    | 772 |
| 7.123oscl_timer.h File Reference . . . . .                   | 774 |
| 7.124oscl_tls.h File Reference . . . . .                     | 775 |
| 7.125oscl_tree.h File Reference . . . . .                    | 776 |
| 7.126oscl_types.h File Reference . . . . .                   | 777 |
| 7.127oscl_udp_socket.h File Reference . . . . .              | 778 |
| 7.128oscl_utf8conv.h File Reference . . . . .                | 779 |
| 7.129oscl_uuid.h File Reference . . . . .                    | 780 |
| 7.130oscl_vector.h File Reference . . . . .                  | 781 |
| 7.131osclconfig.h File Reference . . . . .                   | 782 |
| 7.132osclconfig_ansi_memory.h File Reference . . . . .       | 784 |

|   |     |
|---|-----|
| 7.133osclconfig_check.h File Reference . . . . .                | 785 |
| 7.134osclconfig_compiler_warnings.h File Reference . . . . .    | 786 |
| 7.135osclconfig_error.h File Reference . . . . .                | 787 |
| 7.136osclconfig_error_check.h File Reference . . . . .          | 788 |
| 7.137osclconfig_global_new_delete.h File Reference . . . . .    | 789 |
| 7.138osclconfig_global_placement_new.h File Reference . . . . . | 790 |
| 7.139osclconfig_io.h File Reference . . . . .                   | 791 |
| 7.140osclconfig_io_check.h File Reference . . . . .             | 798 |
| 7.141osclconfig_ix86.h File Reference . . . . .                 | 799 |
| 7.142osclconfig_lib.h File Reference . . . . .                  | 800 |
| 7.143osclconfig_lib_check.h File Reference . . . . .            | 801 |
| 7.144osclconfig_limits_typedefs.h File Reference . . . . .      | 802 |
| 7.145osclconfig_memory.h File Reference . . . . .               | 803 |
| 7.146osclconfig_memory_check.h File Reference . . . . .         | 804 |
| 7.147osclconfig_no_os.h File Reference . . . . .                | 805 |
| 7.148osclconfig_proc.h File Reference . . . . .                 | 806 |
| 7.149osclconfig_proc_check.h File Reference . . . . .           | 807 |
| 7.150osclconfig_proc_unix_android.h File Reference . . . . .    | 809 |
| 7.151osclconfig_proc_unix_common.h File Reference . . . . .     | 811 |
| 7.152osclconfig_time.h File Reference . . . . .                 | 813 |
| 7.153osclconfig_time_check.h File Reference . . . . .           | 814 |
| 7.154osclconfig_unix_android.h File Reference . . . . .         | 815 |
| 7.155osclconfig_unix_common.h File Reference . . . . .          | 819 |
| 7.156osclconfig_util.h File Reference . . . . .                 | 823 |
| 7.157osclconfig_util_check.h File Reference . . . . .           | 824 |
| 7.158pvlogger.h File Reference . . . . .                        | 825 |
| 7.159pvlogger_accessories.h File Reference . . . . .            | 833 |
| 7.160pvlogger_c.h File Reference . . . . .                      | 834 |
| 7.161pvlogger_registry.h File Reference . . . . .               | 836 |

# Chapter 1

## oscl Module Index

### 1.1 oscl Modules

Here is a list of all modules:

|                       |     |
|-----------------------|-----|
| OSCL config . . . . . | 19  |
| OSCL Base . . . . .   | 23  |
| OSCL Memory . . . . . | 44  |
| OSCL Util . . . . .   | 60  |
| OSCL Error . . . . .  | 81  |
| OSCL IO . . . . .     | 91  |
| OSCL Proc . . . . .   | 99  |
| OSCL Init . . . . .   | 103 |

# Chapter 2

## oscl Hierarchical Index

### 2.1 oscl Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

|  |     |
|--|-----|
| _OscIHeapBase . . . . .                  | 106 |
| HeapBase . . . . .                       | 132 |
| Oscl_File . . . . .                      | 174 |
| OSCL_String . . . . .                    | 251 |
| OSCL_FastString . . . . .                | 170 |
| OSCL_HeapString< Alloc > . . . . .       | 191 |
| OSCL_HeapStringA . . . . .               | 193 |
| OSCL_StackString< MaxBufSize > . . . . . | 248 |
| OsclActiveObject . . . . .               | 301 |
| OsclAsyncFile . . . . .                  | 307 |
| OsclDNSRequestAO . . . . .               | 354 |
| OsclGetHostNameRequest . . . . .         | 400 |
| OsclSocketRequestAO . . . . .            | 536 |
| OsclAcceptRequest . . . . .              | 300 |
| OsclBindRequest . . . . .                | 314 |
| OsclConnectRequest . . . . .             | 340 |
| OsclListenRequest . . . . .              | 408 |
| OsclRecvFromRequest . . . . .            | 474 |
| OsclRecvRequest . . . . .                | 477 |
| OsclSendRequest . . . . .                | 510 |
| OsclSendToRequest . . . . .              | 512 |
| OsclShutdownRequest . . . . .            | 517 |
| PVSchedulerStopper . . . . .             | 612 |
| OsclAsyncFileBuffer . . . . .            | 310 |
| OsclBuf . . . . .                        | 331 |
| OsclDNS . . . . .                        | 342 |
| OsclFileCache . . . . .                  | 392 |
| OsclNativeFile . . . . .                 | 448 |
| OsclPtr . . . . .                        | 463 |
| OsclPtrC . . . . .                       | 465 |
| OsclRegistryClient . . . . .             | 495 |
| OsclSocketServ . . . . .                 | 540 |
| OsclTCPSocket . . . . .                  | 549 |

|   |     |
|---|-----|
| OsclTimerObject . . . . .                                   | 569 |
| CallbackTimer< Alloc > . . . . .                            | 120 |
| OsclDNSMethod . . . . .                                     | 349 |
| OsclGetHostByNameMethod . . . . .                           | 399 |
| OsclSocketMethod . . . . .                                  | 531 |
| OsclAcceptMethod . . . . .                                  | 299 |
| OsclBindMethod . . . . .                                    | 313 |
| OsclConnectMethod . . . . .                                 | 339 |
| OsclListenMethod . . . . .                                  | 407 |
| OsclRecvFromMethod . . . . .                                | 472 |
| OsclRecvMethod . . . . .                                    | 476 |
| OsclSendMethod . . . . .                                    | 509 |
| OsclSendToMethod . . . . .                                  | 511 |
| OsclShutdownMethod . . . . .                                | 516 |
| OsclSocketServI . . . . .                                   | 542 |
| OsclUDPSocket . . . . .                                     | 584 |
| OsclExecSchedulerBase . . . . .                             | 382 |
| OsclExecScheduler . . . . .                                 | 380 |
| allocator . . . . .   | 109 |
| BufferMgr . . . . .   | 114 |
| BufferState . . . . .                                       | 115 |
| BuFragGroup< ChainClass, max_frags > . . . . .              | 116 |
| MediaData< ChainClass, max_frags, local_bufsize > . . . . . | 137 |
| BufFragStatusClass . . . . .                                | 119 |
| MediaStatusClass . . . . .                                  | 140 |
| CallbackTimerObserver . . . . .                             | 122 |
| OsclTimer< Alloc > . . . . .                                | 565 |
| CFastRep . . . . .  | 123 |
| CHheapRep . . . . .   | 125 |
| CStackRep . . . . .   | 128 |
| DNSRequestParam . . . . .                                   | 129 |
| GetHostByNameParam . . . . .                                | 131 |
| internalLeave . . . . .                                     | 134 |
| LinkedListElement< LLClass > . . . . .                      | 135 |
| MemAllocator< T > . . . . .                                 | 141 |
| MM_AllocBlockFence . . . . .                                | 142 |
| MM_AllocBlockHdr . . . . .                                  | 143 |
| MM_AllocInfo . . . . .                                      | 144 |
| MM_AllocNode . . . . .                                      | 146 |
| MM_AllocQueryInfo . . . . .                                 | 147 |
| MM_Audit_Imp . . . . .                                      | 148 |
| MM_AuditOverheadStats . . . . .                             | 156 |
| MM_FailInsertParam . . . . .                                | 157 |
| MM_Stats_CB . . . . .                                       | 158 |
| MM_Stats_t . . . . .  | 159 |
| NTPTime . . . . .   | 161 |
| Oscl_Alloc . . . . .  | 165 |
| Oscl_DefAlloc . . . . .                                     | 167 |
| _OsclBasicAllocator . . . . .                               | 104 |
| OsclAllocDestructDealloc . . . . .                          | 305 |
| OsclMemAllocDestructDealloc< T > . . . . .                  | 412 |
| OsclMemBasicAllocDestructDealloc< T > . . . . .             | 425 |

|   |     |
|---|-----|
| OsclMemAllocator . . . . .  | 411 |
| OsclMemBasicAllocator . . . . .   | 424 |
| OsclMemPoolFixedChunkAllocator . . . . .  | 429 |
| OsclMemPoolResizableAllocator . . . . .   | 434 |
| OsclReadyAlloc . . . . .  | 468 |
| Oscl_Dealloc . . . . .  | 166 |
| Oscl_DefAlloc . . . . .   | 167 |
| Oscl_File::OsclCacheObserver . . . . .  | 182 |
| Oscl_File::OsclFixedCacheParam . . . . .  | 183 |
| Oscl_FileFind . . . . .   | 184 |
| Oscl_FileServer . . . . .   | 188 |
| oscl_fsstat . . . . .   | 190 |
| Oscl_Int64_Utils . . . . .  | 197 |
| Oscl_Less< T > . . . . .  | 199 |
| Oscl_Linked_List_Base . . . . .   | 204 |
| Oscl_Linked_List< LLClass, Alloc > . . . . .  | 200 |
| Oscl_Map< Key, T, Alloc, Compare > . . . . .  | 208 |
| Oscl_Map< Key, T, Alloc, Compare >::value_compare . . . . .   | 215 |
| Oscl_MTLinked_List< LLClass, Alloc, TheLock > . . . . .   | 217 |
| Oscl_Opaque_Type_Alloc . . . . .  | 221 |
| Oscl_Queue< T, Alloc > . . . . .  | 227 |
| Oscl_Vector< T, Alloc > . . . . .   | 276 |
| Oscl_Vector< TOsclReady, OsclReadyAlloc > . . . . .   | 276 |
| Oscl_Opaque_Type_Alloc_LL . . . . .   | 222 |
| Oscl_Linked_List< LLClass, Alloc > . . . . .  | 200 |
| Oscl_Opaque_Type_Compare . . . . .  | 224 |
| OsclPriorityQueue< Qelem, Alloc, Container, Compare > . . . . .   | 456 |
| OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReady-Alloc >, OsclReadyCompare > . . . . . | 456 |
| OsclReadyQ . . . . .  | 470 |
| OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReady-Alloc >, OsclTimerCompare > . . . . . | 456 |
| OsclTimerQ . . . . .  | 574 |
| Oscl_Pair< T1, T2 > . . . . .   | 226 |
| Oscl_Queue_Base . . . . .   | 230 |
| Oscl_Queue< T, Alloc > . . . . .  | 227 |
| Oscl_Rb_Tree_Base . . . . .   | 237 |
| Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > . . . . .  | 233 |
| Oscl_Rb_Tree_Const_Iterator< Value > . . . . .  | 238 |
| Oscl_Rb_Tree_Iterator< Value > . . . . .  | 241 |
| Oscl_Rb_Tree_Node_Base . . . . .  | 245 |
| Oscl_Rb_Tree_Node< Value > . . . . .  | 244 |
| Oscl_Select1st< V, U > . . . . .  | 247 |
| oscl_stat_buf . . . . .   | 250 |
| Oscl_Tag_Base . . . . .   | 258 |
| Oscl_Tag< Alloc > . . . . .   | 256 |
| Oscl_TagTree< T, Alloc > . . . . .  | 260 |
| Oscl_TagTree< T, Alloc >::const_iterator . . . . .  | 264 |
| Oscl_TagTree< T, Alloc >::iterator . . . . .  | 267 |
| Oscl_TagTree< T, Alloc >::Node . . . . .  | 270 |
| Oscl_TAlloc< T, Alloc >::rebind< U, V > . . . . .   | 275 |

|   |     |
|---|-----|
| Oscl_Vector_Base . . . . .                          | 281 |
| Oscl_Vector< T, Alloc > . . . . .                   | 276 |
| Oscl_Vector< TOsclReady, OsclReadyAlloc > . . . . . | 276 |
| OSCL_wString . . . . .                              | 295 |
| OSCL_wFastString . . . . .                          | 285 |
| OSCL_wHeapString< Alloc > . . . . .                 | 288 |
| OSCL_wHeapStringA . . . . .                         | 290 |
| OSCL_wStackString< MaxBufSize > . . . . .           | 293 |
| OsclAOStatus . . . . .                              | 306 |
| OsclAuditCB . . . . .                               | 312 |
| OsclBinStream . . . . .                             | 327 |
| OsclBinIStream . . . . .                            | 315 |
| OsclBinIStreamBigEndian . . . . .                   | 317 |
| OsclBinIStreamLittleEndian . . . . .                | 320 |
| OsclBinOStream . . . . .                            | 322 |
| OsclBinOStreamBigEndian . . . . .                   | 323 |
| OsclBinOStreamLittleEndian . . . . .                | 325 |
| OsclCompareLess< T > . . . . .                      | 333 |
| OsclComponentRegistry . . . . .                     | 334 |
| OsclComponentRegistryData . . . . .                 | 336 |
| OsclComponentRegistryElement . . . . .              | 337 |
| OsclDestructDealloc . . . . .                       | 341 |
| Oscl_TAlloc< T, Alloc > . . . . .                   | 272 |
| OsclAllocDestructDealloc . . . . .                  | 305 |
| OsclDNSIBase . . . . .                              | 346 |
| OsclDNSI . . . . .                                  | 344 |
| OsclDNSObserver . . . . .                           | 352 |
| OsclDNSRequest . . . . .                            | 353 |
| OsclDoubleLink . . . . .                            | 357 |
| OsclPriorityLink . . . . .                          | 454 |
| OsclDoubleListBase . . . . .                        | 359 |
| OsclDoubleList< T > . . . . .                       | 358 |
| OsclPriorityList< T > . . . . .                     | 455 |
| OsclDoubleRunner< T > . . . . .                     | 361 |
| OsclError . . . . .                                 | 363 |
| OsclErrorAllocator . . . . .                        | 365 |
| OsclErrorTrap . . . . .                             | 367 |
| OsclErrorTrapImp . . . . .                          | 368 |
| OsclException< LeaveCode > . . . . .                | 370 |
| OsclExclusiveArrayPtr< T > . . . . .                | 371 |
| OsclExclusivePtr< T > . . . . .                     | 374 |
| OsclExclusivePtrA< T, Alloc > . . . . .             | 377 |
| OsclExecSchedulerCommonBase . . . . .               | 383 |
| OsclExecScheduler . . . . .                         | 380 |
| OsclFileCacheBuffer . . . . .                       | 394 |
| OsclFileHandle . . . . .                            | 396 |
| OsclFileStats . . . . .                             | 397 |
| OsclFileStatsItem . . . . .                         | 398 |
| OsclInit . . . . .                                  | 401 |
| OsclInteger64Transport . . . . .                    | 402 |
| OsclIPSocketI . . . . .                             | 403 |

|   |     |
|---|-----|
| OsclTCPSocketI . . . . .  | 555 |
| OsclUDPSocketI . . . . .  | 589 |
| OsclJump . . . . .  | 406 |
| OsclLockBase . . . . .  | 409 |
| OsclMutex . . . . .   | 445 |
| OsclNullLock . . . . .  | 453 |
| OsclThreadLock . . . . .  | 562 |
| OsclMem . . . . .   | 410 |
| OsclMemAudit . . . . .  | 414 |
| OSCLMemAutoPtr< T, _Allocator >   | 420 |
| OsclMemGlobalAuditObject . . . . .  | 426 |
| OsclMemoryFragment . . . . .  | 427 |
| BufferFragment . . . . .  | 113 |
| OsclMemPoolAllocator . . . . .  | 428 |
| OsclMemPoolFixedChunkAllocatorObserver . . . . .  | 433 |
| OsclMemPoolResizableAllocator::MemPoolBlockInfo . . . . .   | 440 |
| OsclMemPoolResizableAllocator::MemPoolBufferInfo . . . . .  | 441 |
| OsclMemPoolResizableAllocatorMemoryObserver . . . . .   | 442 |
| OsclMemPoolResizableAllocatorObserver . . . . .   | 443 |
| OsclMemStatsNode . . . . .  | 444 |
| OsclNameString< __len > . . . . .   | 447 |
| OsclNativeFileParams . . . . .  | 451 |
| OsclNetworkAddress . . . . .  | 452 |
| OsclPriorityQueueBase . . . . .   | 460 |
| OsclPriorityQueue< Qelem, Alloc, Container, Compare > . . . . .   | 456 |
| OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReady-Alloc >, OsclReadyCompare > . . . . . | 456 |
| OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReady-Alloc >, OsclTimerCompare > . . . . . | 456 |
| OsclProcStatus . . . . .  | 461 |
| OsclRand . . . . .  | 467 |
| OsclReadyCompare . . . . .  | 469 |
| OsclRefCounter . . . . .  | 478 |
| Oscl_DefAllocWithRefCounter< DefAlloc > . . . . .   | 168 |
| OsclRefCounterDA . . . . .  | 480 |
| OsclRefCounterMTDA< LockType > . . . . .  | 484 |
| OsclRefCounterMTSA< DeallocType, LockType > . . . . .   | 486 |
| OsclRefCounterSA< DeallocType > . . . . .   | 488 |
| OsclRefCounterMemFrag . . . . .   | 482 |
| OsclRegistryAccessClient . . . . .  | 490 |
| OsclRegistryAccessElement . . . . .   | 494 |
| OsclRegistryClientImpl . . . . .  | 497 |
| OsclRegistryAccessClientImpl . . . . .  | 492 |
| OsclRegistryServTlsImpl . . . . .   | 500 |
| OsclRegistryAccessClientTlsImpl . . . . .   | 493 |
| OsclRegistryClientTlsImpl . . . . .   | 499 |
| OsclScheduler . . . . .   | 502 |
| OsclSchedulerObserver . . . . .   | 503 |
| OsclScopedLock< LockClass > . . . . .   | 504 |
| OsclSelect . . . . .  | 505 |
| OsclSemaphore . . . . .   | 507 |
| OsclSharedPtr< TheClass > . . . . .   | 513 |

|  |     |
|--|-----|
| OsclSingleton< T, ID, Registry > . . . . . | 518 |
| OsclSingletonRegistry . . . . .            | 520 |
| OsclSocketIBase . . . . .                  | 526 |
| OsclSocketI . . . . .                      | 521 |
| OsclSocketObserver . . . . .               | 534 |
| OsclSocketRequest . . . . .                | 535 |
| OsclSocketServIBase . . . . .              | 544 |
| OsclSocketServI . . . . .                  | 542 |
| OsclSocketServRequestList . . . . .        | 546 |
| OsclSocketServRequestQElem . . . . .       | 548 |
| OsclThread . . . . .                       | 558 |
| OsclTickCount . . . . .                    | 563 |
| OsclTimerCompare . . . . .                 | 568 |
| OsclTimerObserver . . . . .                | 573 |
| OsclTLS< T, ID, Registry > . . . . .       | 575 |
| OsclTLSEx< T, ID, Registry > . . . . .     | 577 |
| OsclTLSRegistry . . . . .                  | 579 |
| OsclTLSRegistryEx . . . . .                | 580 |
| OsclTrapItem . . . . .                     | 581 |
| OsclTrapStack . . . . .                    | 582 |
| OsclTrapStackItem . . . . .                | 583 |
| OsclUuid . . . . .                         | 591 |
| PVActiveBase . . . . .                     | 593 |
| OsclActiveObject . . . . .                 | 301 |
| OsclTimerObject . . . . .                  | 569 |
| PVActiveStats . . . . .                    | 597 |
| PVLogger . . . . .                         | 598 |
| PVLoggerAppender . . . . .                 | 604 |
| PVLoggerFilter . . . . .                   | 605 |
| AllPassFilter . . . . .                    | 110 |
| PVLoggerLayout . . . . .                   | 607 |
| PVLoggerRegistry . . . . .                 | 609 |
| PVSockBufRecv . . . . .                    | 613 |
| PVSockBufSend . . . . .                    | 614 |
| PVThreadContext . . . . .                  | 615 |
| SocketRequestParam . . . . .               | 623 |
| AcceptParam . . . . .                      | 108 |
| BindParam . . . . .                        | 112 |
| ConnectParam . . . . .                     | 127 |
| ListenParam . . . . .                      | 136 |
| RecvFromParam . . . . .                    | 617 |
| RecvParam . . . . .                        | 619 |
| SendParam . . . . .                        | 620 |
| SendToParam . . . . .                      | 621 |
| ShutdownParam . . . . .                    | 622 |
| StrPtrLen . . . . .                        | 628 |
| StrCSumPtrLen . . . . .                    | 625 |
| TimeValue . . . . .                        | 630 |
| TLSStorageOps . . . . .                    | 636 |
| TReadyQueLink . . . . .                    | 637 |
| WStrPtrLen . . . . .                       | 638 |

# Chapter 3

## oscl Data Structure Index

### 3.1 oscl Data Structures

Here are the data structures with brief descriptions:

|   |     |
|---|-----|
| _OsclBasicAllocator . . . . .                     | 104 |
| _OsclHeapBase . . . . .                           | 106 |
| AcceptParam . . . . .                             | 108 |
| allocator . . . . .                               | 109 |
| AllPassFilter . . . . .                           | 110 |
| BindParam . . . . .                               | 112 |
| BufferFragment . . . . .                          | 113 |
| BufferMgr . . . . .                               | 114 |
| BufferState . . . . .                             | 115 |
| BufFragGroup< ChainClass, max_frags >             | 116 |
| BufFragStatusClass . . . . .                      | 119 |
| CallbackTimer< Alloc > . . . . .                  | 120 |
| CallbackTimerObserver . . . . .                   | 122 |
| CFastRep . . . . .                                | 123 |
| CHheapRep . . . . .                               | 125 |
| ConnectParam . . . . .                            | 127 |
| CStackRep . . . . .                               | 128 |
| DNSRequestParam . . . . .                         | 129 |
| GetHostNameParam . . . . .                        | 131 |
| HeapBase . . . . .                                | 132 |
| internalLeave . . . . .                           | 134 |
| LinkedListElement< LLClass > . . . . .            | 135 |
| ListenParam . . . . .                             | 136 |
| MediaData< ChainClass, max_frags, local_bufsize > | 137 |
| MediaStatusClass . . . . .                        | 140 |
| MemAllocator< T > . . . . .                       | 141 |
| MM_AllocBlockFence . . . . .                      | 142 |
| MM_AllocBlockHdr . . . . .                        | 143 |
| MM_AllocInfo . . . . .                            | 144 |
| MM_AllocNode . . . . .                            | 146 |
| MM_AllocQueryInfo . . . . .                       | 147 |
| MM_Audit_Imp . . . . .                            | 148 |
| MM_AuditOverheadStats . . . . .                   | 156 |

|   |     |
|---|-----|
| MM_FailInsertParam . . . . .  | 157 |
| MM_Stats_CB . . . . .   | 158 |
| MM_Stats_t . . . . .  | 159 |
| NTPTime (Time value as the number of seconds since 0h (UTC) Jan. 1, 1900) . . . . . | 161 |
| OscI_Alloc . . . . .  | 165 |
| OscI_Dealloc . . . . .  | 166 |
| OscI_DefAlloc . . . . .   | 167 |
| OscI_DefAllocWithRefCounter< DefAlloc > . . . . .                                   | 168 |
| OSCL_FastString . . . . .   | 170 |
| OscI_File . . . . .   | 174 |
| OscI_File::OscICacheObserver . . . . .  | 182 |
| OscI_File::OscIFixedCacheParam . . . . .  | 183 |
| OscI_FileFind . . . . .   | 184 |
| OscI_FileServer . . . . .   | 188 |
| oscl_fstat . . . . .  | 190 |
| OSCL_HeapString< Alloc > . . . . .  | 191 |
| OSCL_HeapStringA . . . . .  | 193 |
| OscI_Int64_Utils (Wrapper for commonly used int64/uint64 operations) . . . . .      | 197 |
| OscI_Less< T > . . . . .  | 199 |
| OscI_Linked_List< LLClass, Alloc > . . . . .  | 200 |
| OscI_Linked_List_Base . . . . .   | 204 |
| OscI_Map< Key, T, Alloc, Compare > . . . . .  | 208 |
| OscI_Map< Key, T, Alloc, Compare >::value_compare . . . . .                         | 215 |
| OscI_MTLinked_List< LLClass, Alloc, TheLock > . . . . .                             | 217 |
| OscI_Opaque_Type_Alloc . . . . .  | 221 |
| OscI_Opaque_Type_Alloc_LL . . . . .   | 222 |
| OscI_Opaque_Type_Compare . . . . .  | 224 |
| OscI_Pair< T1, T2 > . . . . .   | 226 |
| OscI_Queue< T, Alloc > . . . . .  | 227 |
| OscI_Queue_Base . . . . .   | 230 |
| OscI_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > . . . . .                    | 233 |
| OscI_Rb_Tree_Base . . . . .   | 237 |
| OscI_Rb_Tree_Const_Iterator< Value > . . . . .                                      | 238 |
| OscI_Rb_Tree_Iterator< Value > . . . . .  | 241 |
| OscI_Rb_Tree_Node< Value > . . . . .  | 244 |
| OscI_Rb_Tree_Node_Base . . . . .  | 245 |
| OscI_Select1st< V, U > . . . . .  | 247 |
| OSCL_StackString< MaxBufSize > . . . . .  | 248 |
| oscl_stat_buf . . . . .   | 250 |
| OSCL_String . . . . .   | 251 |
| OscI_Tag< Alloc > . . . . .   | 256 |
| OscI_Tag_Base . . . . .   | 258 |
| OscI_TagTree< T, Alloc > . . . . .  | 260 |
| OscI_TagTree< T, Alloc >::const_iterator . . . . .                                  | 264 |
| OscI_TagTree< T, Alloc >::iterator . . . . .  | 267 |
| OscI_TagTree< T, Alloc >::Node . . . . .  | 270 |
| OscI_TAlloc< T, Alloc > . . . . .   | 272 |
| OscI_TAlloc< T, Alloc >::rebind< U, V > . . . . .                                   | 275 |
| OscI_Vector< T, Alloc > . . . . .   | 276 |
| OscI_Vector_Base . . . . .  | 281 |
| OSCL_wFastString . . . . .  | 285 |
| OSCL_wHeapString< Alloc > . . . . .   | 288 |
| OSCL_wHeapStringA . . . . .   | 290 |
| OSCL_wStackString< MaxBufSize > . . . . .   | 293 |

|  |     |
|--|-----|
| <b>OSCL_wString</b>  | 295 |
| <b>OsclAcceptMethod</b>  | 299 |
| <b>OsclAcceptRequest</b>   | 300 |
| <b>OsclActiveObject</b>  | 301 |
| <b>OsclAllocDestructDealloc</b>  | 305 |
| <b>OsclAOStatus</b>  | 306 |
| <b>OsclAsyncFile</b>   | 307 |
| <b>OsclAsyncFileBuffer</b>   | 310 |
| <b>OsclAuditCB</b>   | 312 |
| <b>OsclBindMethod</b>  | 313 |
| <b>OsclBindRequest</b>   | 314 |
| <b>OsclBinIStream</b>  | 315 |
| <b>OsclBinIStreamBigEndian</b>   | 317 |
| <b>OsclBinIStreamLittleEndian</b>  | 320 |
| <b>OsclBinOStream</b> (Class OsclBinOStream implements the basic stream functions for an output stream)  | 322 |
| <b>OsclBinOStreamBigEndian</b> (Class OsclBinOStreamBigEndian implements a binary output stream using big endian byte ordering)  | 323 |
| <b>OsclBinOStreamLittleEndian</b> (Class OsclBinOStreamLittleEndian implements a binary output stream using little endian byte ordering)   | 325 |
| <b>OsclBinStream</b>   | 327 |
| <b>OsclBuf</b>   | 331 |
| <b>OsclCompareLess&lt; T &gt;</b>  | 333 |
| <b>OsclComponentRegistry</b>   | 334 |
| <b>OsclComponentRegistryData</b>   | 336 |
| <b>OsclComponentRegistryElement</b>  | 337 |
| <b>OsclConnectMethod</b>   | 339 |
| <b>OsclConnectRequest</b>  | 340 |
| <b>OsclDestructDealloc</b>   | 341 |
| <b>OsclDNS</b>   | 342 |
| <b>OsclDNSI</b>  | 344 |
| <b>OsclDNSIBase</b>  | 346 |
| <b>OsclDNSMethod</b>   | 349 |
| <b>OsclDNSObserver</b>   | 352 |
| <b>OsclDNSRequest</b>  | 353 |
| <b>OsclDNSRequestAO</b>  | 354 |
| <b>OsclDoubleLink</b>  | 357 |
| <b>OsclDoubleList&lt; T &gt;</b>   | 358 |
| <b>OsclDoubleListBase</b>  | 359 |
| <b>OsclDoubleRunner&lt; T &gt;</b>   | 361 |
| <b>OsclError</b>   | 363 |
| <b>OsclErrorAllocator</b> (This class provides static methods to invoke the user defined memory allocation routines)   | 365 |
| <b>OsclErrorTrap</b>   | 367 |
| <b>OsclErrorTrapImp</b>  | 368 |
| <b>OsclException&lt; LeaveCode &gt;</b> (Oscl_exception.h contains all the exception handling macros and classes This template class provides the base exception class that all exceptions derive from)  | 370 |
| <b>OsclExclusiveArrayPtr&lt; T &gt;</b> (Template class that defines an array pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the OsclExclusiveArrayPtr expires, its destructor uses delete to free the memory) | 371 |
| <b>OsclExclusivePtr&lt; T &gt;</b> (Template class that defines a pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the OsclExclusivePtr expires, its destructor uses delete to free the memory)                  | 374 |

|  |     |
|--|-----|
| OsclExclusivePtrA< T, Alloc > (Template class that defines any pointer like object intended to be assigned an address obtained (directly or or indirectly) through Alloc. When the OsclExclusivePtrA expires, Alloc is used to free the memory) . . . . .                                | 377 |
| OsclExecScheduler . . . . .  | 380 |
| OsclExecSchedulerBase . . . . .  | 382 |
| OsclExecSchedulerCommonBase . . . . .  | 383 |
| OsclFileCache . . . . .  | 392 |
| OsclFileCacheBuffer . . . . .  | 394 |
| OsclFileHandle . . . . .   | 396 |
| OsclFileStats . . . . .  | 397 |
| OsclFileStatsItem . . . . .  | 398 |
| OsclGetHostNameMethod . . . . .  | 399 |
| OsclGetHostNameRequest . . . . .   | 400 |
| OsclInit . . . . .   | 401 |
| OsclInteger64Transport . . . . .   | 402 |
| OsclIPSocketI . . . . .  | 403 |
| OsclJump . . . . .   | 406 |
| OsclListenMethod . . . . .   | 407 |
| OsclListenRequest . . . . .  | 408 |
| OsclLockBase . . . . .   | 409 |
| OsclMem . . . . .  | 410 |
| OsclMemAllocator . . . . .   | 411 |
| OsclMemAllocDestructDealloc< T > . . . . .   | 412 |
| OsclMemAudit . . . . .   | 414 |
| OSCLMemAutoPtr< T, _Allocator > (The oscl_auto_ptr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the oscl_auto_ptr expires, its destructor uses delete to free the memory) . . . . . | 420 |
| OsclMemBasicAllocator . . . . .  | 424 |
| OsclMemBasicAllocDestructDealloc< T > . . . . .  | 425 |
| OsclMemGlobalAuditObject . . . . .   | 426 |
| OsclMemoryFragment . . . . .   | 427 |
| OsclMemPoolAllocator . . . . .   | 428 |
| OsclMemPoolFixedChunkAllocator . . . . .   | 429 |
| OsclMemPoolFixedChunkAllocatorObserver . . . . .   | 433 |
| OsclMemPoolResizableAllocator . . . . .  | 434 |
| OsclMemPoolResizableAllocator::MemPoolBlockInfo . . . . .  | 440 |
| OsclMemPoolResizableAllocator::MemPoolBufferInfo . . . . .   | 441 |
| OsclMemPoolResizableAllocatorMemoryObserver . . . . .  | 442 |
| OsclMemPoolResizableAllocatorObserver . . . . .  | 443 |
| OsclMemStatsNode . . . . .   | 444 |
| OsclMutex . . . . .  | 445 |
| OsclNameString< __len > . . . . .  | 447 |
| OsclNativeFile . . . . .   | 448 |
| OsclNativeFileParams . . . . .   | 451 |
| OsclNetworkAddress . . . . .   | 452 |
| OsclNullLock . . . . .   | 453 |
| OsclPriorityLink . . . . .   | 454 |
| OsclPriorityList< T > . . . . .  | 455 |
| OsclPriorityQueue< Qelem, Alloc, Container, Compare > . . . . .  | 456 |
| OsclPriorityQueueBase . . . . .  | 460 |
| OsclProcStatus . . . . .   | 461 |
| OsclPtr . . . . .  | 463 |
| OsclPtrC . . . . .   | 465 |

|  |     |
|--|-----|
| OsclRand . . . . .   | 467 |
| OsclReadyAlloc . . . . .   | 468 |
| OsclReadyCompare . . . . .   | 469 |
| OsclReadyQ . . . . .   | 470 |
| OsclRecvFromMethod . . . . .   | 472 |
| OsclRecvFromRequest . . . . .  | 474 |
| OsclRecvMethod . . . . .   | 476 |
| OsclRecvRequest . . . . .  | 477 |
| OsclRefCounter . . . . .   | 478 |
| OsclRefCounterDA . . . . .   | 480 |
| OsclRefCounterMemFrag . . . . .  | 482 |
| OsclRefCounterMTDA< LockType > . . . . .   | 484 |
| OsclRefCounterMTSA< DeallocType, LockType > . . . . .  | 486 |
| OsclRefCounterSA< DeallocType > . . . . .  | 488 |
| OsclRegistryAccessClient . . . . .   | 490 |
| OsclRegistryAccessClientImpl . . . . .   | 492 |
| OsclRegistryAccessClientTlsImpl . . . . .  | 493 |
| OsclRegistryAccessElement . . . . .  | 494 |
| OsclRegistryClient . . . . .   | 495 |
| OsclRegistryClientImpl . . . . .   | 497 |
| OsclRegistryClientTlsImpl . . . . .  | 499 |
| OsclRegistryServTlsImpl . . . . .  | 500 |
| OsclScheduler . . . . .  | 502 |
| OsclSchedulerObserver . . . . .  | 503 |
| OsclScopedLock< LockClass > (Template class that handles unlocking an abstract class on destruction. This is very useful for ensuring that the lock is released when the OsclScopedLock goes out of scope) . . . . . | 504 |
| OsclSelect . . . . .   | 505 |
| OsclSemaphore . . . . .  | 507 |
| OsclSendMethod . . . . .   | 509 |
| OsclSendRequest . . . . .  | 510 |
| OsclSendToMethod . . . . .   | 511 |
| OsclSendToRequest . . . . .  | 512 |
| OsclSharedPtr< TheClass > (A parameterized smart pointer class) . . . . .  | 513 |
| OsclShutdownMethod . . . . .   | 516 |
| OsclShutdownRequest . . . . .  | 517 |
| OsclSingleton< T, ID, Registry > . . . . .   | 518 |
| OsclSingletonRegistry . . . . .  | 520 |
| OsclSocketI . . . . .  | 521 |
| OsclSocketIBase . . . . .  | 526 |
| OsclSocketMethod . . . . .   | 531 |
| OsclSocketObserver . . . . .   | 534 |
| OsclSocketRequest . . . . .  | 535 |
| OsclSocketRequestAO . . . . .  | 536 |
| OsclSocketServ . . . . .   | 540 |
| OsclSocketServI . . . . .  | 542 |
| OsclSocketServIBase . . . . .  | 544 |
| OsclSocketServRequestList . . . . .  | 546 |
| OsclSocketServRequestQELEM . . . . .   | 548 |
| OsclTCPSocket . . . . .  | 549 |
| OsclTCPSocketI . . . . .   | 555 |
| OsclThread . . . . .   | 558 |
| OsclThreadLock . . . . .   | 562 |
| OsclTickCount . . . . .  | 563 |

|   |     |
|---|-----|
| OsclTimer< Alloc >  | 565 |
| OsclTimerCompare  | 568 |
| OsclTimerObject   | 569 |
| OsclTimerObserver   | 573 |
| OsclTimerQ  | 574 |
| OsclTLS< T, ID, Registry >  | 575 |
| OsclTLSEx< T, ID, Registry >  | 577 |
| OsclTLSRegistry   | 579 |
| OsclTLSRegistryEx   | 580 |
| OsclTrapItem  | 581 |
| OsclTrapStack   | 582 |
| OsclTrapStackItem   | 583 |
| OsclUDPSocket   | 584 |
| OsclUDPSocketI  | 589 |
| OsclUuid  | 591 |
| PVActiveBase  | 593 |
| PVActiveStats   | 597 |
| PVLogger  | 598 |
| PVLoggerAppender  | 604 |
| PVLoggerFilter  | 605 |
| PVLoggerLayout  | 607 |
| PVLoggerRegistry  | 609 |
| PVSchedulerStopper  | 612 |
| PVSockBufRecv   | 613 |
| PVSockBufSend   | 614 |
| PVThreadContext   | 615 |
| RecvFromParam   | 617 |
| RecvParam   | 619 |
| SendParam   | 620 |
| SendToParam   | 621 |
| ShutdownParam   | 622 |
| SocketRequestParam  | 623 |
| StrCSumPtrLen (Same as <a href="#">StrPtrLen</a> , but includes checksum field and method to speed up querying) | 625 |
| StrPtrLen (This data structure encapsulates a set of functions used to perform)                                 | 628 |
| TimeValue (Time value in a format native to the system)   | 630 |
| TLSStorageOps   | 636 |
| TReadyQueLink   | 637 |
| WStrPtrLen (This data structure encapsulates a set of functions used to perform)                                | 638 |

# Chapter 4

## oscl File Index

### 4.1 oscl File List

Here is a list of all files with brief descriptions:

|  |     |
|--|-----|
| <code>oscl_aostatus.h</code> (Some basic types used with active objects) . . . . .   | 640 |
| <code>oscl_assert.h</code> (The file <code>oscl_assert.h</code> provides an OSCL_ASSERT macro to document assumptions and test them during development) . . . . .                                    | 641 |
| <code>oscl_base.h</code> (The file <code>oscl_base.h</code> is the public header that should be included to pick up the platform configuration, basic type definitions, and common macros) . . . . . | 642 |
| <code>oscl_base_alloc.h</code> (A basic allocator that does not rely on other modules) . . . . .   | 643 |
| <code>oscl_base_macros.h</code> (This file defines common macros and constants for basic compilation support)  | 644 |
| <code>oscl_bin_stream.h</code> (Defines a set of binary stream classes which handle portable input / output of binary data regardless of the native byte order) . . . . .                            | 645 |
| <code>oscl_byte_order.h</code> (This file defines functions providing byte ordering utility (e.g., switching between big and little endian orders)) . . . . .  | 646 |
| <code>oscl_defalloc.h</code> (The file defines simple default memory allocator classes. These allocators are used by the <code>Oscl_Vector</code> and <code>Oscl_Map</code> class, etc) . . . . .    | 647 |
| <code>oscl_dll.h</code> (Defines a DLL entry point) . . . . .  | 648 |
| <code>oscl_dns.h</code> (The file <code>oscl_socket.h</code> defines the OSCL DNS APIs) . . . . .  | 649 |
| <code>oscl_dns_gethostbyname.h</code> . . . . .  | 650 |
| <code>oscl_dns_imp.h</code> . . . . .  | 651 |
| <code>oscl_dns_imp_base.h</code> . . . . .   | 652 |
| <code>oscl_dns_imp_pv.h</code> . . . . .   | 653 |
| <code>oscl_dns_method.h</code> . . . . .   | 654 |
| <code>oscl_dns_param.h</code> . . . . .  | 655 |
| <code>oscl_dns_request.h</code> . . . . .  | 656 |
| <code>oscl_dns_tuneables.h</code> . . . . .  | 657 |
| <code>oscl_double_list.h</code> (Internal use types for scheduler) . . . . .   | 658 |
| <code>oscl_errno.h</code> (Defines functions to access additional information on errors where supported through an errno or similar service) . . . . .   | 659 |
| <code>oscl_error.h</code> (OSCL Error trap and cleanup include file) . . . . .   | 660 |
| <code>oscl_error_allocator.h</code> (Defines a memory allocation class used by the oscl error layer) . . . . .   | 661 |
| <code>oscl_error_codes.h</code> (Defines basic error and leave codes) . . . . .  | 662 |
| <code>oscl_error_imp.h</code> (Internal error implementation support) . . . . .  | 663 |
| <code>oscl_error_imp_cppexceptions.h</code> (Implementation File for Leave using C++ exceptions) . . . . .   | 664 |
| <code>oscl_error_imp_fatalerror.h</code> (Implementation File for Leave using system fatal error) . . . . .  | 665 |
| <code>oscl_error_imp_jumps.h</code> (Implemenation of using Setjmp / Longjmp) . . . . .  | 666 |

|   |     |
|---|-----|
| <a href="#">oscl_error_trapcleanup.h</a> (OSCL Error trap and cleanup implementation include file) . . . . .  | 668 |
| <a href="#">oscl_exception.h</a> (Contains all the exception handling macros and classes) . . . . .   | 669 |
| <a href="#">oscl_exclusive_ptr.h</a> (This file defines the <code>OsclExclusivePtr</code> template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error) . . . . .  | 670 |
| <a href="#">oscl_file_async_read.h</a> . . . . .  | 671 |
| <a href="#">oscl_file_cache.h</a> (The file <code>oscl_file_cache.h</code> defines the class <code>OsclFileCache</code> ) . . . . .   | 672 |
| <a href="#">oscl_file_dir_utils.h</a> (The file <code>oscl_file_dir_utils.h</code> defines some unix-style directory ops) . . . . .   | 673 |
| <a href="#">oscl_file_find.h</a> (The file <code>oscl_file_find.h</code> defines the class <code>Oscl_FileFind</code> ) . . . . .   | 675 |
| <a href="#">oscl_file_handle.h</a> (The file <code>oscl_file_handle.h</code> defines the class <code>OsclFileHandle</code> ) . . . . .  | 676 |
| <a href="#">oscl_file_io.h</a> (The file <code>oscl_file_io.h</code> defines the class <code>Oscl_File</code> . This is the public API to the basic file I/O operations) . . . . .  | 677 |
| <a href="#">oscl_file_native.h</a> (The file <code>oscl_file_native.h</code> defines the class <code>OsclNativeFile</code> . This is the porting layer for basic file I/O operations) . . . . .   | 678 |
| <a href="#">oscl_file_server.h</a> (The file <code>oscl_file_server.h</code> defines the class <code>Oscl_FileServer</code> . This is the porting layer for file server implementations) . . . . .  | 679 |
| <a href="#">oscl_file_stats.h</a> (File stats class) . . . . .  | 680 |
| <a href="#">oscl_file_types.h</a> (The file <code>oscl_file_types.h</code> defines some constants and types for file I/O implementations. Anything that needs to be shared across implementation modules can go here) . . . . .   | 681 |
| <a href="#">oscl_heapbase.h</a> (OSCL Heap Base include file) . . . . .   | 682 |
| <a href="#">oscl_init.h</a> (Global oscl initialization) . . . . .  | 683 |
| <a href="#">oscl_int64_utils.h</a> . . . . .  | 684 |
| <a href="#">oscl_ip_socket.h</a> . . . . .  | 685 |
| <a href="#">oscl_linked_list.h</a> (The file <code>oscl_linked_list.h</code> defines the template class <code>Oscl_Linked_List</code> which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter) . . . . .   | 686 |
| <a href="#">oscl_lock_base.h</a> (This file defines an abstract lock class, <code>OsclLockBase</code> , that is used for APIs potentially requiring multi-thread safety. A null-lock implementation, <code>OsclNullLock</code> , is also provided for single-thread configurations (basically a noop for lock/unlock). Also provides the <code>OsclScopedLock</code> class which is template class takes care of freeing the lock when the class goes out of scope) . . . . . | 687 |
| <a href="#">oscl_map.h</a> (The file <code>oscl_map.h</code> defines the template class <code>Oscl_Map</code> which has a very similar API as the STL Map class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter) . . . . .  | 688 |
| <a href="#">oscl_math.h</a> (Provides math functions) . . . . .   | 689 |
| <a href="#">oscl_media_data.h</a> (Defines a container class for media data made up of a collection of memory fragments) . . . . .  | 690 |
| <a href="#">oscl_media_status.h</a> (Defines a status values for the <code>MediaData</code> containers) . . . . .   | 691 |
| <a href="#">oscl_mem.h</a> (This file contains basic memory definitions for common use across platforms) . . . . .  | 692 |
| <a href="#">oscl_mem_align.h</a> . . . . .  | 695 |
| <a href="#">oscl_mem_audit.h</a> (This file contains the definition and partial implementation of MM_Audit class) . . . . .   | 696 |
| <a href="#">oscl_mem_audit_internals.h</a> (This file contains the internal definitions for the mem audit library) . . . . .  | 698 |
| <a href="#">oscl_mem_auto_ptr.h</a> (This file defines the <code>oscl_mem_auto_ptr</code> template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error) . . . . .  | 699 |
| <a href="#">oscl_mem_basic_functions.h</a> (This file contains prototypes for the basic memory functions) . . . . .   | 700 |
| <a href="#">oscl_mem_inst.h</a> (The file defines default memory instrumentation level) . . . . .   | 701 |
| <a href="#">oscl_mem_mempool.h</a> (This file contains the definition of memory pool allocators) . . . . .  | 702 |
| <a href="#">oscl_mempool_allocator.h</a> (This file contains the definition of memory pool allocator for leave/trap) . . . . .  | 703 |
| <a href="#">oscl_mutex.h</a> (This file provides implementation of mutex) . . . . .   | 704 |
| <a href="#">oscl_namestring.h</a> (Name string class include file) . . . . .  | 705 |

|   |     |
|---|-----|
| <code>oscl_opaque_type.h</code> (The file <code>oscl_opaque_type.h</code> defines pure virtual classes for working with opaque types) . . . . .   | 706 |
| <code>oscl_pqueue.h</code> (Implements a priority queue data structure similar to STL) . . . . .  | 707 |
| <code>oscl_proctstatus.h</code> . . . . .   | 708 |
| <code>oscl_queue.h</code> (The file <code>oscl_queue.h</code> defines the template class <code>Oscl_Queue</code> . It is similar to the <code>STL::queue</code> class, with some differences: - less complete - based on array rather than a deque - some interfaces modeled on <code>oscl_vector</code> , for ease of transition Memory allocation is abstracted through the use of an allocator template parameter) . . . . . | 709 |
| <code>oscl_rand.h</code> (Provides pseudo-random number generation) . . . . .   | 710 |
| <code>oscl_refcounter.h</code> (A general purpose reference counter to object lifetimes) . . . . .  | 711 |
| <code>oscl_refcounter_memfrag.h</code> (This file provides the definition of reference counted memory fragment, which provides access to a buffer and helps manage its lifetime through the refcount) . . . . .   | 712 |
| <code>oscl_registry_access_client.h</code> (Client-side implementation Registry Access implementation) . . . . .  | 713 |
| <code>oscl_registry_client.h</code> (Client-side implementation of <code>OsclRegistry</code> ) . . . . .  | 714 |
| <code>oscl_registry_client_impl.h</code> (Client-side implementation of <code>OsclRegistryInterface</code> ) . . . . .  | 715 |
| <code>oscl_registry_serv_impl.h</code> (Server-side implementation of <code>OsclRegistry</code> interfaces) . . . . .   | 716 |
| <code>oscl_registry_serv_impl_global.h</code> . . . . .   | 717 |
| <code>oscl_registry_serv_impl_tls.h</code> . . . . .  | 718 |
| <code>oscl_registry_types.h</code> (Common types used in <code>Oscl registry</code> interfaces) . . . . .   | 719 |
| <code>oscl_scheduler.h</code> . . . . .   | 720 |
| <code>oscl_scheduler_ao.h</code> ( <code>Oscl Scheduler</code> user execution object classes) . . . . .   | 721 |
| <code>oscl_scheduler_aobase.h</code> ( <code>Oscl Scheduler</code> internal active object classes) . . . . .  | 722 |
| <code>oscl_scheduler_readyq.h</code> (Ready q types for <code>oscl scheduler</code> ) . . . . .   | 723 |
| <code>oscl_scheduler_threadcontext.h</code> (Thread context functions needed by <code>oscl scheduler</code> ) . . . . .   | 724 |
| <code>oscl_scheduler_tuneables.h</code> (Tunable settings for <code>Oscl Scheduler</code> ) . . . . .   | 725 |
| <code>oscl_scheduler_types.h</code> (Scheduler common types include file) . . . . .   | 726 |
| <code>oscl_semaphore.h</code> (This file provides implementation of mutex) . . . . .  | 727 |
| <code>oscl_shared_ptr.h</code> (This file defines a template class <code>OsclSharedPtr</code> which is a "smart pointer" to the parameterized type) . . . . .   | 728 |
| <code>oscl_singleton.h</code> (This file defines the <code>OsclSingleton</code> class. This class provides a container which used to give access to a set of process-level singleton objects. Each object is indexed by an integer ID, listed below. There can only be one instance of each object per process at a given time) . . . . .   | 729 |
| <code>oscl_snprintf.h</code> (Provides a portable implementation of <code>snprintf</code> ) . . . . .   | 731 |
| <code>oscl_socket.h</code> (The file <code>oscl_socket.h</code> defines the OSCL Socket APIs) . . . . .   | 732 |
| <code>oscl_socket_accept.h</code> . . . . .   | 733 |
| <code>oscl_socket_bind.h</code> . . . . .   | 734 |
| <code>oscl_socket_connect.h</code> . . . . .  | 735 |
| <code>oscl_socket_imp.h</code> . . . . .  | 736 |
| <code>oscl_socket_imp_base.h</code> . . . . .   | 737 |
| <code>oscl_socket_imp_pv.h</code> . . . . .   | 738 |
| <code>oscl_socket_listen.h</code> . . . . .   | 739 |
| <code>oscl_socket_method.h</code> . . . . .   | 740 |
| <code>oscl_socket_recv.h</code> . . . . .   | 741 |
| <code>oscl_socket_recv_from.h</code> . . . . .  | 742 |
| <code>oscl_socket_request.h</code> . . . . .  | 743 |
| <code>oscl_socket_send.h</code> . . . . .   | 744 |
| <code>oscl_socket_send_to.h</code> . . . . .  | 745 |
| <code>oscl_socket_serv_imp.h</code> . . . . .   | 746 |
| <code>oscl_socket_serv_imp_base.h</code> . . . . .  | 747 |
| <code>oscl_socket_serv_imp_pv.h</code> . . . . .  | 748 |
| <code>oscl_socket_serv_imp_reqlist.h</code> . . . . .   | 749 |
| <code>oscl_socket_shutdown.h</code> . . . . .   | 750 |

|   |     |
|---|-----|
| <a href="#">oscl_socket_stats.h</a>   | 751 |
| <a href="#">oscl_socket_tunables.h</a>  | 753 |
| <a href="#">oscl_socket_types.h</a>   | 755 |
| <a href="#">oscl_stdstring.h</a> (This file provides standard string operations such as strlen, strncpy, etc. ANSI defines undefined behavior when the destination pointer is null for operations such as strncpy, strncat, etc. But, we chose to define one. In such cases, we return the destination as null)   | 757 |
| <a href="#">oscl_str_ptr_len.h</a> (Defines a data structure for string containment/manipulations where the storage for the string is maintained externally)  | 759 |
| <a href="#">oscl_string.h</a> (Provides a standardized set of string containers that can be used in place of character arrays)  | 760 |
| <a href="#">oscl_string_containers.h</a> (Provides a standardized set of string containers that can be used in place of character arrays)   | 761 |
| <a href="#">oscl_string_rep.h</a> (Contains some internal implementation for string containers)   | 762 |
| <a href="#">oscl_string_uri.h</a> (Utilities to unescape URIs)  | 763 |
| <a href="#">oscl_string_utf8.h</a> (Utilities to validate and truncate UTF-8 encoded strings)   | 764 |
| <a href="#">oscl_string_utils.h</a> (Utilities to parse and convert strings)  | 765 |
| <a href="#">oscl_string_xml.h</a> (Utilities to escape special characters in XML strings)   | 766 |
| <a href="#">oscl_tagtree.h</a> (The file <a href="#">oscl_tagtree.h</a> ..)   | 767 |
| <a href="#">oscl_tcp_socket.h</a>   | 768 |
| <a href="#">oscl_thread.h</a>   | 769 |
| <a href="#">oscl_tickcount.h</a> (Defines a data structure for string containment/manipulations where the storage for the string is maintained externally)  | 771 |
| <a href="#">oscl_time.h</a> (The file <a href="#">oscl_time.h</a> defines two classes <a href="#">NTPTime</a> and <a href="#">TimeValue</a> for getting, manipulating, and formatting time values. The <a href="#">TimeValue</a> class is based on the native system time format while <a href="#">NTPTime</a> is used for the standard Network Time Protocol format) | 772 |
| <a href="#">oscl_timer.h</a>  | 774 |
| <a href="#">oscl_tls.h</a>  | 775 |
| <a href="#">oscl_tree.h</a> (The file <a href="#">oscl_tree.h</a> defines the template class <a href="#">Oscl_Rb_Tree</a> which has a very similar API as the STL Tree class. It is an implementation of a Red-Black Tree for use by the <a href="#">Oscl_Map</a> class. Memory allocation is abstracted through the use of an allocator template parameter)          | 776 |
| <a href="#">oscl_types.h</a> (This file contains basic type definitions for common use across platforms)  | 777 |
| <a href="#">oscl_udp_socket.h</a>   | 778 |
| <a href="#">oscl_utf8conv.h</a> (Utilities to convert unicode to utf8 and vice versa)   | 779 |
| <a href="#">oscl_uuid.h</a> (This file defines the OSCL UUID structure used for unique identifiers as well as the short (32-bit) identifiers OsclUid32)   | 780 |
| <a href="#">oscl_vector.h</a> (The file <a href="#">oscl_vector.h</a> defines the template class <a href="#">Oscl_Vector</a> which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter)                                      | 781 |
| <a href="#">osclconfig.h</a> (This file contains configuration information for the linux platform)  | 782 |
| <a href="#">osclconfig_ansi_memory.h</a> (This file contains common typedefs based on the ANSI C limits.h header)   | 784 |
| <a href="#">osclconfig_check.h</a>  | 785 |
| <a href="#">osclconfig_compiler_warnings.h</a> (This file contains the ability to turn off/on compiler warnings)  | 786 |
| <a href="#">osclconfig_error.h</a> (This file contains the common typedefs and header files needed to compile osclerror)  | 787 |
| <a href="#">osclconfig_error_check.h</a>  | 788 |
| <a href="#">osclconfig_global_new_delete.h</a>  | 789 |
| <a href="#">osclconfig_global_placement_new.h</a>   | 790 |
| <a href="#">osclconfig_io.h</a> (This file contains common typedefs based on the ANSI C limits.h header)  | 791 |
| <a href="#">osclconfig_io_check.h</a>   | 798 |
| <a href="#">osclconfig_ix86.h</a> (This file contains configuration information for the ix86 processor family)  | 799 |

|   |     |
|---|-----|
| osclconfig_lib.h (This file contains configuration information for the ANSI build) . . . . .                            | 800 |
| osclconfig_lib_check.h . . . . .  | 801 |
| osclconfig_limits_typedefs.h (This file contains common typedefs based on the ANSI C limits.h header) . . . . .         | 802 |
| osclconfig_memory.h . . . . .   | 803 |
| osclconfig_memory_check.h . . . . .   | 804 |
| osclconfig_no_os.h . . . . .  | 805 |
| osclconfig_proc.h (This file contains configuration information for the linux platform) . . . . .                       | 806 |
| osclconfig_proc_check.h . . . . .   | 807 |
| osclconfig_proc_unix_android.h . . . . .  | 809 |
| osclconfig_proc_unix_common.h . . . . .   | 811 |
| osclconfig_time.h . . . . .   | 813 |
| osclconfig_time_check.h . . . . .   | 814 |
| osclconfig_unix_android.h . . . . .   | 815 |
| osclconfig_unix_common.h . . . . .  | 819 |
| osclconfig_util.h . . . . .   | 823 |
| osclconfig_util_check.h . . . . .   | 824 |
| pvlogger.h (This file contains basic logger interfaces for common use across platforms) . . . . .                       | 825 |
| pvlogger_accessories.h . . . . .  | 833 |
| pvlogger_c.h (This file contains basic logger interfaces for common use across platforms. C-callable version) . . . . . | 834 |
| pvlogger_registry.h . . . . .   | 836 |

# Chapter 5

## oscl Module Documentation

### 5.1 OSCL config

#### Defines

- #define OSCL\_ASSERT\_ALWAYS 0
- #define OSCL\_INTEGERS\_WORD\_ALIGNED 1
- #define OSCL\_BYTE\_ORDER\_BIG\_ENDIAN 0
- #define OSCL\_BYTE\_ORDER\_LITTLE\_ENDIAN 1
- #define OSCL\_HAS\_UNIX\_SUPPORT 0
- #define OSCL\_HAS\_MSWIN\_SUPPORT 0
- #define OSCL\_HAS\_MSWIN\_PARTIAL\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_SUPPORT 0
- #define OSCL\_HAS\_SAVAJE\_SUPPORT 0
- #define OSCL\_HAS\_PV\_C\_OS\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_ERRORTRAP 0
- #define OSCL\_HAS\_SYMBIAN\_MEMORY\_FUNCS 0
- #define OSCL\_HAS\_PV\_C\_OS\_API\_MEMORY\_FUNCS 0
- #define OSCL\_HAS\_PV\_C\_OS\_TIME\_FUNCS 0
- #define OSCL\_HAS\_UNIX\_TIME\_FUNCS 0
- #define OSCL\_HAS\_SYMBIAN\_TIMERS 0
- #define OSCL\_HAS\_SYMBIAN\_MATH 0
- #define OSCL\_HAS\_SYMBIAN\_SCHEDULER 0
- #define OSCL\_HAS\_SEM\_TIMEDWAIT\_SUPPORT 0
- #define OSCL\_HAS\_PTHREAD\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_COMPATIBLE\_IO\_FUNCTION 0
- #define OSCL\_HAS\_SAVAJE\_IO\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_SOCKET\_SERVER 0
- #define OSCL\_HAS\_SYMBIAN\_DNS\_SERVER 0
- #define OSCL\_HAS\_BERKELEY\_SOCKETS 0

## Typedefs

- `typedef int8 __int8__check__`
- `typedef uint8 __uint8__check__`
- `typedef int16 __int16__check__`
- `typedef uint16 __uint16__check__`
- `typedef int32 __int32__check__`
- `typedef uint32 __uint32__check__`

### 5.1.1 Define Documentation

#### 5.1.1.1 #define OSCL\_ASSERT\_ALWAYS 0

macro should be set to 0 or 1. When set to 1, OSCL\_ASSERT will be compiled in release mode as well as debug mode.

#### 5.1.1.2 #define OSCL\_BYTE\_ORDER\_BIG\_ENDIAN 0

macro should be set to 1 if the target platform uses big-endian byte order in memory. Otherwise it should be set to 0.

#### 5.1.1.3 #define OSCL\_BYTE\_ORDER\_LITTLE\_ENDIAN 1

macro should be set to 1 if the target platform uses little-endian byte order in memory. Otherwise it should be set to 0.

- 5.1.1.4 #define OSCL\_HAS\_BERKELEY\_SOCKETS 0
- 5.1.1.5 #define OSCL\_HAS\_MSWIN\_PARTIAL\_SUPPORT 0
- 5.1.1.6 #define OSCL\_HAS\_MSWIN\_SUPPORT 0
- 5.1.1.7 #define OSCL\_HAS\_PTHREAD\_SUPPORT 0
- 5.1.1.8 #define OSCL\_HAS\_PV\_C\_OS\_API\_MEMORY\_FUNCS 0
- 5.1.1.9 #define OSCL\_HAS\_PV\_C\_OS\_SUPPORT 0
- 5.1.1.10 #define OSCL\_HAS\_PV\_C\_OS\_TIME\_FUNCS 0
- 5.1.1.11 #define OSCL\_HAS\_SAVAJE\_IO\_SUPPORT 0
- 5.1.1.12 #define OSCL\_HAS\_SAVAJE\_SUPPORT 0
- 5.1.1.13 #define OSCL\_HAS\_SEM\_TIMEDWAIT\_SUPPORT 0
- 5.1.1.14 #define OSCL\_HAS\_SYMBIAN\_COMPATIBLE\_IO\_FUNCTION 0
- 5.1.1.15 #define OSCL\_HAS\_SYMBIAN\_DNS\_SERVER 0
- 5.1.1.16 #define OSCL\_HAS\_SYMBIAN\_ERRORTRAP 0
- 5.1.1.17 #define OSCL\_HAS\_SYMBIAN\_MATH 0
- 5.1.1.18 #define OSCL\_HAS\_SYMBIAN\_MEMORY\_FUNCS 0
- 5.1.1.19 #define OSCL\_HAS\_SYMBIAN\_SCHEDULER 0
- 5.1.1.20 #define OSCL\_HAS\_SYMBIAN\_SOCKET\_SERVER 0
- 5.1.1.21 #define OSCL\_HAS\_SYMBIAN\_SUPPORT 0
- 5.1.1.22 #define OSCL\_HAS\_SYMBIAN\_TIMERS 0
- 5.1.1.23 #define OSCL\_HAS\_UNIX\_SUPPORT 0
- 5.1.1.24 #define OSCL\_HAS\_UNIX\_TIME\_FUNCS 0
- 5.1.1.25 #define OSCL\_INTEGERS\_WORD\_ALIGNED 1

macro should be set to 1 if the target platform requires integers to be word-aligned in memory. Otherwise it should be set to 0.

## 5.1.2 Typedef Documentation

- 5.1.2.1 `typedef int16 __int16__check__`
- 5.1.2.2 `typedef int32 __int32__check__`
- 5.1.2.3 `typedef int8 __int8__check__`
- 5.1.2.4 `typedef uint16 __uint16__check__`
- 5.1.2.5 `typedef uint32 __uint32__check__`
- 5.1.2.6 `typedef uint8 __uint8__check__`

## 5.2 OSCL Base

### Files

- file [oscl\\_assert.h](#)

*The file [oscl\\_assert.h](#) provides an OSCL\_ASSERT macro to document assumptions and test them during development.*

- file [oscl\\_base.h](#)

*The file [oscl\\_base.h](#) is the public header that should be included to pick up the platform configuration, basic type definitions, and common macros.*

- file [oscl\\_base\\_alloc.h](#)

*A basic allocator that does not rely on other modules.*

- file [oscl\\_base\\_macros.h](#)

*This file defines common macros and constants for basic compilation support.*

- file [oscl\\_byte\\_order.h](#)

*This file defines functions providing byte ordering utility (e.g., switching between big and little endian orders).*

- file [oscl\\_defalloc.h](#)

*The file defines simple default memory allocator classes. These allocators are used by the [Oscl\\_Vector](#) and [Oscl\\_Map](#) class, etc.*

- file [oscl\\_dll.h](#)

*Defines a DLL entry point.*

- file [oscl\\_exclusive\\_ptr.h](#)

*This file defines the [OsclExclusivePtr](#) template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error.*

- file [oscl\\_linked\\_list.h](#)

*The file [oscl\\_linked\\_list.h](#) defines the template class [Oscl\\_Linked\\_List](#) which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.*

- file [oscl\\_lock\\_base.h](#)

*This file defines an abstract lock class, [OsclLockBase](#), that is used for APIs potentially requiring multi-thread safety. A null-lock implementation, [OsclNullLock](#), is also provided for single-thread configurations (basically a noop for lock/unlock). Also provides the [OsclScopedLock](#) class which is template class takes care of freeing the lock when the class goes out of scope.*

- file [oscl\\_map.h](#)

*The file [oscl\\_map.h](#) defines the template class [Oscl\\_Map](#) which has a very similar API as the STL Map class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.*

- file [oscl\\_mem\\_inst.h](#)

*The file defines default memory instrumentation level.*

- file [oscl\\_opaque\\_type.h](#)

*The file [oscl\\_opaque\\_type.h](#) defines pure virtual classes for working with opaque types.*

- file [oscl\\_queue.h](#)

*The file [oscl\\_queue.h](#) defines the template class [Oscl\\_Queue](#). It is similar to the `STL::queue` class, with some differences: - less complete - based on array rather than a deque - some interfaces modeled on `oscl_vector`, for ease of transition Memory allocation is abstracted through the use of an allocator template parameter.*

- file [oscl\\_refcounter.h](#)

*A general purpose reference counter to object lifetimes.*

- file [oscl\\_refcounter\\_memfrag.h](#)

*This file provides the definition of reference counted memory fragment, which provides access to a buffer and helps manage its lifetime through the refcount.*

- file [oscl\\_shared\\_ptr.h](#)

*This file defines a template class [OsclSharedPtr](#) which is a "smart pointer" to the parameterized type.*

- file [oscl\\_stdstring.h](#)

*This file provides standard string operations such as `strlen`, `strncpy`, etc. ANSI defines undefined behavior when the destination pointer is null for operations such as `strncpy`, `strncat`, etc. But, we chose to define one. In such cases, we return the destination as null.*

- file [oscl\\_tagtree.h](#)

*The file [oscl\\_tagtree.h](#) ...*

- file [oscl\\_time.h](#)

*The file [oscl\\_time.h](#) defines two classes [NTPTime](#) and [TimeValue](#) for getting, manipulating, and formatting time values. The [TimeValue](#) class is based on the native system time format while [NTPTime](#) is used for the standard Network Time Protocol format.*

- file [oscl\\_tree.h](#)

*The file [oscl\\_tree.h](#) defines the template class [Oscl\\_Rb\\_Tree](#) which has a very similar API as the `STL Tree` class. It is an implementation of a Red-Black Tree for use by the [Oscl\\_Map](#) class. Memory allocation is abstracted through the use of an allocator template parameter.*

- file [oscl\\_types.h](#)

*This file contains basic type definitions for common use across platforms.*

- file [oscl\\_vector.h](#)

*The file [oscl\\_vector.h](#) defines the template class [Oscl\\_Vector](#) which has a very similar API as the `STL Vector` class (it basically provides a subset of the `STL` functionality). Memory allocation is abstracted through the use of an allocator template parameter.*

## Data Structures

- class [\\_OsclBasicAllocator](#)
- class [LinkedListElement](#)
- class [NTPTime](#)

*The [NTPTime](#) class represents a time value as the number of seconds since 0h (UTC) Jan. 1, 1900.*

- class Oscl\_Alloc
- class Oscl\_Dealloc
- class Oscl\_DefAlloc
- class Oscl\_DefAllocWithRefCounter
- struct Oscl\_Less
- class Oscl\_Linked\_List
- class Oscl\_Linked\_List\_Base
- class Oscl\_Map
- class Oscl\_MTLinked\_List
- class Oscl\_Opaque\_Type\_Alloc
- class Oscl\_Opaque\_Type\_Alloc\_LL
- class Oscl\_Opaque\_Type\_Compare
- struct Oscl\_Pair
- class Oscl\_Queue
- class Oscl\_Queue\_Base
- class Oscl\_Rb\_Tree
- class Oscl\_Rb\_Tree\_Base
- struct Oscl\_Rb\_Tree\_Const\_Iterator
- struct Oscl\_Rb\_Tree\_Iterator
- struct Oscl\_Rb\_Tree\_Node
- struct Oscl\_Rb\_Tree\_Node\_Base
- struct Oscl\_Select1st
- struct Oscl\_Tag
- struct Oscl\_Tag\_Base
- class Oscl\_TagTree
- class Oscl\_TAlloc
- class Oscl\_Vector
- class Oscl\_Vector\_Base
- class OsclAllocDestructDealloc
- class OsclDestructDealloc
- class OsclExclusiveArrayPtr

The `OsclExclusiveArrayPtr` class is a template class that defines an array pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the `OsclExclusiveArrayPtr` expires, its destructor uses delete to free the memory.

- class OsclExclusivePtr

The `OsclExclusivePtr` class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the `OsclExclusivePtr` expires, its destructor uses delete to free the memory.

- class OsclExclusivePtrA

The `OsclExclusivePtrA` class is a template class that defines any pointer like object intended to be assigned an address obtained (directly or or indirectly) through Alloc. When the `OsclExclusivePtrA` expires, Alloc is used to free the memory.

- class OsclLockBase
- struct OsclMemoryFragment
- class OsclNullLock
- class OsclRefCounter
- class OsclRefCounterDA

- class [OsclRefCounterMemFrag](#)
- class [OsclRefCounterMTDA](#)
- class [OsclRefCounterMTSA](#)
- class [OsclRefCounterSA](#)
- class [OsclScopedLock](#)

*The OsclScopedLock class is a template class that handles unlocking an abstract class on destruction. This is very useful for ensuring that the lock is released when the OsclScopedLock goes out of scope.*

- class [OsclSharedPtr](#)

*A parameterized smart pointer class.*

- class [OsclTLS](#)
- class [OsclTLSRegistry](#)
- class [TimeValue](#)

*The TimeValue class represents a time value in a format native to the system.*

- class [TLSStorageOps](#)

## Defines

- #define [OSCL\\_ASSERT](#)(*\_expr*) ((*\_expr*)?((void)0):OSCL Assert(# *\_expr*, \_\_FILE\_\_, \_\_LINE\_\_))
- #define [OSCL\\_HAS\\_SINGLETON\\_SUPPORT](#) 1
- #define [NULL\\_TERM\\_CHAR](#) '\0'

*The NULL\_TERM\_CHAR is used to terminate c-style strings.*

- #define [NULL](#) (0)

*if the NULL macro isn't already defined, then define it as zero.*

- #define [OSCL\\_INLINE](#) inline
- #define [OSCL\\_COND\\_EXPORT\\_REF](#)
- #define [OSCL\\_COND\\_IMPORT\\_REF](#)
- #define [OSCL\\_CONST\\_CAST](#)(*type*, *exp*) ((*type*)(*exp*))

*Type casting macros.*

- #define [OSCL\\_STATIC\\_CAST](#)(*type*, *exp*) ((*type*)(*exp*))
- #define [OSCL\\_REINTERPRET\\_CAST](#)(*type*, *exp*) ((*type*)(*exp*))
- #define [OSCL\\_DYNAMIC\\_CAST](#)(*type*, *exp*) ((*type*)(*exp*))
- #define [OSCL\\_UNUSED\\_ARG](#)(*vbl*) (void)(*vbl*)
- #define [OSCL\\_UNUSED\\_RETURN](#)(*value*) return *value*
- #define [OSCL\\_MIN](#)(*a*, *b*) ((*a*) < (*b*) ? (*a*) : (*b*))
- #define [OSCL\\_MAX](#)(*a*, *b*) ((*a*) > (*b*) ? (*a*) : (*b*))
- #define [OSCL\\_ABS](#)(*a*) ((*a*) > (0) ? (*a*) : -(*a*))
- #define [OSCL\\_TEMPLATED\\_DESTRUCTOR\\_CALL](#)(*type*, *simple\_type*) *type* :: ~*simple\_type* ()
- #define [OSCL\\_UNSIGNED\\_CONST](#)(*x*) *x*
- #define [OSCL\\_PACKED\\_VAR](#) "error"
- #define [OSCL\\_BEGIN\\_PACKED](#) "error"
- #define [OSCL\\_END\\_PACKED](#) "error"
- #define [OSCL\\_DISABLE\\_WARNING\\_TRUNCATE\\_DEBUG\\_MESSAGE](#)
- #define [ALLOCATE](#)(*n*) *allocate\_fl*(*n*, \_\_FILE\_\_, \_\_LINE\_\_)
- #define [ALLOC\\_AND\\_CONSTRUCT](#)(*n*) *alloc\_and\_construct\_fl*(*n*, \_\_FILE\_\_, \_\_LINE\_\_)

- #define **OSCL\_DLL\_ENTRY\_POINT()** void oscl\_dll\_entry\_point() {}
- #define **OSCL\_DLL\_ENTRY\_POINT\_DEFAULT()**
- #define **PVMEM\_INST\_LEVEL** 1
- #define **OSCL\_DISABLE\_WARNING\_RETURN\_TYPE\_NOT\_UDT**
- #define **OSCL\_TLS\_BASE\_SLOTS** OSCL\_TLS\_ID\_BASE\_LAST +1
- #define **OSCL\_TLS\_EXTERNAL\_SLOTS** 0
- #define **OSCL\_TLS\_MAX\_SLOTS** ( OSCL\_TLS\_BASE\_SLOTS + OSCL\_TLS\_EXTERNAL\_SLOTS)

## Typedefs

- typedef char **CtimeStrBuf** [CTIME\_BUFFER\_SIZE]
- typedef char **PV8601timeStrBuf** [PV8601TIME\_BUFFER\_SIZE]
- typedef **OsclAny** TOsclTlsKey
- typedef int **c\_bool**  
*The c\_bool type is mapped to an integer to provide a bool type for C interfaces.*
- typedef void **OsclAny**  
*The OsclAny is meant to be used the context of a generic pointer (i.e., no specific type).*
- typedef char **mbchar**  
*mbchar is multi-byte char (e.g., UTF-8) with null termination.*
- typedef unsigned int **uint**  
*The uint type is a convenient abbreviation for unsigned int.*
- typedef uint8 **octet**  
*The octet type is meant to be used for referring to a byte or collection bytes without suggesting anything about the underlying meaning of the bytes.*
- typedef float **OsclFloat**  
*The Float type defined as OsclFloat.*
- typedef OSCL\_NATIVE\_INT64\_TYPE **int64**
- typedef OSCL\_NATIVE\_UINT64\_TYPE **uint64**
- typedef OSCL\_NATIVE\_WCHAR\_TYPE **oscl\_wchar**
- typedef **oscl\_wchar OSCL\_TCHAR**  
*define OSCL\_TCHAR*

## Enumerations

- enum **TimeUnits** { **SECONDS** = 0, **MILLISECONDS** = 1, **MICROSECONDS** = 2 }
- The TimeUnits enum can be used when constructing a **TimeValue** class.*

## Functions

- OSCL\_COND\_IMPORT\_REF void [\\_OSCL\\_Abort\(\)](#)  
*This function terminates the current process abnormally.*
- OSCL\_IMPORT\_REF void [OSCL\\_Assert](#) (const char \*expr, const char \*filename, int line\_number)  
*OSCL\_ASSERT macro evaluates an expression and when the result is false, prints a diagnostic message and aborts the program.*
- void [PVOsclBase\\_Init\(\)](#)
- void [PVOsclBase\\_Cleanup\(\)](#)
- void [little\\_endian\\_to\\_host](#) (char \*data, uint32 size)  
*Convert little endian to host format.*
- void [host\\_to\\_little\\_endian](#) (char \*data, unsigned int size)  
*Convert host to little endian format.*
- void [big\\_endian\\_to\\_host](#) (char \*data, unsigned int size)  
*Convert big endian to host format.*
- void [host\\_to\\_big\\_endian](#) (char \*data, unsigned int size)  
*Convert host to big endian format.*
- OSCL\_IMPORT\_REF uint32 [oscl\\_strlen](#) (const char \*str)
- OSCL\_IMPORT\_REF uint32 [oscl\\_strlen](#) (const [oscl\\_wchar](#) \*str)
- OSCL\_IMPORT\_REF char \* [oscl\\_strncpy](#) (char \*dest, const char \*src, uint32 count)
- OSCL\_IMPORT\_REF [oscl\\_wchar](#) \* [oscl\\_strncpy](#) ([oscl\\_wchar](#) \*dest, const [oscl\\_wchar](#) \*src, uint32 count)
- OSCL\_IMPORT\_REF int32 [oscl\\_strcmp](#) (const char \*str1, const char \*str2)
- OSCL\_IMPORT\_REF int32 [oscl\\_strcmp](#) (const [oscl\\_wchar](#) \*str1, const [oscl\\_wchar](#) \*str2)
- OSCL\_IMPORT\_REF int32 [oscl\\_strncmp](#) (const char \*str1, const char \*str2, uint32 count)
- OSCL\_IMPORT\_REF int32 [oscl\\_strncmp](#) (const [oscl\\_wchar](#) \*str1, const [oscl\\_wchar](#) \*str2, uint32 count)
- OSCL\_IMPORT\_REF char \* [oscl\\_strncat](#) (char \*dest, const char \*src, uint32 count)
- OSCL\_IMPORT\_REF [oscl\\_wchar](#) \* [oscl\\_strncat](#) ([oscl\\_wchar](#) \*dest, const [oscl\\_wchar](#) \*src, uint32 count)
- OSCL\_IMPORT\_REF const char \* [oscl\\_strchr](#) (const char \*str, int32 c)
- OSCL\_IMPORT\_REF char \* [oscl\\_strchr](#) (char \*str, int32 c)
- OSCL\_IMPORT\_REF const [oscl\\_wchar](#) \* [oscl\\_strchr](#) (const [oscl\\_wchar](#) \*str, int32 c)
- OSCL\_IMPORT\_REF [oscl\\_wchar](#) \* [oscl\\_strchr](#) ([oscl\\_wchar](#) \*str, int32 c)
- OSCL\_IMPORT\_REF const char \* [oscl strrchr](#) (const char \*str, int32 c)
- OSCL\_IMPORT\_REF char \* [oscl strrchr](#) (char \*str, int32 c)
- OSCL\_IMPORT\_REF const [oscl\\_wchar](#) \* [oscl strrchr](#) (const [oscl\\_wchar](#) \*str, int32 c)
- OSCL\_IMPORT\_REF [oscl\\_wchar](#) \* [oscl strrchr](#) ([oscl\\_wchar](#) \*str, int32 c)
- OSCL\_IMPORT\_REF char \* [oscl\\_strset](#) (char \*dest, char val, uint32 count)
- OSCL\_IMPORT\_REF [oscl\\_wchar](#) \* [oscl\\_strset](#) ([oscl\\_wchar](#) \*dest, [oscl\\_wchar](#) val, uint32 count)
- OSCL\_IMPORT\_REF int32 [oscl\\_CIstrcmp](#) (const char \*str1, const char \*str2)
- OSCL\_IMPORT\_REF int32 [oscl\\_CIstrcmp](#) (const [oscl\\_wchar](#) \*str1, const [oscl\\_wchar](#) \*str2)
- OSCL\_IMPORT\_REF int32 [oscl\\_CIstrncmp](#) (const char \*str1, const char \*str2, uint32 count)

- OSCL\_IMPORT\_REF int32 `oscl_Clstrncmp` (const `oscl_wchar` \*str1, const `oscl_wchar` \*str2, uint32 count)
- OSCL\_IMPORT\_REF char `oscl_tolower` (const char car)
- OSCL\_IMPORT\_REF `oscl_wchar oscl_tolower` (const `oscl_wchar` car)
- OSCL\_IMPORT\_REF bool `oscl_isLetter` (const char car)
- OSCL\_IMPORT\_REF const char \* `oscl_strstr` (const char \*str1, const char \*str2)
- OSCL\_IMPORT\_REF char \* `oscl_strstr` (char \*str1, const char \*str2)
- OSCL\_IMPORT\_REF const `oscl_wchar` \* `oscl_strstr` (const `oscl_wchar` \*str1, const `oscl_wchar` \*str2)
- OSCL\_IMPORT\_REF `oscl_wchar` \* `oscl_strstr` (`oscl_wchar` \*str1, const `oscl_wchar` \*str2)
- OSCL\_IMPORT\_REF char \* `oscl_streat` (char \*dest, const char \*src)
- OSCL\_IMPORT\_REF `oscl_wchar` \* `oscl_streat` (`oscl_wchar` \*dest, const `oscl_wchar` \*src)
- OSCL\_IMPORT\_REF void `PV8601ToRFC822` (`PV8601timeStrBuf` pv8601\_buffer, `CtimeStrBuf` ctime\_buffer)
- OSCL\_IMPORT\_REF void `RFC822ToPV8601` (`CtimeStrBuf` ctime\_buffer, `PV8601timeStrBuf`)
- OSCL\_COND\_IMPORT\_REF `TimeValue operator-` (const `TimeValue` &a, const `TimeValue` &b)
- bool `operator==` (const `OsclSharedPtr` &b) const

*Test for equality to see if two PVHandles wrap the same object.*

- void `Bind` (const `OsclSharedPtr` &inHandle)

*Use this function to bind an existing `OsclSharedPtr` to a already-wrapped object.*

- void `Bind` (TheClass \*ptr, `OsclRefCounter` \*in\_refcnt)

*Use this function to bind an existing `OsclSharedPtr` to a new (unwrapped) object.*

## Variables

- const int `CTIME_BUFFER_SIZE` = 26
- const int `PV8601TIME_BUFFER_SIZE` = 21
- const long `USEC_PER_SEC` = 1000000
- const long `MSEC_PER_SEC` = 1000
- const uint32 `unix_ntp_offset` = 2208988800U
- const uint32 `OSCL_TLS_ID_MAGICNUM` = 0
- const uint32 `OSCL_TLS_ID_ERRORHOOK` = 1
- const uint32 `OSCL_TLS_ID_PVLOGGER` = 2
- const uint32 `OSCL_TLS_ID_TEST` = 3
- const uint32 `OSCL_TLS_ID_PVSCHEDULER` = 4
- const uint32 `OSCL_TLS_ID_PVERRORTRAP` = 5
- const uint32 `OSCL_TLS_ID_SDPMEDIAPARSER` = 6
- const uint32 `OSCL_TLS_ID_PAYLOADPARSER` = 7
- const uint32 `OSCL_TLS_ID_PVMFRECOGNIZER` = 8
- const uint32 `OSCL_TLS_ID_WMDRM` = 9
- const uint32 `OSCL_TLS_ID_OSCLREGISTRY` = 10
- const uint32 `OSCL_TLS_ID_SQLITE3` = 11
- const uint32 `OSCL_TLS_ID_BASE_LAST` = 11

### 5.2.1 Detailed Description

Additional osclbase comment

Additional osclbase comment

Additional osclbase comment

### 5.2.2 Define Documentation

**5.2.2.1 #define ALLOC\_AND\_CONSTRUCT(n) alloc\_and\_construct\_fl(n,\_\_FILE\_\_,\_\_LINE\_\_)**

**5.2.2.2 #define ALLOCATE(n) allocate\_fl(n,\_\_FILE\_\_,\_\_LINE\_\_)**

**5.2.2.3 #define NULL (0)**

if the NULL macro isn't already defined, then define it as zero.

**5.2.2.4 #define NULL\_TERM\_CHAR '\0'**

The NULL\_TERM\_CHAR is used to terminate c-style strings.

**5.2.2.5 #define OSCL\_ABS(a) ((a) > (0) ? (a) : -(a))**

**5.2.2.6 #define OSCL\_ASSERT(\_expr) ((\_expr)?((void)0):OSCLAssert(#\_expr,\_\_FILE\_\_,\_\_LINE\_\_))**

**5.2.2.7 #define OSCL\_BEGIN\_PACKED "error"**

**5.2.2.8 #define OSCL\_COND\_EXPORT\_REF**

**5.2.2.9 #define OSCL\_COND\_IMPORT\_REF**

**5.2.2.10 #define OSCL\_CONST\_CAST(type, exp) ((type)(exp))**

Type casting macros.

**Parameters:**

*type* Destination type of cast

*exp* Expression to cast

**5.2.2.11 #define OSCL\_DISABLE\_WARNING\_RETURN\_TYPE\_NOT\_UDT**

**5.2.2.12 #define OSCL\_DISABLE\_WARNING\_TRUNCATE\_DEBUG\_MESSAGE**

**5.2.2.13 #define OSCL\_DLL\_ENTRY\_POINT() void oscl\_dll\_entry\_point() {}**

DLL entry/exit point.

Allows you to define custom operations at the entry and exit of the DLL. Place this macro within one source file for each DLL.

Functions with the custom commands for the DLL entry and exit point must also be defined. The entry point custom function is LocalDllEntry(), and the exit point custom function is LocalDllExit().

These functions will be called as a result of executing this macro.

Usage :

```
LocalDllEntry() { custom operations... }  
LocalDllExit() { custom operations... }  
OSCL_DLL_ENTRY_POINT()
```

#### **5.2.2.14 #define OSCL\_DLL\_ENTRY\_POINT\_DEFAULT()**

Default DLL entry/exit point function.

The body of the DLL entry point is given. The macro only needs to be declared within the source file.

Usage :

```
OSCL_DLL_ENTRY_POINT_DEFAULT()
```

#### **5.2.2.15 #define OSCL\_DYNAMIC\_CAST(type, exp) ((type)(exp))**

#### **5.2.2.16 #define OSCL\_END\_PACKED "error"**

#### **5.2.2.17 #define OSCL\_HAS\_SINGLETON\_SUPPORT 1**

#### **5.2.2.18 #define OSCL\_INLINE inline**

#### **5.2.2.19 #define OSCL\_MAX(a, b) ((a) > (b) ? (a) : (b))**

#### **5.2.2.20 #define OSCL\_MIN(a, b) ((a) < (b) ? (a) : (b))**

#### **5.2.2.21 #define OSCL\_PACKED\_VAR "error"**

#### **5.2.2.22 #define OSCL\_REINTERPRET\_CAST(type, exp) ((type)(exp))**

#### **5.2.2.23 #define OSCL\_STATIC\_CAST(type, exp) ((type)(exp))**

#### **5.2.2.24 #define OSCL\_TEMPLATED\_DESTRUCTOR\_CALL(type, simple\_type) type :: ~simple\_type ()**

#### **5.2.2.25 #define OSCL\_TLS\_BASE\_SLOTS OSCL\_TLS\_ID\_BASE\_LAST +1**

#### **5.2.2.26 #define OSCL\_TLS\_EXTERNAL\_SLOTS 0**

#### **5.2.2.27 #define OSCL\_TLS\_MAX\_SLOTS ( OSCL\_TLS\_BASE\_SLOTS + OSCL\_TLS\_EXTERNAL\_SLOTS )**

#### **5.2.2.28 #define OSCL\_UNSIGNED\_CONST(x) x**

#### **5.2.2.29 #define OSCL\_UNUSED\_ARG(vbl) (void)(vbl)**

The following two macros are used to avoid compiler warnings.

`OSCL_UNUSED_ARG(vbl)` is used to "reference" an otherwise unused parameter or variable, often one which is used only in an `OSCL_ASSERT` and thus unreferenced in release mode. `OSCL_UNUSED_RETURN(val)` provides a "return" of a value, in places which will not actually be executed, such as after an `OSCL_LEAVE` or `Thread::exit` or `abort`. The value needs to be of an appropriate type for the current function, though zero will usually suffice. Note that `OSCL_UNUSED_RETURN` will not be necessary for 'void' functions, as there is no requirement for a value-return operation.

#### 5.2.2.30 `#define OSCL_UNUSED_RETURN(value) return value`

#### 5.2.2.31 `#define PVMEM_INST_LEVEL 1`

### 5.2.3 Typedef Documentation

#### 5.2.3.1 `typedef int c_bool`

The `c_bool` type is mapped to an integer to provide a `bool` type for C interfaces.

#### 5.2.3.2 `typedef char CtimeStrBuff[CTIME_BUFFER_SIZE]`

#### 5.2.3.3 `typedef OSCL_NATIVE_INT64_TYPE int64`

#### 5.2.3.4 `typedef char mbchar`

`mbchar` is multi-byte char (e.g., UTF-8) with null termination.

#### 5.2.3.5 `typedef uint8 octet`

The `octet` type is meant to be used for referring to a byte or collection bytes without suggesting anything about the underlying meaning of the bytes.

#### 5.2.3.6 `typedef oscl_wchar OSCL_TCHAR`

define `OSCL_TCHAR`

#### 5.2.3.7 `typedef OSCL_NATIVE_WCHAR_TYPE oscl_wchar`

#### 5.2.3.8 `typedef void OsclAny`

The `OsclAny` is meant to be used the context of a generic pointer (i.e., no specific type).

#### 5.2.3.9 `typedef float OsclFloat`

The `Float` type defined as `OsclFloat`.

**5.2.3.10 `typedef char PV8601timeStrBuf[PV8601TIME_BUFFER_SIZE]`**

**5.2.3.11 `typedef OsclAny TOsclTlsKey`**

**5.2.3.12 `typedef unsigned int uint`**

The uint type is a convenient abbreviation for unsigned int.

**5.2.3.13 `typedef OSCL_NATIVE_UINT64_TYPE uint64`**

## 5.2.4 Enumeration Type Documentation

**5.2.4.1 `enum TimeUnits`**

The TimeUnits enum can be used when constructing a [TimeValue](#) class.

**Enumeration values:**

**SECONDS**

**MILLISECONDS**

**MICROSECONDS**

## 5.2.5 Function Documentation

**5.2.5.1 `OSCL_COND_IMPORT_REF void _OSCL_Abort()`**

This function terminates the current process abnormally.

**5.2.5.2 `void big_endian_to_host (char * data, unsigned int size)`**

Convert big endian to host format.

This function takes a buffer of data which is assumed to be in big endian order and rearranges it to the native order of the machine running the code. If the machine is a big endian machine, nothing is done.

**Parameters:**

*data* A pointer to the input/output buffer

*size* The number of bytes in the buffer.

**5.2.5.3 `template<class TheClass> void OsclSharedPtr< TheClass >::Bind (TheClass * ptr, OsclRefCounter * in_refcnt) [inline, inherited]`**

Use this function to bind an existing OsclSharedPtr to a new (unwrapped) object.

**5.2.5.4 `template<class TheClass> void OsclSharedPtr< TheClass >::Bind (const OsclSharedPtr< TheClass > & inHandle) [inline, inherited]`**

Use this function to bind an existing OsclSharedPtr to a already-wrapped object.

### 5.2.5.5 void host\_to\_big\_endian (char \* *data*, unsigned int *size*)

Convert host to big endian format.

This function takes a buffer of data which is assumed to be in native host order and rearranges it to big endian format. If the machine is a big endian machine, nothing is done.

**Parameters:**

- data* A pointer to the input/output buffer
- size* The number of bytes in the buffer.

### 5.2.5.6 void host\_to\_little\_endian (char \* *data*, unsigned int *size*)

Convert host to little endian format.

This function takes a buffer of data which is assumed to be in the host's native order and rearranges it to the little endian format. If the machine is a little endian machine, nothing is done.

**Parameters:**

- data* A pointer to the input/output buffer
- size* The number of bytes in the buffer.

### 5.2.5.7 void little\_endian\_to\_host (char \* *data*, uint32 *size*)

Convert little endian to host format.

This function takes a buffer of data which is assumed to be in little endian order and rearranges it to the native order of the machine running the code. If the machine is a little endian machine, nothing is done.

**Parameters:**

- data* A pointer to the input/output buffer
- size* The number of bytes in the buffer.

### 5.2.5.8 OSCL\_COND\_IMPORT\_REF TimeValue operator- (const TimeValue & *a*, const TimeValue & *b*)

### 5.2.5.9 template<class TheClass> bool OsclSharedPtr<TheClass>::operator== (const OsclSharedPtr<TheClass> & *b*) const [inline, inherited]

Test for equality to see if two PVHandles wrap the same object.

### 5.2.5.10 OSCL\_IMPORT\_REF void OSCL\_Assert (const char \* *expr*, const char \* *filename*, int *line\_number*)

OSCL\_ASSERT macro evaluates an expression and when the result is false, prints a diagnostic message and aborts the program.

**Parameters:**

- expr* is the expression to be evaluated
- filename* is the name of the current source file
- line\_number* is the line number in the current source file

**5.2.5.11 OSCL\_IMPORT\_REF int32 oscl\_CIstrcmp (const oscl\_wchar \* str1, const oscl\_wchar \* str2)**

Case in-sensitive string comparision.

**Parameters:**

- str1* string to compare
- str2* string to compare

**Returns:**

Negative if str1 < str2 Positive if str1 > str2 Zero if equal

**5.2.5.12 OSCL\_IMPORT\_REF int32 oscl\_CIstrcmp (const char \* str1, const char \* str2)**

Case in-sensitive string comparision.

**Parameters:**

- str1* string to compare
- str2* string to compare

**Returns:**

Negative if str1 < str2 Positive if str1 > str2 Zero if equal

**5.2.5.13 OSCL\_IMPORT\_REF int32 oscl\_CIstrncmp (const oscl\_wchar \* str1, const oscl\_wchar \* str2, uint32 count)**

Lexicographically compares(case in-sensitive), at most, the first count characters in str1 and str2 and returns a value indicating the relationship between the substrings.

**Parameters:**

- str1* string to compare
- str2* string to compare
- count* Number of characters to compare

**Returns:**

Negative if str1 < str2 Positive if str1 > str2 Zero if equal

**5.2.5.14 OSCL\_IMPORT\_REF int32 oscl\_CIstrncmp (const char \* str1, const char \* str2, uint32 count)**

Lexicographically compares(case in-sensitive), at most, the first count characters in str1 and str2 and returns a value indicating the relationship between the substrings.

**Parameters:**

- str1* string to compare
- str2* string to compare
- count* Number of characters to compare

**Returns:**

Negative if str1 < str2 Positive if str1 > str2 Zero if equal

**5.2.5.15 OSCL\_IMPORT\_REF bool oscl\_isLetter (const char *car*)**

check if supplied parameter is an alphabet (ASCII only).

**Parameters:**

*car*

**Returns:**

1 if car is an alphabet 0 if car is not an alphabet.

**5.2.5.16 OSCL\_IMPORT\_REF oscl\_wchar\* oscl\_streat (oscl\_wchar \* *dest*, const oscl\_wchar \* *src*)**

Appends up to count characters from string src to string dest, and then appends a terminating null character. The initial character of src overwrites the null character at the end of dest. Subsequent characters in src are appended to dest until either the end of src is reached or count characters have been copied. If copying takes place between objects that overlap, the behavior is undefined.

**Parameters:**

*dest* null terminated destination string

*src* source string

*count* number of characters to append.

**Returns:**

dest

**5.2.5.17 OSCL\_IMPORT\_REF char\* oscl\_streat (char \* *dest*, const char \* *src*)**

Appends string src to string dest, and then appends a terminating null character. The initial character of src overwrites the null character at the end of dest. Subsequent characters in src are appended to dest until the end of src is reached. If copying takes place between objects that overlap, the behavior is undefined.

**Parameters:**

*dest* null terminated destination string

*src* source string

**Returns:**

dest

**5.2.5.18 OSCL\_IMPORT\_REF oscl\_wchar\* oscl\_strchr (oscl\_wchar \* *str*, int32 *c*)****5.2.5.19 OSCL\_IMPORT\_REF const oscl\_wchar\* oscl\_strchr (const oscl\_wchar \* *str*, int32 *c*)**

Finds the first occurrence of c in string, or it returns NULL if c is not found. The null-terminating character is included in the search.

**Parameters:**

*str* null terminated source string

*c* character to search for

**Returns:**

**5.2.5.20 OSCL\_IMPORT\_REF char\* oscl\_strchr (char \* str, int32 c)****5.2.5.21 OSCL\_IMPORT\_REF const char\* oscl\_strchr (const char \* str, int32 c)**

Finds the first occurrence of c in string, or it returns NULL if c is not found. The null-terminating character is included in the search.

**Parameters:**

*str* null terminated source string

*c* character to search for

**Returns:****5.2.5.22 OSCL\_IMPORT\_REF int32 oscl\_strcmp (const oscl\_wchar \* str1, const oscl\_wchar \* str2)**

Lexicographically compares two NULL terminated strings, str1 and str2, and returns a value indicating the relationship between them.

**Parameters:**

*str1* String to compare

*str2* String to compare

**Returns:**

Negative if str1 < str2 Positive if str1 > str2 Zero if equal

**5.2.5.23 OSCL\_IMPORT\_REF int32 oscl\_strcmp (const char \* str1, const char \* str2)**

Lexicographically compares two NULL terminated strings, str1 and str2, and returns a value indicating the relationship between them.

**Parameters:**

*str1* String to compare

*str2* String to compare

**Returns:**

Negative if str1 < str2 Positive if str1 > str2 Zero if equal

**5.2.5.24 OSCL\_IMPORT\_REF uint32 oscl\_strlen (const oscl\_wchar \* str)**

Gets the length of a wide char string

**Parameters:**

*str* NULL terminated string.

**Returns:**

Returns the number of characters in string, excluding the terminal NULL.

**5.2.5.25 OSCL\_IMPORT\_REF uint32 oscl\_strlen (const char \* str)**

Gets the length of a string

**Parameters:**

*str* NULL terminated string.

**Returns:**

Returns the number of characters in string, excluding the terminal NULL.

**5.2.5.26 OSCL\_IMPORT\_REF oscl\_wchar\* oscl\_strncat (oscl\_wchar \* dest, const oscl\_wchar \* src, uint32 count)**

Appends up to count characters from string src to string dest, and then appends a terminating null character. The initial character of src overwrites the null character at the end of dest. Subsequent characters in src are appended to dest until either the end of src is reached or count characters have been copied. If copying takes place between objects that overlap, the behavior is undefined.

**Parameters:**

*dest* null terminated destination string

*src* source string

*count* number of characters to append.

**Returns:**

dest

**5.2.5.27 OSCL\_IMPORT\_REF char\* oscl\_strncat (char \* dest, const char \* src, uint32 count)**

Appends up to count characters from string src to string dest, and then appends a terminating null character. The initial character of src overwrites the null character at the end of dest. Subsequent characters in src are appended to dest until either the end of src is reached or count characters have been copied. If copying takes place between objects that overlap, the behavior is undefined.

**Parameters:**

*dest* null terminated destination string

*src* source string

*count* number of characters to append.

**Returns:**

dest

**5.2.5.28 OSCL\_IMPORT\_REF int32 oscl\_strcmp (const oscl\_wchar \* str1, const oscl\_wchar \* str2, uint32 count)**

Lexicographically compares, at most, the first count characters in str1 and str2 and returns a value indicating the relationship between the substrings.

**Parameters:**

*str1* String to compare

***str2*** String to compare

***count*** Number of characters to compare

**Returns:**

Negative if str1 < str2 Positive if str1 > str2 Zero if equal

#### 5.2.5.29 OSCL\_IMPORT\_REF int32 oscl\_strncmp (const char \* *str1*, const char \* *str2*, uint32 *count*)

Lexicographically compares, at most, the first count characters in str1 and str2 and returns a value indicating the relationship between the substrings.

**Parameters:**

***str1*** String to compare

***str2*** String to compare

***count*** Number of characters to compare

**Returns:**

Negative if str1 < str2 Positive if str1 > str2 Zero if equal

#### 5.2.5.30 OSCL\_IMPORT\_REF oscl\_wchar\* oscl\_strncpy (oscl\_wchar \* *dest*, const oscl\_wchar \* *src*, uint32 *count*)

Copies the chars of one string to another.

Copies the initial count characters of src to dest and returns dest. If count is less than or equal to the length of src, a null character is not appended automatically to the copied string. If count is greater than the length of src, the destination string is padded with null characters up to length count. The behavior of strncpy is undefined if the source and destination strings overlap.

**Parameters:**

***dest*** Destination string

***src*** NULL terminated source string

***count*** Number of chars to copy

**Returns:**

Returns dest.

#### 5.2.5.31 OSCL\_IMPORT\_REF char\* oscl\_strncpy (char \* *dest*, const char \* *src*, uint32 *count*)

Copies the chars of one string to another.

Copies the initial count characters of src to dest and returns dest. If count is less than or equal to the length of src, a null character is not appended automatically to the copied string. If count is greater than the length of src, the destination string is padded with null characters up to length count. The behavior of strncpy is undefined if the source and destination strings overlap.

**Parameters:**

***dest*** Destination string

*src* NULL terminated source string

*count* Number of chars to copy

**Returns:**

Returns dest.

**5.2.5.32 OSCL\_IMPORT\_REF oscl\_wchar\* oscl\_strrchr (oscl\_wchar \*str, int32 c)**

**5.2.5.33 OSCL\_IMPORT\_REF const oscl\_wchar\* oscl\_strrchr (const oscl\_wchar \*str, int32 c)**

**5.2.5.34 OSCL\_IMPORT\_REF char\* oscl\_strrchr (char \*str, int32 c)**

**5.2.5.35 OSCL\_IMPORT\_REF const char\* oscl\_strrchr (const char \*str, int32 c)**

Finds the last occurrence of *c* in string, or it returns NULL if *c* is not found. The null-terminating character is included in the search.

**Parameters:**

*str* null terminated source string

*c* character to search for

**Returns:**

**5.2.5.36 OSCL\_IMPORT\_REF oscl\_wchar\* oscl\_strset (oscl\_wchar \*dest, oscl\_wchar val, uint32 count)**

Sets the characters of a string to a specified character

**Parameters:**

*dest* buffer to modify

*val* character to set

*count* number of chars to set

**Returns:**

the value of dest

**5.2.5.37 OSCL\_IMPORT\_REF char\* oscl\_strset (char \*dest, char val, uint32 count)**

Sets the characters of a string to a specified character

**Parameters:**

*dest* buffer to modify

*val* character to set

*count* number of chars to set

**Returns:**

the value of dest

**5.2.5.38 OSCL\_IMPORT\_REF oscl\_wchar\* oscl\_strstr (oscl\_wchar \* str1, const oscl\_wchar \* str2)**

**5.2.5.39 OSCL\_IMPORT\_REF const oscl\_wchar\* oscl\_strstr (const oscl\_wchar \* str1, const oscl\_wchar \* str2)**

find the occurrence of sub-string in a string.

**Parameters:**

*str1* string.

*str2* sub-string

**Returns:**

pointer to the begining of sub-string.

**5.2.5.40 OSCL\_IMPORT\_REF char\* oscl\_strstr (char \* str1, const char \* str2)**

**5.2.5.41 OSCL\_IMPORT\_REF const char\* oscl\_strstr (const char \* str1, const char \* str2)**

find the occurrence of sub-string in a string.

**Parameters:**

*str1* string.

*str2* sub-string

**Returns:**

pointer to the begining of sub-string.

**5.2.5.42 OSCL\_IMPORT\_REF oscl\_wchar oscl\_tolower (const oscl\_wchar car)**

convert upper case ASCII character to lower case. behaviour of this function for non-ASCII characters is not defined.

**Parameters:**

*car* upper case character.

**Returns:**

lower case character.

**5.2.5.43 OSCL\_IMPORT\_REF char oscl\_tolower (const char car)**

convert upper case ASCII character to lower case. behaviour of this function for non-ASCII characters is not defined.

**Parameters:**

*car* upper case character.

**Returns:**

lower case character.

**5.2.5.44 OSCL\_IMPORT\_REF void PV8601ToRFC822 (**PV8601timeStrBuf** *pv8601\_buffer*,  
**CtimeStrBuf** *ctime\_buffer*)**

**5.2.5.45 void PVOsclBase\_Cleanup ()**

Cleanup OsclBase functionality OsclBase should be cleaned once OsclBase functions are no longer needed

**5.2.5.46 void PVOsclBase\_Init ()**

Initializes OsclBase functionality. OsclBase must be initialized before any OsclBase functionality can be used.

**Exceptions:**

*leaves* if out-of-memory

5.2.5.47 OSCL\_IMPORT\_REF void RFC822ToPV8601 (**CtimeStrBuf** *ctime\_buffer*,  
**PV8601timeStrBuf**)

## 5.2.6 Variable Documentation

5.2.6.1 const int **CTIME\_BUFFER\_SIZE** = 26

5.2.6.2 const long **MSEC\_PER\_SEC** = 1000

5.2.6.3 const uint32 **OSCL\_TLS\_ID\_BASE\_LAST** = 11

5.2.6.4 const uint32 **OSCL\_TLS\_ID\_ERRORHOOK** = 1

5.2.6.5 const uint32 **OSCL\_TLS\_ID\_MAGICNUM** = 0

5.2.6.6 const uint32 **OSCL\_TLS\_ID\_OSCLREGISTRY** = 10

5.2.6.7 const uint32 **OSCL\_TLS\_ID\_PAYLOADPARSER** = 7

5.2.6.8 const uint32 **OSCL\_TLS\_ID\_PVERRORTRAP** = 5

5.2.6.9 const uint32 **OSCL\_TLS\_ID\_PVLOGGER** = 2

5.2.6.10 const uint32 **OSCL\_TLS\_ID\_PVMFRECOGNIZER** = 8

5.2.6.11 const uint32 **OSCL\_TLS\_ID\_PVSCHEDULER** = 4

5.2.6.12 const uint32 **OSCL\_TLS\_ID\_SDPMEDIAPARSER** = 6

5.2.6.13 const uint32 **OSCL\_TLS\_ID\_SQLITE3** = 11

5.2.6.14 const uint32 **OSCL\_TLS\_ID\_TEST** = 3

5.2.6.15 const uint32 **OSCL\_TLS\_ID\_WMDRM** = 9

5.2.6.16 const int **PV8601TIME\_BUFFER\_SIZE** = 21

5.2.6.17 const uint32 **unix\_ntp\_offset** = 2208988800U

5.2.6.18 const long **USEC\_PER\_SEC** = 1000000

## 5.3 OSCL Memory

### Files

- file [oscl\\_mem.h](#)

*This file contains basic memory definitions for common use across platforms.*

- file [oscl\\_mem\\_audit.h](#)

*This file contains the definition and partial implementation of MM\_Audit class.*

- file [oscl\\_mem\\_audit\\_internals.h](#)

*This file contains the internal definitions for the mem audit library.*

- file [oscl\\_mem\\_auto\\_ptr.h](#)

*This file defines the oscl\_mem\_auto\_ptr template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error.*

- file [oscl\\_mem\\_basic\\_functions.h](#)

*This file contains prototypes for the basic memory functions.*

- file [oscl\\_mem\\_mempool.h](#)

*This file contains the definition of memory pool allocators.*

### Data Structures

- class [allocator](#)
- class [allocator](#)
- class [HeapBase](#)
- struct [MM\\_AllocBlockFence](#)
- struct [MM\\_AllocBlockHdr](#)
- struct [MM\\_AllocInfo](#)
- struct [MM\\_AllocNode](#)
- struct [MM\\_AllocQueryInfo](#)
- class [MM\\_Audit\\_Imp](#)
- struct [MM\\_AuditOverheadStats](#)
- struct [MM\\_FailInsertParam](#)
- struct [MM\\_Stats\\_CB](#)
- struct [MM\\_Stats\\_t](#)
- class [OsclAuditCB](#)
- class [OsclMem](#)
- class [OsclMemAllocator](#)
- class [OsclMemAllocator](#)
- class [OsclMemAllocDestructDealloc](#)
- class [OsclMemAllocDestructDealloc](#)
- class [OsclMemAudit](#)
- class [OSCLMemAutoPtr](#)

*The oscl\_auto\_ptr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the oscl\_auto\_ptr expires, its destructor uses delete to free the memory.*

- class OsclMemBasicAllocator
- class OsclMemBasicAllocator
- class OsclMemBasicAllocDestructDealloc
- class OsclMemBasicAllocDestructDealloc
- class OsclMemGlobalAuditObject
- class OsclMemPoolFixedChunkAllocator
- class OsclMemPoolFixedChunkAllocatorObserver
- class OsclMemPoolResizableAllocator
- class OsclMemPoolResizableAllocatorMemoryObserver
- class OsclMemPoolResizableAllocatorObserver
- class OsclMemStatsNode

## Defines

- #define OSCL\_DISABLE\_WARNING\_TRUNCATE\_DEBUG\_MESSAGE
- #define OSCL\_HAS\_GLOBAL\_NEW\_DELETE 1
- #define OSCL\_CLEANUP\_BASE\_CLASS(T) \_OSCL\_CLEANUP\_BASE\_CLASS(T)
- #define OSCL\_ALLOC\_NEW(T\_allocator, T, params) new(T\_allocator.allocate(1)) T params
- #define OSCL\_TRAP\_ALLOC\_NEW(T\_ptr, T\_allocator, T, params) \_OSCL\_TRAP\_NEW(T\_allocator.allocate(1),T\_allocator.deallocate,T\_ptr,T,params)
- #define OSCL\_ALLOC\_DELETE(ptr, T\_allocator, T)
- #define OSCL\_MALLOC(count) \_oscl\_default\_audit\_malloc(count)
- #define oscl\_malloc(a) OSCL\_MALLOC(a)
- #define OSCL\_DEFAULT\_MALLOC(x) OSCL\_MALLOC(x)
- #define OSCL\_AUDIT\_MALLOC(auditCB, count) \_oscl\_audit\_malloc(count, auditCB)
- #define OSCL\_CALLOC(num, size) \_oscl\_default\_audit\_calloc(num,size)
- #define oscl\_calloc(a, b) OSCL\_CALLOC(a,b)
- #define OSCL\_AUDIT\_CALLOC(auditCB, num, size) \_oscl\_audit\_calloc(num,size, auditCB)
- #define OSCL\_REALLOC(ptr, new\_size) \_oscl\_default\_audit\_realloc(ptr,new\_size)
- #define oscl\_realloc(a, b) OSCL\_REALLOC(a,b)
- #define OSCL\_AUDIT\_REALLOC(auditCB, ptr, new\_size) \_oscl\_audit\_realloc(ptr,new\_size, auditCB)
- #define OSCL\_FREE(ptr) \_oscl\_audit\_free(ptr)
- #define oscl\_free(x) OSCL\_FREE(x)
- #define OSCL\_DEFAULT\_FREE(x) OSCL\_FREE(x)
- #define OSCL\_NEW(T, params) new T params
- #define OSCL\_PLACEMENT\_NEW(ptr, constructor) new(ptr) constructor
- #define OSCL\_TRAP\_NEW(T\_ptr, T, params) \_OSCL\_TRAP\_NEW(\_oscl\_default\_audit\_new(sizeof(T)),\_oscl\_audit\_free,T\_ptr,T,params)
- #define OSCL\_AUDIT\_NEW(auditCB, T, params) new(\_oscl\_audit\_new(sizeof(T),auditCB)) T params
- #define OSCL\_TRAP\_AUDIT\_NEW(T\_ptr, auditCB, T, params) \_OSCL\_TRAP\_NEW(\_oscl\_audit\_new(sizeof(T),auditCB),\_oscl\_audit\_free,T\_ptr,T,params)
- #define OSCL\_DELETE(ptr)
- #define OSCL\_AUDIT\_ARRAY\_NEW(auditCB, T, count) new(\_oscl\_audit\_new(sizeof(T)\*(count),auditCB)) T
- #define OSCL\_ARRAY\_NEW(T, count) new T[count]
- #define OSCL\_ARRAY\_DELETE(ptr) delete [ ] ptr
- #define OSCL\_TRAP\_NEW(exp, freeFunc, T\_ptr, T, params)

- #define **\_OSCL\_CLEANUP\_BASE\_CLASS**(T) this → T::~T()
- #define **MM\_ALLOC\_MAX\_QUERY\_FILENAME\_LEN** 128
- #define **MM\_ALLOC\_MAX\_QUERY\_TAG\_LEN** 64
- #define **MM\_AUDIT\_VALIDATE\_BLOCK** 1
- #define **MM\_AUDIT\_PREFILL\_FLAG** 0x1
- #define **MM\_AUDIT\_POSTFILL\_FLAG** 0x2
- #define **MM\_AUDIT\_VALIDATE\_ALL\_HEAP\_FLAG** 0x4
- #define **MM\_AUDIT\_VALIDATE\_ON\_FREE\_FLAG** 0x8
- #define **MM\_AUDIT\_ALLOC\_NODE\_ENABLE\_FLAG** 0x10
- #define **MM\_AUDIT\_SUPPRESS\_FILENAME\_FLAG** 0x20
- #define **DEFAULT\_MM\_AUDIT\_MODE** 0
- #define **MM\_AUDIT\_ALLOC\_NODE\_SUPPORT** 1
- #define **MM\_AUDIT\_FENCE\_SUPPORT** 0
- #define **MM\_AUDIT\_INCLUDE\_ALL\_HEAP\_VALIDATION** 1
- #define **MM\_AUDIT\_FILL\_SUPPORT** 0
- #define **MM\_AUDIT\_FAILURE\_SIMULATION\_SUPPORT** 1
- #define **FENCE\_PATTERN** 0xAA
- #define **MIN\_FENCE\_SIZE** 4
- #define **MEM\_ALIGN\_SIZE** 8
- #define **COMPUTE\_MEM\_ALIGN\_SIZE**(x, y, z) (y+((x+y)%z) ? (z - (x+y)%z) : 0))
- #define **DEFAULT\_PREFILL\_PATTERN** 0x96
- #define **DEFAULT\_POSTFILL\_PATTERN** 0x5A
- #define **OSCL\_DISABLE\_WARNING\_RETURN\_TYPE\_NOT\_UDT**

## Typedefs

- typedef **OSCLMemAutoPtr< char, Oscl\_TAlloc< char, OsclMemBasicAllocator > >** **MMAudit\_CharAutoPtr**
- typedef **OSCLMemAutoPtr< uint8, Oscl\_TAlloc< uint8, \_OsclBasicAllocator > >** **MMAudit\_Uint8AutoPtr**
- typedef **OSCLMemAutoPtr< MM\_AllocNode, Oscl\_TAlloc< MM\_AllocNode, OsclMemBasicAllocator > >** **MM\_AllocNodeAutoPtr**
- typedef **OSCLMemAutoPtr< OsclMemStatsNode, Oscl\_TAlloc< OsclMemStatsNode, OsclMemBasicAllocator > >** **MM\_StatsNodeTagTreeType**
- typedef **OSCLMemAutoPtr< OsclMemStatsNode, Oscl\_TAlloc< OsclMemStatsNode, OsclMemBasicAllocator > >** **OsclMemStatsNodeAutoPtr**
- typedef **Oscl\_TAlloc< MM\_StatsNodeTagTreeType, OsclMemBasicAllocator >** **TagTreeAllocator**
- typedef **Oscl\_TagTree< MM\_StatsNodeTagTreeType, TagTreeAllocator >** **OsclTagTreeType**

## Functions

- **OSCL\_COND\_IMPORT\_REF void \* \_oscl\_malloc** (int32 count)
- **OSCL\_COND\_IMPORT\_REF void \* \_oscl\_calloc** (int32 nelems, int32 size)
- **OSCL\_COND\_IMPORT\_REF void \* \_oscl\_realloc** (void \*src, int32 count)
- **OSCL\_COND\_IMPORT\_REF void \_oscl\_free** (void \*src)
- **OSCL\_COND\_IMPORT\_REF void \* oscl\_memcpy** (void \*dest, const void \*src, uint32 count)
- **OSCL\_COND\_IMPORT\_REF void \* oscl\_memmove** (void \*dest, const void \*src, uint32 count)
- **OSCL\_COND\_IMPORT\_REF void \* oscl\_memmove32** (void \*dest, const void \*src, uint32 count)
- **OSCL\_COND\_IMPORT\_REF void \* oscl\_memset** (void \*dest, uint8 val, uint32 count)

- OSCL\_COND\_IMPORT\_REF int `oscl_memcmp` (const void \*buf1, const void \*buf2, uint32 count)
- OSCL\_COND\_IMPORT\_REF uint `oscl_mem_aligned_size` (uint size)
- OSCL\_IMPORT\_REF void `OsclMemInit` (OsclAuditCB &auditCB)
- OSCL\_IMPORT\_REF void \* `_oscl_audit_malloc` (size\_t, OsclAuditCB &, const char \*f=NULL, const int l=0)
- OSCL\_IMPORT\_REF void \* `_oscl_audit_calloc` (size\_t, size\_t, OsclAuditCB &, const char \*f=NULL, const int l=0)
- OSCL\_IMPORT\_REF void \* `_oscl_audit_realloc` (void \*, size\_t, OsclAuditCB &, const char \*f=NULL, const int l=0)
- OSCL\_IMPORT\_REF void \* `_oscl_audit_new` (size\_t, OsclAuditCB &, const char \*f=NULL, const int l=0)
- OSCL\_IMPORT\_REF void \* `_oscl_default_audit_malloc` (size\_t, const char \*f=NULL, const int l=0)
- OSCL\_IMPORT\_REF void \* `_oscl_default_audit_calloc` (size\_t, size\_t, const char \*f=NULL, const int l=0)
- OSCL\_IMPORT\_REF void \* `_oscl_default_audit_realloc` (void \*, size\_t, const char \*f=NULL, const int l=0)
- OSCL\_IMPORT\_REF void \* `_oscl_default_audit_new` (size\_t, const char \*f=NULL, const int l=0)
- OSCL\_IMPORT\_REF void `_oscl_audit_free` (void \*)
- void \* `operator new` (size\_t aSize, const char \*aFile, int aLine)
- void \* `operator new` (size\_t)
- void `operator delete` (void \*)
- void \* `operator new[]` (size\_t aSize, const char \*aFile, int aLine)
- void \* `operator new[]` (size\_t aSize)
- void `operator delete[]` (void \*aPtr)

## Variables

- const uint32 `ALLOC_NODE_FLAG` = 0x80000000

### 5.3.1 Define Documentation

#### 5.3.1.1 #define \_OSCL\_CLEANUP\_BASE\_CLASS(T) this → T::~T()

This macro is used to cleanup the base class in a derived-class constructor just before a leave occurs.

**Parameters:**

*T*: base class name.

#### 5.3.1.2 #define \_OSCL\_TRAP\_NEW(exp, freeFunc, T\_ptr, T, params)

**Value:**

```
{
    int32 __err; \
    OsclAny* __ptr=exp; \
    OSCL_TRY(__err,T_ptr=new(__ptr) T params); \
    if(__err){ \
        freeFunc(__ptr); \
        T_ptr=NULL; \
        OsclError::Leave(__err); \
    } \
}
```

Internal-use macro to catch leaves in constructors. If the constructor leaves, this will free the memory before allowing the leave to propagate to the next level. It is the constructor's responsibility to cleanup any memory in the partially constructed object before leaving. This cleanup may include cleaning up the base class using the OSCL\_CLEANUP\_BASE\_CLASS macro.

**Parameters:**

*exp*: expression to allocate memory.

*Tptr:variable* to hold result.

*T*: type

*params*: constructor arg list

*freeFunc*: delete or free function.

- 5.3.1.3 #define COMPUTE\_MEM\_ALIGN\_SIZE(x, y, z) (y+((x+y)%z) ? (z - (x+y)%z) : 0))
- 5.3.1.4 #define DEFAULT\_MM\_AUDIT\_MODE 0
- 5.3.1.5 #define DEFAULT\_POSTFILL\_PATTERN 0x5A
- 5.3.1.6 #define DEFAULT\_PREFILL\_PATTERN 0x96
- 5.3.1.7 #define FENCE\_PATTERN 0xAA
- 5.3.1.8 #define MEM\_ALIGN\_SIZE 8
- 5.3.1.9 #define MIN\_FENCE\_SIZE 4
- 5.3.1.10 #define MM\_ALLOC\_MAX\_QUERY\_FILENAME\_LEN 128
- 5.3.1.11 #define MM\_ALLOC\_MAX\_QUERY\_TAG\_LEN 64
- 5.3.1.12 #define MM\_AUDIT\_ALLOC\_NODE\_ENABLE\_FLAG 0x10
- 5.3.1.13 #define MM\_AUDIT\_ALLOC\_NODE\_SUPPORT 1
- 5.3.1.14 #define MM\_AUDIT\_FAILURE\_SIMULATION\_SUPPORT 1
- 5.3.1.15 #define MM\_AUDIT\_FENCE\_SUPPORT 0
- 5.3.1.16 #define MM\_AUDIT\_FILL\_SUPPORT 0
- 5.3.1.17 #define MM\_AUDIT\_INCLUDE\_ALL\_HEAP\_VALIDATION 1
- 5.3.1.18 #define MM\_AUDIT\_POSTFILL\_FLAG 0x2
- 5.3.1.19 #define MM\_AUDIT\_PREFILL\_FLAG 0x1
- 5.3.1.20 #define MM\_AUDIT\_SUPPRESS\_FILENAME\_FLAG 0x20
- 5.3.1.21 #define MM\_AUDIT\_VALIDATE\_ALL\_HEAP\_FLAG 0x4
- 5.3.1.22 #define MM\_AUDIT\_VALIDATE\_BLOCK 1
- 5.3.1.23 #define MM\_AUDIT\_VALIDATE\_ON\_FREE\_FLAG 0x8
- 5.3.1.24 #define OSCL\_ALLOC\_DELETE(ptr, T\_allocator, T)

**Value:**

```
{\
  ptr->~T();\
  T_allocator.deallocate(ptr);\
}
```

Deletes the object of type T using the given allocator

**Parameters:**

*T\_allocator* allocator for objects of type T

*T* type of object to delete

*ptr* pointer to previously created object

**Exceptions:**

*none* , unless thrown by the given allocator

**5.3.1.25 #define OSCL\_ALLOC\_NEW(T\_allocator, T, params) new(T\_allocator.allocate(1)) T  
params**

Creates an object of type T using the given allocator to acquire the memory needed.

**Parameters:**

*T\_allocator* allocator for objects of type T, must be an [Oscl\\_TAlloc<T, Allocator>](#), where Allocator is an [Oscl\\_DefAlloc](#)

*T* type of object to create

*params* object initialization parameters

**Returns:**

pointer to created object

**Exceptions:**

*none* , unless thrown by the given allocator

**5.3.1.26 #define OSCL\_ARRAY\_DELETE(ptr) delete [] ptr**

Oscl array delete operator..

**Parameters:**

*ptr* pointer to memory block previously allocated with OSCL\_ARRAY\_NEW

**Returns:**

void

**5.3.1.27 #define OSCL\_ARRAY\_NEW(T, count) new T[count]**

Oscl array "new" operator. This uses the global memory audit object.

**Parameters:**

*T* data type for 'new' operation

*count* number of elements to create

**Returns:**

pointer to the newly created object array of type T

**Exceptions:**

*may* leave with code = bad alloc

**5.3.1.28 #define OSCL\_AUDIT\_ARRAY\_NEW(auditCB, T, count)**  
`new(_oscl_audit_new(sizeof(T)*(count),auditCB)) T`

Oscl array "new" operator. This uses the input memory audit object.

**Parameters:**

*auditCB* input memory management audit object  
*T* data type for 'new' operation  
*count* number of elements to create

**Returns:**

pointer to the newly created object array of type T

**Exceptions:**

*may* leave with code = bad alloc

**5.3.1.29 #define OSCL\_AUDIT\_CALLOC(auditCB, num, size) \_oscl\_audit\_calloc(num,size, auditCB)**

Allocates a memory block using the specified audit object. The block is initialized to zero.

**Parameters:**

*auditCB* input memory management audit object  
*num* number of elements  
*size* number of bytes to allocate for each element

**Returns:**

a void pointer to the allocated space, or NULL if there is insufficient memory available.

**Exceptions:**

*none*

**5.3.1.30 #define OSCL\_AUDIT\_MALLOC(auditCB, count) \_oscl\_audit\_malloc(count, auditCB)**

Allocates a memory block using the given audit object.

**Parameters:**

*auditCB* input memory management audit object  
*count* number of bytes to allocate

**Returns:**

a void pointer to the allocated space, or NULL if there is insufficient memory available.

**Exceptions:**

*none*

**5.3.1.31 #define OSCL\_AUDIT\_NEW(auditCB, T, params) new(\_oscl\_audit\_new(sizeof(T),audit-CB)) T params**

Oscl "new" operator. This uses the specified memory audit object.

**Parameters:**

*auditCB* input memory management audit object  
*T* data type for 'new' operation  
*params* object initialization parameters

**Returns:**

pointer to the newly created object of type *T*

**Exceptions:**

*may* leave with code = bad alloc

**5.3.1.32 #define OSCL\_AUDIT\_REALLOC(auditCB, ptr, new\_size)  
\_oscl\_audit\_realloc(ptr,new\_size,auditCB)**

Re-Allocates a memory block using the specified audit object.

**Parameters:**

*auditCB* input memory management audit object  
*ptr* original memory block  
*new\_size* New size of the block

**Returns:**

a void pointer to the allocated space, or NULL if there is insufficient memory available.

**Exceptions:**

*none*

**5.3.1.33 #define oscl\_calloc(a, b) OSCL\_CALLOC(a,b)****5.3.1.34 #define OSCL\_CALLOC(num, size) \_oscl\_default\_audit\_calloc(num,size)**

Allocates a memory block using the memory management's global audit object. The block is initialized to zero.

**Parameters:**

*num* number of elements  
*size* number of bytes to allocate for each element

**Returns:**

a void pointer to the allocated space, or NULL if there is insufficient memory available.

**Exceptions:**

*none*

**5.3.1.35 #define OSCL\_CLEANUP\_BASE\_CLASS(T) \_OSCL\_CLEANUP\_BASE\_CLASS(T)**

Cleans up the base class of a partially-constructed derived class. This macro will call the destructor if necessary, based on the error-handling implementation.

**Parameters:**

*T*: name of the base class.

**5.3.1.36 #define OSCL\_DEFAULT\_FREE(x) OSCL\_FREE(x)**

Another back-compatibility definition.

**5.3.1.37 #define OSCL\_DEFAULT\_MALLOC(x) OSCL\_MALLOC(x)**

Another back-compatibility definition.

**5.3.1.38 #define OSCL\_DELETE(ptr)****Value:**

```
{ \
    if(ptr){delete(ptr);} \
}
```

Oscl "delete" operator.

**Parameters:**

*ptr* pointer to memory block previously allocated with OSCL\_NEW

**Returns:**

void

**5.3.1.39 #define OSCL\_DISABLE\_WARNING\_RETURN\_TYPE\_NOT\_UDT****5.3.1.40 #define OSCL\_DISABLE\_WARNING\_TRUNCATE\_DEBUG\_MESSAGE**

Previously this was in oscl\_mem\_imp.h

**5.3.1.41 #define oscl\_free(x) OSCL\_FREE(x)****5.3.1.42 #define OSCL\_FREE(ptr) \_oscl\_audit\_free(ptr)**

Deallocates or frees a memory block.

**Parameters:**

*ptr* pointer to previously allocated memory block using the given audit object

**5.3.1.43 #define OSCL\_HAS\_GLOBAL\_NEW\_DELETE 1**

**5.3.1.44 #define oscl\_malloc(a) OSCL\_MALLOC(a)**

**5.3.1.45 #define OSCL\_MALLOC(count) \_oscl\_default\_audit\_malloc(count)**

Allocates a memory block using the memory management's global audit object.

**Parameters:**

*count* number of bytes to allocate

**Returns:**

a void pointer to the allocated space, or NULL if there is insufficient memory available.

**Exceptions:**

*none*

**5.3.1.46 #define OSCL\_NEW(T, params) new T params**

Oscl "new" operator. This uses the global memory audit object.

**Parameters:**

*T* data type for 'new' operation

*params* object initialization parameters

**Returns:**

pointer to the newly created object of type T

**Exceptions:**

*may* leave with code = bad alloc

**5.3.1.47 #define OSCL\_PLACEMENT\_NEW(ptr, constructor) new(ptr) constructor**

**5.3.1.48 #define oscl\_realloc(a, b) OSCL\_REALLOC(a,b)**

**5.3.1.49 #define OSCL\_REALLOC(ptr, new\_size) \_oscl\_default\_audit\_realloc(ptr,new\_size)**

Re-Allocates a memory block using the memory management's global audit object.

**Parameters:**

*ptr* original memory block

*new\_size* New size of the block

**Returns:**

a void pointer to the allocated space, or NULL if there is insufficient memory available.

**Exceptions:**

*none*

**5.3.1.50 #define OSCL\_TRAP\_ALLOC\_NEW(T\_ptr, T\_allocator, T, params)**  
**\_OSCL\_TRAP\_NEW(T\_allocator.allocate(1),T\_allocator.deallocate,T\_ptr,T,params)**

Creates an object of type T using the given allocator to acquire the memory needed. This macro is similar to OSCL\_ALLOC\_NEW except that it handles constructors that leave. If the constructor leaves, the destructor will be called, and allocated memory will be freed before allowing the leave to propagate to the next level.

**Parameters:**

*T\_ptr* variable to hold return value— pointer to new object of type T.

*T\_allocator* allocator for objects of type T, must be an [Oscl\\_TAlloc<T, Allocator>](#), where Allocator is an [Oscl\\_DefAlloc](#)

*T* type of object to create

*params* object initialization parameters

**Returns:**

pointer to created object

**Exceptions:**

*none* , unless thrown by the given allocator

**5.3.1.51 #define OSCL\_TRAP\_AUDIT\_NEW(T\_ptr, auditCB, T, params) \_OSCL\_TRAP\_-  
 \_NEW(\_oscl\_audit\_new(sizeof(T),auditCB),\_oscl\_audit\_free,T\_ptr,T,params)**

Oscl "new" operator. This uses the specified memory audit object. This macro is similar to OSCL\_AUDIT\_NEW except that it will handle constructors that leave. If the constructor leaves, the destructor will be called, and allocated memory will be freed before allowing the leave to propagate to the next level.

**Parameters:**

*T\_ptr* variable to hold return value— pointer to new object of type T.

*auditCB* input memory management audit object

*T* data type for 'new' operation

*params* object initialization parameters

**Returns:**

pointer to the newly created object of type T

**Exceptions:**

*may* leave with code = bad alloc

**5.3.1.52 #define OSCL\_TRAP\_NEW(T\_ptr, T, params) \_OSCL\_TRAP\_NEW(\_oscl\_default\_-  
 audit\_new(sizeof(T)),\_oscl\_audit\_free,T\_ptr,T,params)**

Oscl "new" operator. This uses the global memory audit object. This operator is similar to OSCL\_NEW except that it will handle constructors that leave. If the constructor leaves, the destructor will be called, and allocated memory will be freed before allowing the leave to propagate to the next level.

**Parameters:**

*T\_ptr* variable to hold return value— pointer to new object of type T.

*T* data type for 'new' operation

*params* object initialization parameters

**Returns:**

pointer to the newly created object of type T

**Exceptions:**

*may* leave with code = bad alloc

### 5.3.2 Typedef Documentation

5.3.2.1 `typedef OSCLMemAutoPtr<MM_AllocNode, Oscl_TAlloc<MM_AllocNode, OsclMemBasicAllocator> > MM_AllocNodeAutoPtr`

5.3.2.2 `typedef OSCLMemAutoPtr<OsclMemStatsNode, Oscl_TAlloc<OsclMemStatsNode, OsclMemBasicAllocator> > MM_StatsNodeTagTreeType`

5.3.2.3 `typedef OSCLMemAutoPtr<char, Oscl_TAlloc<char, OsclMemBasicAllocator> > MMAuditCharAutoPtr`

5.3.2.4 `typedef OSCLMemAutoPtr<uint8, Oscl_TAlloc<uint8, _OsclBasicAllocator> > MMAuditUInt8AutoPtr`

5.3.2.5 `typedef OSCLMemAutoPtr<OsclMemStatsNode, Oscl_TAlloc<OsclMemStatsNode, OsclMemBasicAllocator> > OsclMemStatsNodeAutoPtr`

5.3.2.6 `typedef Oscl_TagTree<MM_StatsNodeTagTreeType, TagTree_Allocator> OsclTagTreeType`

5.3.2.7 `typedef Oscl_TAlloc<MM_StatsNodeTagTreeType, OsclMemBasicAllocator> TagTree_Allocator`

### 5.3.3 Function Documentation

5.3.3.1 `OSCL_IMPORT_REF void* _oscl_audit_calloc (size_t, size_t, OsclAuditCB &, const char *f = NULL, const int l = 0)`

5.3.3.2 `OSCL_IMPORT_REF void _oscl_audit_free (void *)`

5.3.3.3 `OSCL_IMPORT_REF void* _oscl_audit_malloc (size_t, OsclAuditCB &, const char *f = NULL, const int l = 0)`

\*\*\*\*\* Macros for malloc/free with memory management.

**5.3.3.4 OSCL\_IMPORT\_REF void\* \_oscl\_audit\_new (size\_t, OsclAuditCB &, const char \**f*=NULL, const int *l*=0)**

**5.3.3.5 OSCL\_IMPORT\_REF void\* \_oscl\_audit\_realloc (void \*, size\_t, OsclAuditCB &, const char \**f*=NULL, const int *l*=0)**

**5.3.3.6 OSCL\_COND\_IMPORT\_REF void\* \_oscl\_calloc (int32 *nelems*, int32 *size*)**

**5.3.3.7 OSCL\_IMPORT\_REF void\* \_oscl\_default\_audit\_calloc (size\_t, size\_t, const char \**f*=NULL, const int *l*=0)**

**5.3.3.8 OSCL\_IMPORT\_REF void\* \_oscl\_default\_audit\_malloc (size\_t, const char \**f*=NULL, const int *l*=0)**

**5.3.3.9 OSCL\_IMPORT\_REF void\* \_oscl\_default\_audit\_new (size\_t, const char \**f*=NULL, const int *l*=0)**

**5.3.3.10 OSCL\_IMPORT\_REF void\* \_oscl\_default\_audit\_realloc (void \*, size\_t, const char \**f*=NULL, const int *l*=0)**

**5.3.3.11 OSCL\_COND\_IMPORT\_REF void \_oscl\_free (void \**src*)**

**5.3.3.12 OSCL\_COND\_IMPORT\_REF void\* \_oscl\_malloc (int32 *count*)**

**5.3.3.13 OSCL\_COND\_IMPORT\_REF void\* \_oscl\_realloc (void \**src*, int32 *count*)**

**5.3.3.14 void operator delete (void \*) [inline]**

**5.3.3.15 ]**

void operator delete[] (void \**aPtr*) [inline]

**5.3.3.16 void\* operator new (size\_t) [inline]**

**5.3.3.17 void\* operator new (size\_t *aSize*, const char \**aFile*, int *aLine*) [inline]**

**5.3.3.18 ]**

void\* operator new[] (size\_t *aSize*) [inline]

**5.3.3.19 ]**

void\* operator new[] (size\_t *aSize*, const char \**aFile*, int *aLine*) [inline]

**5.3.3.20 OSCL\_COND\_IMPORT\_REF uint oscl\_mem\_aligned\_size (uint *size*)**

Get memory-aligned size of an object.

**Parameters:**

*size* size of object

**Returns:**

memory-aligned size

**5.3.3.21 OSCL\_COND\_IMPORT\_REF int oscl\_memcmp (const void \* buf1, const void \* buf2, uint32 count)**

Compare characters in two buffers

**Parameters:**

*buf1* first buffer

*buf2* second buffer

*count* number of bytes to compare

**Returns:**

<0 buf1 less than buf2 0 buf1 equal to buf2 >0 buf1 greater than buf2

**5.3.3.22 OSCL\_COND\_IMPORT\_REF void\* oscl\_memcpy (void \* dest, const void \* src, uint32 count)**

Copies characters between buffers The oscl\_memcpy function copies count bytes of src to dest. If the source and destination overlap, this function does not ensure that the original source bytes in the overlapping region are copied before being overwritten. Use oscl\_memmove to handle overlapping regions

**Parameters:**

*dest* new buffer

*src* buffer to copy

*count* number of bytes to copy

**Returns:**

the value of dest

**5.3.3.23 OSCL\_COND\_IMPORT\_REF void\* oscl\_memmove (void \* dest, const void \* src, uint32 count)**

Moves chars from one buffer to another The memmove function copies count bytes of characters from src to dest. If some regions of the source area and the destination overlap, memmove ensures that the original source bytes in the overlapping region are copied before being overwritten.

**Parameters:**

*dest* new buffer

*src* buffer to copy

*count* number of bytes to copy

**Returns:**

the value of dest

**5.3.3.24 OSCL\_COND\_IMPORT\_REF void\* oscl\_memmove32 (void \* dest, const void \* src, uint32 count)**

Same functionality as oscl\_memmove, yet optimized for memory aligned on 32-bit boundary

**Parameters:**

*dest* new buffer  
*src* buffer to copy  
*count* number of bytes to copy

**Returns:**

the value of dest

**5.3.3.25 OSCL\_COND\_IMPORT\_REF void\* oscl\_memset (void \* dest, uint8 val, uint32 count)**

Sets the bytes of a buffer to a specified character

**Parameters:**

*dest* buffer to modify  
*val* character to set  
*count* number of bytes to set

**Returns:**

the value of dest

**5.3.3.26 OSCL\_IMPORT\_REF void OsclMemInit ([OsclAuditCB](#) & auditCB)**

Initialize an [OsclAuditCB](#) object. Sets the stats node pointer to null, and sets the audit pointer to the global audit object.

**Parameters:**

*auditCB* memory management audit object

## 5.3.4 Variable Documentation

**5.3.4.1 const uint32 MM\_AllocBlockHdr::ALLOC\_NODE\_FLAG = 0x80000000 [static, inherited]**

## 5.4 OSCL Util

### Files

- file [oscl\\_bin\\_stream.h](#)  
*Defines a set of binary stream classes which handle portable input / output of binary data regardless of the native byte order.*
- file [oscl\\_math.h](#)  
*Provides math functions.*
- file [oscl\\_media\\_data.h](#)  
*Defines a container class for media data made up of a collection of memory fragments.*
- file [oscl\\_media\\_status.h](#)  
*Defines a status values for the [MediaData](#) containers.*
- file [oscl\\_pqueue.h](#)  
*Implements a priority queue data structure similar to STL.*
- file [oscl\\_rand.h](#)  
*Provides pseudo-random number generation.*
- file [oscl\\_registry\\_access\\_client.h](#)  
*Client-side implementation Registry Access implementation.*
- file [oscl\\_registry\\_client.h](#)  
*Client-side implementation of OsclRegistry.*
- file [oscl\\_registry\\_client\\_impl.h](#)  
*Client-side implementation of OsclRegistryInterface.*
- file [oscl\\_registry\\_serv\\_impl.h](#)  
*Server-side implementation of OsclRegistry interfaces.*
- file [oscl\\_registry\\_types.h](#)  
*Common types used in Oscl registry interfaces.*
- file [oscl\\_snprintf.h](#)  
*Provides a portable implementation of snprintf.*
- file [oscl\\_str\\_ptr\\_len.h](#)  
*Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.*
- file [oscl\\_string.h](#)  
*Provides a standardized set of string containers that can be used in place of character arrays.*
- file [oscl\\_string\\_containers.h](#)  
*Provides a standardized set of string containers that can be used in place of character arrays.*

- file [oscl\\_string\\_rep.h](#)  
*Contains some internal implementation for string containers.*
- file [oscl\\_string\\_uri.h](#)  
*Utilities to unescape URIs.*
- file [oscl\\_string\\_utf8.h](#)  
*Utilities to validate and truncate UTF-8 encoded strings.*
- file [oscl\\_string\\_utils.h](#)  
*Utilities to parse and convert strings.*
- file [oscl\\_string\\_xml.h](#)  
*Utilities to escape special characters in XML strings.*
- file [oscl\\_tickcount.h](#)  
*Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.*
- file [oscl\\_utf8conv.h](#)  
*Utilities to convert unicode to utf8 and vice versa.*

## Data Structures

- class [BufferFragment](#)
- class [BufferMgr](#)
- class [BufferState](#)
- class [BufFragGroup](#)
- class [BufFragStatusClass](#)
- class [CFastRep](#)
- class [CHheapRep](#)
- class [CStackRep](#)
- class [MediaData](#)
- class [MediaStatusClass](#)
- class [MemAllocator](#)
- class [OSCL\\_FastString](#)
- class [OSCL\\_HeapString](#)
- class [OSCL\\_HeapStringA](#)
- class [OSCL\\_StackString](#)
- class [OSCL\\_String](#)
- class [OSCL\\_wFastString](#)
- class [OSCL\\_wHeapString](#)
- class [OSCL\\_wHeapStringA](#)
- class [OSCL\\_wStackString](#)
- class [OSCL\\_wString](#)
- class [OsclBinIStream](#)
- class [OsclBinIStreamBigEndian](#)
- class [OsclBinIStreamLittleEndian](#)

- class [OsclBinOStream](#)

*Class OsclBinOStream implements the basic stream functions for an output stream.*

- class [OsclBinOStreamBigEndian](#)

*Class OsclBinOStreamBigEndian implements a binary output stream using big endian byte ordering.*

- class [OsclBinOStreamLittleEndian](#)

*Class OsclBinOStreamLittleEndian implements a binary output stream using little endian byte ordering.*

- class [OsclBinStream](#)

- class [OsclCompareLess](#)

- class [OsclComponentRegistry](#)

- class [OsclComponentRegistryData](#)

- class [OsclComponentRegistryElement](#)

- class [OsclPriorityQueue](#)

- class [OsclPriorityQueueBase](#)

- class [OsclRand](#)

- class [OsclRegistryAccessClient](#)

- class [OsclRegistryAccessClientImpl](#)

- class [OsclRegistryAccessClientTlsImpl](#)

- class [OsclRegistryAccessElement](#)

- class [OsclRegistryClient](#)

- class [OsclRegistryClientImpl](#)

- class [OsclRegistryClientTlsImpl](#)

- class [OsclRegistryServTlsImpl](#)

- class [OsclTickCount](#)

- struct [StrCSumPtrLen](#)

*same as [StrPtrLen](#), but includes checksum field and method to speed up querying*

- struct [StrPtrLen](#)

*This data structure encapsulates a set of functions used to perform.*

- struct [WStrPtrLen](#)

*This data structure encapsulates a set of functions used to perform.*

## Defines

- #define [oscl\\_isdigit](#)(c) ((c) >= '0' && (c) <= '9')
- #define [OSCLTICKCOUNT\\_MAX\\_TICKS](#) 0xffffffff

## TypeDefs

- typedef [OsclAny](#) \* [OsclComponentFactory](#)
- typedef void(\* [BufferFreeFuncPtr](#) )(void \*)
- typedef uint32 [MediaTimestamp](#)
- typedef [StrPtrLen](#) [StrPtrLen](#)

*This data structure encapsulates a set of functions used to perform.*

- **typedef WStrPtrLen WStrPtrLen**  
*This data structure encapsulates a set of functions used to perform.*
- **typedef StrCSumPtrLen StrCSumPtrLen**  
*same as [StrPtrLen](#), but includes checksum field and method to speed up querying*
- **typedef WStrPtrLen OSCL\_TStrPtrLen**

## Functions

- **OSCL\_IMPORT\_REF const char \* skip\_whitespace (const char \*ptr)**
- **OSCL\_IMPORT\_REF char \* skip\_whitespace (char \*ptr)**
- **OSCL\_IMPORT\_REF const char \* skip\_whitespace (const char \*start, const char \*end)**
- **OSCL\_IMPORT\_REF const char \* skip\_to\_whitespace (const char \*start, const char \*end)**
- **OSCL\_IMPORT\_REF const char \* skip\_to\_line\_term (const char \*start\_ptr, const char \*end\_ptr)**
- **OSCL\_IMPORT\_REF const char \* skip\_whitespace\_and\_line\_term (const char \*start, const char \*end)**
- **OSCL\_IMPORT\_REF int extract\_string (const char \*in\_ptr, char \*outstring, int maxsize)**
- **OSCL\_IMPORT\_REF int extract\_string (const char \*start, const char \*end, char \*outstring, int maxsize)**
- **OSCL\_IMPORT\_REF bool PV\_atoi (const char \*buf, const char new\_format, uint32 &value)**
- **OSCL\_IMPORT\_REF bool PV\_atoi (const char \*buf, const char new\_format, int length, uint32 &value)**
- **OSCL\_IMPORT\_REF bool PV\_atoi (const char \*buf, const char new\_format, int length, **uint64** &value)**
- **OSCL\_IMPORT\_REF bool PV\_atof (const char \*buf, **OsclFloat** &value)**
- **OSCL\_IMPORT\_REF bool PV\_atof (const char \*buf, int length, **OsclFloat** &value)**
- **OSCL\_IMPORT\_REF int oscl\_abs (int aVal)**
- **OSCL\_COND\_IMPORT\_REF double oscl\_log (double value)**
- **OSCL\_COND\_IMPORT\_REF double oscl\_log10 (double value)**
- **OSCL\_COND\_IMPORT\_REF double oscl\_sqrt (double value)**
- **OSCL\_COND\_IMPORT\_REF double oscl\_pow (double x, double y)**
- **OSCL\_COND\_IMPORT\_REF double oscl\_exp (double value)**
- **OSCL\_COND\_IMPORT\_REF double oscl\_sin (double value)**
- **OSCL\_COND\_IMPORT\_REF double oscl\_cos (double value)**
- **OSCL\_COND\_IMPORT\_REF double oscl\_tan (double value)**
- **OSCL\_COND\_IMPORT\_REF double oscl\_asin (double value)**
- **OSCL\_COND\_IMPORT\_REF double oscl\_atan (double value)**
- **OSCL\_COND\_IMPORT\_REF double oscl\_floor (double value)**
- **OSCL\_IMPORT\_REF int32 oscl\_snprintf (char \*str, uint32 count, const char \*fmt,...)**
- **OSCL\_IMPORT\_REF int32 oscl\_snprintf (**oscl\_wchar** \*str, uint32 count, const **oscl\_wchar** \*fmt,...)**
- **OSCL\_IMPORT\_REF int32 oscl\_vsnprintf (char \*str, uint32 count, const char \*fmt, va\_list args)**
- **OSCL\_IMPORT\_REF int32 oscl\_vsnprintf (**oscl\_wchar** \*str, uint32 count, const **oscl\_wchar** \*fmt, va\_list args)**
- **OSCL\_IMPORT\_REF bool oscl\_str\_unescape\_uri (const char \*str\_buf\_in, char \*str\_buf\_out, uint32 max\_out\_buf\_bytes, uint32 max\_bytes, uint32 &out\_buf\_len)**  
*unescape any of the special escape sequence in the uri string*
- **OSCL\_IMPORT\_REF bool oscl\_str\_unescape\_uri (const **OSCL\_String** &oscl\_str\_in, **OSCL\_String** &oscl\_str\_out, uint32 &out\_buf\_len)**

*unescape any of the special escape sequence in the uri string*

- OSCL\_IMPORT\_REF bool [oscl\\_str\\_is\\_valid\\_utf8](#) (const uint8 \*str\_buf, uint32 &num\_valid\_characters, uint32 max\_bytes=0, uint32 max\_char\_2\_valid=0, uint32 \*num\_byte\_4\_char=NULL)

*Check if the input string contains any illegal UTF-8 character. The function scans the string and validate that each character is a valid utf-8. It stops at the first NULL character, invalid character or the max\_byte value. The string is valid if and only if every character is a valid utf-8 character and the scanning stopped on a character boundary.*

- OSCL\_IMPORT\_REF int32 [oscl\\_str\\_truncate\\_utf8](#) (uint8 \*str\_buf, uint32 max\_char, uint32 max\_bytes=0)

*Truncates the UTF-8 string upto the required size.*

- OSCL\_IMPORT\_REF bool [oscl\\_str\\_need\\_escape\\_xml](#) (const char \*str\_buf, uint32 &num\_escape\_bytes, uint32 max\_bytes=0)

*Check if the input string contains any special ASCII character like &, <, >, ', ". The function scans the string and check if each character is a special character. It stops at the first NULL character (if max\_bytes = 0), or the max\_byte value.*

- OSCL\_IMPORT\_REF int32 [oscl\\_str\\_escape\\_xml](#) (const char \*str\_buf\_in, char \*str\_buf\_out, uint32 max\_out\_buf\_bytes, uint32 max\_bytes=0, uint32 \*num\_bytes\_written=NULL)

*Escape any of the following special characters in the string Special ASCII characters: &, <, >, ', ".*

- OSCL\_IMPORT\_REF int32 [oscl\\_UTF8ToUnicode](#) (const char \*input, int32 inLength, oscl\_wchar \*output, int32 outLength)

*Convert UTF8 byte sequence to Unicode string.*

- OSCL\_IMPORT\_REF int32 [oscl\\_UncodeToUTF8](#) (const oscl\_wchar \*input, int32 inLength, char \*output, int32 outLength)

*Convert Unicode string to UTF8 byte sequence.*

- [BufferFragment \\* GetFragment](#) (const int32 idx)
- [BufferState \\* GetBufferState](#) (const int32 idx)
- uint32 [get\\_size](#) () const
- uint32 [get\\_size](#) () const
- uint32 [get\\_maxsize](#) () const
- uint32 [get\\_maxsize](#) () const
- const chartype \* [get\\_cstr](#) () const
- const chartype \* [get\\_cstr](#) () const
- chartype \* [get\\_str](#) () const
- chartype \* [get\\_str](#) () const
- [OSCL\\_HeapString](#) ()
- [OSCL\\_wHeapString](#) ()
- [OSCL\\_HeapString](#) (const chartype \*cstr)
- [OSCL\\_wHeapString](#) (const chartype \*cstr)
- void [set](#) (const chartype \*buf, uint32 length)
- void [set](#) (const chartype \*buf, uint32 length)
- [OSCL\\_HeapString](#) (const chartype \*buf, uint32 length)
- [OSCL\\_wHeapString](#) (const chartype \*buf, uint32 length)
- [OSCL\\_HeapString](#) (const OSCL\_HeapString &src)
- [OSCL\\_wHeapString](#) (const OSCL\_wHeapString &src)

- `OSCL_HeapString` (const `OSCL_String` &src)
- `OSCL_wHeapString` (const `OSCL_wString` &src)
- `~OSCL_HeapString` ()
- `~OSCL_wHeapString` ()
- `OSCL_HeapString & operator=` (const `OSCL_HeapString` &src)
- `OSCL_wHeapString & operator=` (const `OSCL_wHeapString` &src)
- `OSCL_HeapString & operator=` (const `OSCL_String` &src)
- `OSCL_wHeapString & operator=` (const `OSCL_wString` &src)
- `OSCL_HeapString & operator=` (const chartype \*cstr)
- `OSCL_wHeapString & operator=` (const chartype \*cstr)
- `uint32 get_size` () const
- `uint32 get_size` () const
- `uint32 get_maxsize` () const
- `uint32 get_maxsize` () const
- `const chartype * get_cstr` () const
- `const chartype * get_cstr` () const
- `chartype * get_str` () const
- `chartype * get_str` () const
- `OSCL_StackString` ()
- `OSCL_wStackString` ()
- `OSCL_StackString` (const chartype \*cstr)
- `OSCL_wStackString` (const chartype \*cstr)
- `void set` (const chartype \*buf, uint32 length)
- `void set` (const chartype \*buf, uint32 length)
- `OSCL_StackString` (const chartype \*buf, uint32 length)
- `OSCL_wStackString` (const chartype \*buf, uint32 length)
- `OSCL_StackString` (const `OSCL_StackString` &src)
- `OSCL_wStackString` (const `OSCL_wStackString` &src)
- `OSCL_StackString` (const `OSCL_String` &src)
- `OSCL_wStackString` (const `OSCL_wString` &src)
- `~OSCL_StackString` ()
- `~OSCL_wStackString` ()
- `OSCL_StackString & operator=` (const `OSCL_StackString` &src)
- `OSCL_wStackString & operator=` (const `OSCL_wStackString` &src)
- `OSCL_StackString & operator=` (const `OSCL_String` &src)
- `OSCL_wStackString & operator=` (const `OSCL_wString` &src)
- `OSCL_StackString & operator=` (const chartype \*cstr)
- `OSCL_wStackString & operator=` (const chartype \*cstr)

## Variables

- `const int32 APPEND_MEDIA_AT_END = -1`
- `const uint8 OSCL_ASCII_CASE_MAGIC_BIT = 0x20`

### 5.4.1 Define Documentation

5.4.1.1 `#define oscl_isdigit(c) ((c) >= '0' && (c) <= '9')`

5.4.1.2 `#define OSCLTICKCOUNT_MAX_TICKS 0xffffffff`

### 5.4.2 Typedef Documentation

5.4.2.1 `typedef void(* BufferFreeFuncPtr)(void *)`

5.4.2.2 `typedef uint32 MediaTimestamp`

5.4.2.3 `typedef WStrPtrLen OSCL_TStrPtrLen`

5.4.2.4 `typedef OsclAny* OsclComponentFactory`

`OsclComponentFactory` is an opaque pointer.

5.4.2.5 `typedef StrCSumPtrLen StrCSumPtrLen`

same as `StrPtrLen`, but includes checksum field and method to speed up querying

5.4.2.6 `typedef struct StrPtrLen StrPtrLen`

This data structure encapsulates a set of functions used to perform.

standard string operations. It should be used for null-terminated constant (non-modifiable) strings of char type.

5.4.2.7 `typedef struct WStrPtrLen WStrPtrLen`

This data structure encapsulates a set of functions used to perform.

standard string operations. It should be used for null-terminated constant strings (non-modifiable) of wchar type.

### 5.4.3 Function Documentation

5.4.3.1 `OSCL_IMPORT_REF int extract_string (const char * start, const char * end, char * outstring, int maxsize)`

5.4.3.2 `OSCL_IMPORT_REF int extract_string (const char * in_ptr, char * outstring, int maxsize)`

5.4.3.3 `template<uint32 MaxBufSize> const OSCL_wStackString< MaxBufSize >::chartype * OSCL_wStackString< MaxBufSize >::get_cstr () [virtual, inherited]`

Implements `OSCL_wString`.

**5.4.3.4 template<uint32 MaxBufSize> const OSCL\_StackString< MaxBufSize >::chartype \* OSCL\_StackString< MaxBufSize >::get\_cstr () [virtual, inherited]**

This function returns the C-style string for read access.

Implements [OSCL\\_String](#).

**5.4.3.5 template<class Alloc> const OSCL\_wHeapString< Alloc >::chartype \* OSCL\_wHeapString< Alloc >::get\_cstr () [virtual, inherited]**

Implements [OSCL\\_wString](#).

**5.4.3.6 template<class Alloc> const OSCL\_HeapString< Alloc >::chartype \* OSCL\_HeapString< Alloc >::get\_cstr () [virtual, inherited]**

This function returns the C-style string for read access.

Implements [OSCL\\_String](#).

**5.4.3.7 template<uint32 MaxBufSize> uint32 OSCL\_wStackString< MaxBufSize >::get\_maxsize () [virtual, inherited]**

Implements [OSCL\\_wString](#).

**5.4.3.8 template<uint32 MaxBufSize> uint32 OSCL\_StackString< MaxBufSize >::get\_maxsize () [virtual, inherited]**

This function returns the maximum available storage size, not including null terminator. The maximum size may be larger than the current string size.

Implements [OSCL\\_String](#).

**5.4.3.9 template<class Alloc> uint32 OSCL\_wHeapString< Alloc >::get\_maxsize () [virtual, inherited]**

Implements [OSCL\\_wString](#).

**5.4.3.10 template<class Alloc> uint32 OSCL\_HeapString< Alloc >::get\_maxsize () [virtual, inherited]**

This function returns the maximum available storage size, not including null terminator. The maximum size may be larger than the current string size.

Implements [OSCL\\_String](#).

**5.4.3.11 template<uint32 MaxBufSize> uint32 OSCL\_wStackString< MaxBufSize >::get\_size () [virtual, inherited]**

Implements [OSCL\\_wString](#).

**5.4.3.12 template<uint32 MaxBufSize> uint32 OSCL\_StackString< MaxBufSize >::get\_size () [virtual, inherited]**

Pure virtuals from [OSCL\\_String](#)

Implements [OSCL\\_String](#).

**5.4.3.13 template<class Alloc> uint32 OSCL\_wHeapString< Alloc >::get\_size () [virtual, inherited]**

Implements [OSCL\\_wString](#).

**5.4.3.14 template<class Alloc> uint32 OSCL\_HeapString< Alloc >::get\_size () [virtual, inherited]**

Pure virtuals from [OSCL\\_String](#)

Implements [OSCL\\_String](#).

**5.4.3.15 template<uint32 MaxBufSize> OSCL\_wStackString< MaxBufSize >::chartype \* OSCL\_wStackString< MaxBufSize >::get\_str () [virtual, inherited]**

Implements [OSCL\\_wString](#).

**5.4.3.16 template<uint32 MaxBufSize> OSCL\_StackString< MaxBufSize >::chartype \* OSCL\_StackString< MaxBufSize >::get\_str () [virtual, inherited]**

This function returns the C-style string for write access. If the string is not writable it returns NULL.

Implements [OSCL\\_String](#).

**5.4.3.17 template<class Alloc> OSCL\_wHeapString< Alloc >::chartype \* OSCL\_wHeapString< Alloc >::get\_str () [virtual, inherited]**

Implements [OSCL\\_wString](#).

**5.4.3.18 template<class Alloc> OSCL\_HeapString< Alloc >::chartype \* OSCL\_HeapString< Alloc >::get\_str () [virtual, inherited]**

This function returns the C-style string for write access. If the string is not writable it returns NULL.

Implements [OSCL\\_String](#).

- 5.4.3.19 template<class ChainClass, uint32 max\_frags> **BufferState** \* **BuffFragGroup**<  
 ChainClass, max\_frags >::GetBufferState (const int32 *idx*) [inline, inherited]
- 5.4.3.20 template<class ChainClass, uint32 max\_frags> **BufferFragment** \* **BuffFragGroup**<  
 ChainClass, max\_frags >::GetFragment (const int32 *idx*) [inline, inherited]
- 5.4.3.21 template<uint32 MaxBufSize> **OSCL\_wStackString**< MaxBufSize > &  
**OSCL\_wStackString**< MaxBufSize >::operator= (const **chartype** \* *cstr*) [inherited]

Reimplemented from [OSCL\\_wString](#).

- 5.4.3.22 template<uint32 MaxBufSize> **OSCL\_StackString**< MaxBufSize > &  
**OSCL\_StackString**< MaxBufSize >::operator= (const **chartype** \* *cstr*) [inherited]

Assignment operator

**am:** null-terminated string

Reimplemented from [OSCL\\_String](#).

- 5.4.3.23 template<uint32 MaxBufSize> **OSCL\_wStackString**< MaxBufSize > &  
**OSCL\_wStackString**< MaxBufSize >::operator= (const [OSCL\\_wString](#) & *src*)  
 [inherited]

Reimplemented from [OSCL\\_wString](#).

- 5.4.3.24 template<uint32 MaxBufSize> **OSCL\_StackString**< MaxBufSize > &  
**OSCL\_StackString**< MaxBufSize >::operator= (const [OSCL\\_String](#) & *src*)  
 [inherited]

Assignment operator

Reimplemented from [OSCL\\_String](#).

- 5.4.3.25 template<uint32 MaxBufSize> **OSCL\_wStackString**< MaxBufSize > &  
**OSCL\_wStackString**< MaxBufSize >::operator= (const [OSCL\\_wStackString](#)<  
 MaxBufSize > & *src*) [inherited]

- 5.4.3.26 template<uint32 MaxBufSize> **OSCL\_StackString**< MaxBufSize > &  
**OSCL\_StackString**< MaxBufSize >::operator= (const [OSCL\\_StackString](#)< MaxBufSize  
 > & *src*) [inherited]

Assignment operators

- 5.4.3.27 template<class Alloc> **OSCL\_wHeapString**< Alloc > & **OSCL\_wHeapString**< Alloc  
 >::operator= (const **chartype** \* *cstr*) [inherited]

Reimplemented from [OSCL\\_wString](#).

**5.4.3.28 template<class Alloc> OSCL\_HeapString< Alloc > & OSCL\_HeapString< Alloc >::operator= (const chartype \* *cstr*) [inherited]**

Assignment operator

**am:** null-terminated string

Reimplemented from [OSCL\\_String](#).

**5.4.3.29 template<class Alloc> OSCL\_wHeapString< Alloc > & OSCL\_wHeapString< Alloc >::operator= (const OSCL\_wString & *src*) [inherited]**

Reimplemented from [OSCL\\_wString](#).

**5.4.3.30 template<class Alloc> OSCL\_HeapString< Alloc > & OSCL\_HeapString< Alloc >::operator= (const OSCL\_String & *src*) [inherited]**

Assignment operator

Reimplemented from [OSCL\\_String](#).

**5.4.3.31 template<class Alloc> OSCL\_wHeapString< Alloc > & OSCL\_wHeapString< Alloc >::operator= (const OSCL\_wHeapString< Alloc > & *src*) [inherited]**

**5.4.3.32 template<class Alloc> OSCL\_HeapString< Alloc > & OSCL\_HeapString< Alloc >::operator= (const OSCL\_HeapString< Alloc > & *src*) [inherited]**

Assignment operators

**5.4.3.33 OSCL\_IMPORT\_REF int oscl\_abs (int *aVal*)**

**5.4.3.34 OSCL\_COND\_IMPORT\_REF double oscl\_asin (double *value*)**

Calculates the arc sine of a number

**Parameters:**

*value* source value

**5.4.3.35 OSCL\_COND\_IMPORT\_REF double oscl\_atan (double *value*)**

Calculates the arc tangent of a number

**Parameters:**

*value* source value

**5.4.3.36 OSCL\_COND\_IMPORT\_REF double oscl\_cos (double *value*)**

Calculates the cosine of a number

**Parameters:**

*value* source value

**5.4.3.37 OSCL\_COND\_IMPORT\_REF double oscl\_exp (double *value*)**

Calculates the exponential of e for a number

**Parameters:**

*value* source value

**5.4.3.38 OSCL\_COND\_IMPORT\_REF double oscl\_floor (double *value*)**

Calculates the floor of a number

**Parameters:**

*value* source value

**5.4.3.39 template<class Alloc> OSCL\_HeapString< Alloc >::OSCL\_HeapString (const OSCL\_String & *src*) [inherited]****5.4.3.40 template<class Alloc> OSCL\_HeapString< Alloc >::OSCL\_HeapString (const OSCL\_HeapString< Alloc > & *src*) [inherited]**

Creates a heap string that contains a copy of the input string.

**Parameters:**

*src*: input string.

**5.4.3.41 template<class Alloc> OSCL\_HeapString< Alloc >::OSCL\_HeapString (const chartype \* *buf*, uint32 *length*) [inherited]**

Creates a heap string that contains a copy of the input string or character array.

**Parameters:**

*src*: character array, not necessarily null-terminated.

*length*: number of characters to copy.

**5.4.3.42 template<class Alloc> OSCL\_HeapString< Alloc >::OSCL\_HeapString (const chartype \* *cstr*) [inherited]**

Creates a heap string that contains a copy of the input string.

**Parameters:**

*cp*: null-terminated string.

**5.4.3.43 template<class Alloc> OSCL\_HeapString< Alloc >::OSCL\_HeapString ()  
[inherited]**

The default constructor creates an empty string.

**5.4.3.44 OSCL\_COND\_IMPORT\_REF double oscl\_log (double *value*)**

Calculates the natural log of a number

**Parameters:**

*value* source value

**5.4.3.45 OSCL\_COND\_IMPORT\_REF double oscl\_log10 (double *value*)**

Calculates the logarithm to base 10 of a number

**Parameters:**

*value* source value

**5.4.3.46 OSCL\_COND\_IMPORT\_REF double oscl\_pow (double *x*, double *y*)**

Calculates the value of *x* to the power of *y*

**Parameters:**

*x* base value

*y* power

**5.4.3.47 OSCL\_COND\_IMPORT\_REF double oscl\_sin (double *value*)**

Calculates the sine of a number

**Parameters:**

*value* source value

**5.4.3.48 OSCL\_IMPORT\_REF int32 oscl\_snprintf (*oscl\_wchar* \* *str*, uint32 *count*, const *oscl\_wchar* \* *fmt*, ...)****5.4.3.49 OSCL\_IMPORT\_REF int32 oscl\_snprintf (char \* *str*, uint32 *count*, const char \* *fmt*, ...)****5.4.3.50 OSCL\_COND\_IMPORT\_REF double oscl\_sqrt (double *value*)**

Calculates the square root of a number

**Parameters:**

*value* source value

**5.4.3.51 template<uint32 MaxBufSize> OSCL\_StackString< MaxBufSize >::OSCL\_StackString  
(const OSCL\_String & src) [inherited]**

**5.4.3.52 template<uint32 MaxBufSize> OSCL\_StackString< MaxBufSize >::OSCL\_StackString  
(const OSCL\_StackString< MaxBufSize > & src) [inherited]**

Creates an OSCL\_StackString with a copy of the input string. The string may be truncated to fit the available storage.

**Parameters:**

*src*: input string.

**5.4.3.53 template<uint32 MaxBufSize> OSCL\_StackString< MaxBufSize >::OSCL\_StackString  
(const chartype \* buf, uint32 length) [inherited]**

Creates an OSCL\_StackString with a copy of the input string. The string may be truncated to fit the available storage.

**Parameters:**

*src*: a character array, not necessarily null-terminated.

*length*: the number of characters to copy.

**5.4.3.54 template<uint32 MaxBufSize> OSCL\_StackString< MaxBufSize >::OSCL\_StackString  
(const chartype \* cstr) [inherited]**

Creates an OSCL\_StackString with a copy of the input string. The string may be truncated to fit the available storage.

**Parameters:**

*cp*: a null-terminated string.

**5.4.3.55 template<uint32 MaxBufSize> OSCL\_StackString< MaxBufSize >::OSCL\_StackString  
() [inherited]**

Creates an OSCL\_StackString initialized with an empty string.

**5.4.3.56 OSCL\_IMPORT\_REF int32 oscl\_str\_escape\_xml (const char \* str\_buf\_in, char \*  
str\_buf\_out, uint32 max\_out\_buf\_bytes, uint32 max\_bytes = 0, uint32 \* num\_bytes\_written  
= NULL)**

Escape any of the following special characters in the string Special ASCII characters: &, <, >, ', ".

The function scans the string and replaces each special character with its corresponding escape sequence. It stops at the first NULL character, the max\_byte value.

**Parameters:**

*str\_buf\_in* Ptr to an input string

*str\_buf\_out* Ptr to an output buffer which stores the modified string

*max\_out\_buf\_bytes* The size of str\_buf\_out.

**max\_bytes** The maximum number of bytes to read (a zero value means read to the first NULL character). It is the length of str\_buf\_in.

**num\_bytes\_written** Number of bytes written in the output buffer, str\_buf\_out

**Returns:**

It returns the number of bytes in the str\_buf\_out if succeeded. It returns negative number if failed, and its absolute value indicates the total number bytes written to the output buffer, str\_buf\_out, if str\_buf\_out != null.

#### 5.4.3.57 OSCL\_IMPORT\_REF bool oscl\_str\_is\_valid\_utf8 (const uint8 \* str\_buf, uint32 & num\_valid\_characters, uint32 max\_bytes = 0, uint32 max\_char\_2\_valid = 0, uint32 \* num\_byte\_4\_char = NULL)

Check if the input string contains any illegal UTF-8 character. The function scans the string and validate that each character is a valid utf-8. It stops at the first NULL character, invalid character or the max\_byte value. The string is valid if and only if every character is a valid utf-8 character and the scanning stopped on a character boundary.

**Parameters:**

**str\_buf** Ptr to an input string, which may not terminate with null, to be checked

**num\_valid\_chars** This is an output parameter which is the number of valid utf-8 characters actually read.

**max\_bytes** The maximum number of bytes to read (a zero value means read to the first NULL character).

**max\_char\_2\_valid** This is an input parameter. Specify the number of utf-8 characters the caller wants to validate.

**num\_byte\_4\_char** This is an output parameter. The number of bytes used by the max\_char characters

**Returns:**

True if the string is valid and false otherwise.

#### 5.4.3.58 OSCL\_IMPORT\_REF bool oscl\_str\_need\_escape\_xml (const char \* str\_buf, uint32 & num\_escape\_bytes, uint32 max\_bytes = 0)

Check if the input string contains any special ASCII character like &, <, >, ', ". The function scans the string and check if each character is a special character. It stops at the first NULL character (if max\_bytes = 0), or the max\_byte value.

**Parameters:**

**str\_buf** Ptr to an input string, which may not terminate with null, to be checked

**num\_escape\_bytes** This is an output parameter which is the number of bytes needed to hold the result string. Value 0 indicates that there is no special character found. If max\_bytes = 0, the return value does not include the null character.

**max\_bytes** The maximum number of bytes to read (a zero value means read to the first NULL character).

**Returns:**

True if the function succeeds, and num\_escape\_bytes = 0 means that no special character is found, num\_escape\_bytes >0 means the number of bytes of the result string. False if there is any error occurred.

**5.4.3.59 OSCL\_IMPORT\_REF int32 oscl\_str\_truncate\_utf8 (uint8 \* str\_buf, uint32 max\_char, uint32 max\_bytes = 0)**

Truncates the UTF-8 string upto the required size.

The function will modify the str\_buf so that it contains AT MOST len valid utf-8 characters. If a NULL character is found before reading len utf-8 characters, then the function does not modify the string and simply returns the number of characters. If an invalid character is found, then it will insert a NULL character after the last valid character and return the length. Otherwise, it will insert a NULL character after len valid utf-8 characters and return the length.

**Parameters:**

**str\_buf** Ptr to an input string which may not terminate with null

**max\_char** The max number of the UTF-8 CHARACTERS

**max\_bytes** The maximum number of bytes to read (a zero value means read to the first NULL character).

**Returns:**

It returns the length of the truncated string in utf-8 characters.

**5.4.3.60 OSCL\_IMPORT\_REF bool oscl\_str\_unescape\_uri (const OSCL\_String & oscl\_str\_in, OSCL\_String & oscl\_str\_out, uint32 & out\_buf\_len)**

unescape any of the special escape sequence in the uri string

The function scans the string and replaces each escape sequence with its corresponding character. It stops at the first null character, or the max\_byte value. It returns false if the string contains any illegal escape sequence or the output buffer is not big enough. The out\_buf\_len value indicates the needed buffer length or the index of the byte that causes the error respectively.

**Parameters:**

**oscl\_str\_in** Ptr to an input **OSCL\_String**

**oscl\_str\_out** Ptr to an output **OSCL\_String** which stores the modified string

**out\_buf\_len** The length of the result string (not including the null character)

**Returns:**

It returns true if succeeds, otherwise false.

**5.4.3.61 OSCL\_IMPORT\_REF bool oscl\_str\_unescape\_uri (const char \* str\_buf\_in, char \* str\_buf\_out, uint32 max\_out\_buf\_bytes, uint32 max\_bytes, uint32 & out\_buf\_len)**

unescape any of the special escape sequence in the uri string

The function scans the string and replaces each escape sequence with its corresponding character. It stops at the first null character, or the max\_byte value. It returns false if the string contains any illegal escape sequence or the output buffer is not big enough. The out\_buf\_len value indicates the needed buffer length or the index of the byte that causes the error respectively.

**Parameters:**

**str\_buf\_in** Ptr to an input string

**str\_buf\_out** Ptr to an output buffer which stores the modified string

*max\_out\_buf\_bytes* The size of str\_buf\_out.

*max\_bytes* The maximum number of bytes to read. It is the length of str\_buf\_in.

*out\_buf\_len* The length of the result string (not including the null character)

**Returns:**

It returns true if succeeds, otherwise false.

#### 5.4.3.62 OSCL\_COND\_IMPORT\_REF double oscl\_tan (double *value*)

Calculates the tangential of a number

**Parameters:**

*value* source value

#### 5.4.3.63 OSCL\_IMPORT\_REF int32 oscl\_UnicodeToUTF8 (const oscl\_wchar \* *input*, int32 *inLength*, char \* *output*, int32 *outLength*)

Convert Unicode string to UTF8 byte sequence.

The function converts Unicode string to UTF8 byte sequence. The length of input Unicode string is specified. It stops at two conditions: (A) Whole input Unicode string is successfully converted. (B) Destination buferr is not enough for output. In case of (A), it adds a terminated '\0' at the end of the output UTF8 byte sequence. and returns length of the output UTF8 byte sequence(without counting terminated '\0'). In case of (B), it converts as much as possible to the output buffer and adds a terminated '\0' at the end of the output UTF8 byte sequence"(no '\0' added if outLength is less than or equal to 0, return 0)", and returns 0.

**Parameters:**

*input* Ptr to an input Unicode string. '\0' termanation is not neccesary.

*inLength* The length of the input Unicode string, without counting terminated '\0'(if any).

*output* Ptr to an output buffer which output UTF8 byte sequence is written in.

*outLength* The size of output buffer, also the maximum number of char could be written in.

**Returns:**

length of output (excludes '\0') : completely converts all input string and appends '\0' to output; 0 : insufficient buffer or error in conversion

#### 5.4.3.64 OSCL\_IMPORT\_REF int32 oscl\_UTF8ToUnicode (const char \* *input*, int32 *inLength*, oscl\_wchar \* *output*, int32 *outLength*)

Convert UTF8 byte sequence to Unicode string.

The function converts UTF8 byte sequence (or ASCII sequence) to Unicode string. The length of input UTF8 byte sequence is specified. It stops at two conditions: (A) Whole input UTF8 byte sequence is successfully converted. (B) Output buferr is not enough for output, or parse error. In case of (A), it adds a terminated '\0' at the end of the output Unicode string, and returns length of the output Unicode string(without counting terminated '\0'). In case of (B), it converts as much as possible to the output buffer and adds a terminated '\0' at the end of the output Unicode string"(no '\0' added if outLength is less than or equal to 0, return 0)", and returns 0.

**Parameters:**

*input* Ptr to an input UTF8 byte sequence. '\0' termination is not necessary.

*inLength* The length of the input UTF8 byte sequence, without counting terminated '\0'(if any).

*output* Ptr to an output buffer which output Unicode string is written in.

*outLength* The size of output buffer, also the maximum number of oscl\_wchar could be written in.

**Returns:**

Length of output (excludes '\0') : completely converts all input string and appends '\0' to output; 0 : insufficient buffer or error in conversion



- 5.4.3.65 **OSCL\_IMPORT\_REF int32 oscl\_vsnprintf (oscl\_wchar \* str, uint32 count, const oscl\_wchar \* fmt, va\_list args)**
- 5.4.3.66 **OSCL\_IMPORT\_REF int32 oscl\_vsnprintf (char \* str, uint32 count, const char \* fmt, va\_list args)**
- 5.4.3.67 **template<class Alloc> OSCL\_wHeapString< Alloc >::OSCL\_wHeapString (const OSCL\_wString & src) [inherited]**
- 5.4.3.68 **template<class Alloc> OSCL\_wHeapString< Alloc >::OSCL\_wHeapString (const OSCL\_wHeapString< Alloc > & src) [inherited]**
- 5.4.3.69 **template<class Alloc> OSCL\_wHeapString< Alloc >::OSCL\_wHeapString (const chartype \* buf, uint32 length) [inherited]**
- 5.4.3.70 **template<class Alloc> OSCL\_wHeapString< Alloc >::OSCL\_wHeapString (const chartype \* cstr) [inherited]**
- 5.4.3.71 **template<class Alloc> OSCL\_wHeapString< Alloc >::OSCL\_wHeapString () [inherited]**
- 5.4.3.72 **template<uint32 MaxBufSize> OSCL\_wStackString< MaxBufSize >::OSCL\_wStackString (const OSCL\_wString & src) [inherited]**
- 5.4.3.73 **template<uint32 MaxBufSize> OSCL\_wStackString< MaxBufSize >::OSCL\_wStackString (const OSCL\_wStackString< MaxBufSize > & src) [inherited]**
- 5.4.3.74 **template<uint32 MaxBufSize> OSCL\_wStackString< MaxBufSize >::OSCL\_wStackString (const chartype \* buf, uint32 length) [inherited]**
- 5.4.3.75 **template<uint32 MaxBufSize> OSCL\_wStackString< MaxBufSize >::OSCL\_wStackString (const chartype \* cstr) [inherited]**
- 5.4.3.76 **template<uint32 MaxBufSize> OSCL\_wStackString< MaxBufSize >::OSCL\_wStackString () [inherited]**
- 5.4.3.77 **OSCL\_IMPORT\_REF bool PV\_atof (const char \* buf, int length, OsclFloat & value)**
- 5.4.3.78 **OSCL\_IMPORT\_REF bool PV\_atof (const char \* buf, OsclFloat & value)**
- 5.4.3.79 **OSCL\_IMPORT\_REF bool PV\_atoi (const char \* buf, const char new\_format, int length, uint64 & value)**
- 5.4.3.80 **OSCL\_IMPORT\_REF bool PV\_atoi (const char \* buf, const char new\_format, int length, uint32 & value)**
- 5.4.3.81 **OSCL\_IMPORT\_REF bool PV\_atoi (const char \* buf, const char new\_format, uint32 & value)**
- 5.4.3.82 **template<uint32 MaxBufSize> void OSCL\_wStackString< MaxBufSize >::set (const chartype \* buf, uint32 length) [inherited]**
- 5.4.3.83 **template<uint32 MaxBufSize> void OSCL\_StackString< MaxBufSize >::set (const chartype \* buf, uint32 length) [inherited]**

**Parameters:**

*buf*: string or character array.

*length*: number of characters to copy.

**5.4.3.84 template<class Alloc> void OSCL\_wHeapString< Alloc >::set (const chartype \* *buf*,  
                  uint32 *length*) [inherited]**

**5.4.3.85 template<class Alloc> void OSCL\_HeapString< Alloc >::set (const chartype \* *buf*,  
                  uint32 *length*) [inherited]**

Set the contents of this string to a new string or character array.

**Parameters:**

*buf*: string or character array.

*length*: number of characters to copy.

**5.4.3.86 OSCL\_IMPORT\_REF const char\* skip\_to\_line\_term (const char \* *start\_ptr*, const char \*  
                  *end\_ptr*)**

**5.4.3.87 OSCL\_IMPORT\_REF const char\* skip\_to\_whitespace (const char \* *start*, const char \*  
                  *end*)**

**5.4.3.88 OSCL\_IMPORT\_REF const char\* skip\_whitespace (const char \* *start*, const char \* *end*)**

**5.4.3.89 OSCL\_IMPORT\_REF char\* skip\_whitespace (char \* *ptr*)**

**5.4.3.90 OSCL\_IMPORT\_REF const char\* skip\_whitespace (const char \* *ptr*)**

**5.4.3.91 OSCL\_IMPORT\_REF const char\* skip\_whitespace\_and\_line\_term (const char \* *start*,  
                  const char \* *end*)**

**5.4.3.92 template<class Alloc> OSCL\_HeapString< Alloc >::~OSCL\_HeapString ()  
                  [inherited]**

**5.4.3.93 template<uint32 MaxBufSize> OSCL\_StackString< MaxBufSize  
                  >::~OSCL\_StackString () [inherited]**

**5.4.3.94 template<class Alloc> OSCL\_wHeapString< Alloc >::~OSCL\_wHeapString ()  
                  [inherited]**

**5.4.3.95 template<uint32 MaxBufSize> OSCL\_wStackString< MaxBufSize  
                  >::~OSCL\_wStackString () [inherited]**

## 5.4.4 Variable Documentation

**5.4.4.1 const int32 APPEND\_MEDIA\_AT\_END = -1**

**5.4.4.2 const uint8 OSCL\_ASCII\_CASE\_MAGIC\_BIT = 0x20**

## 5.5 OSCL Error

### Files

- file [oscl\\_errno.h](#)  
*Defines functions to access additional information on errors where supported through an errno or similar service.*
- file [oscl\\_error.h](#)  
*OSCL Error trap and cleanup include file.*
- file [oscl\\_error\\_allocator.h](#)  
*Defines a memory allocation class used by the oscl error layer.*
- file [oscl\\_error\\_codes.h](#)  
*Defines basic error and leave codes.*
- file [oscl\\_error\\_imp.h](#)  
*Internal error implementation support.*
- file [oscl\\_error\\_imp\\_cppexceptions.h](#)  
*Implementation File for Leave using C++ exceptions.*
- file [oscl\\_error\\_imp\\_fatalerror.h](#)  
*Implementation File for Leave using system fatal error.*
- file [oscl\\_error\\_imp\\_jumps.h](#)  
*Implementation of using Setjmp / Longjmp.*
- file [oscl\\_error\\_trapcleanup.h](#)  
*OSCL Error trap and cleanup implementation include file.*
- file [oscl\\_exception.h](#)  
*contains all the exception handling macros and classes*
- file [oscl\\_heapbase.h](#)  
*OSCL Heap Base include file.*
- file [oscl\\_mempool\\_allocator.h](#)  
*This file contains the definition of memory pool allocator for leave/trap.*
- file [oscl\\_namestring.h](#)  
*Name string class include file.*

### Data Structures

- class [\\_OsclHeapBase](#)
- class [internalLeave](#)
- class [OsclError](#)

- class [OsclErrorAllocator](#)

*This class provides static methods to invoke the user defined memory allocation routines.*

- class [OsclErrorTrap](#)
- class [OsclErrorTrapImp](#)
- class [OsclException](#)

*oscl\_exception.h contains all the exception handling macros and classes This template class provides the base exception class that all exceptions derive from*

- class [OsclJump](#)
- class [OsclMemPoolAllocator](#)
- class [OsclNameString](#)
- class [OsclTLSEx](#)
- class [OsclTLSRegistryEx](#)
- class [OsclTrapItem](#)
- class [OsclTrapStack](#)
- class [OsclTrapStackItem](#)

## Defines

- #define [OSCL\\_TRAPSTACK\\_PUSH](#)(a) OsclError::PushL(a)
- #define [OSCL\\_TRAPSTACK\\_POP](#)() OsclError::Pop()
- #define [OSCL\\_TRAPSTACK\\_POPDEALLOC](#)() OsclError::PopDealloc()
- #define [OsclErrNone](#) 0
- #define [OsclErrGeneral](#) 100
- #define [OsclErrNoMemory](#) 101
- #define [OsclErrCancelled](#) 102
- #define [OsclErrNotSupported](#) 103
- #define [OsclErrArgument](#) 104
- #define [OsclErrBadHandle](#) 105
- #define [OsclErrAlreadyExists](#) 106
- #define [OsclErrBusy](#) 107
- #define [OsclErrNotReady](#) 108
- #define [OsclErrCorrupt](#) 109
- #define [OsclErrTimeout](#) 110
- #define [OsclErrOverflow](#) 111
- #define [OsclErrUnderflow](#) 112
- #define [OsclErrInvalidState](#) 113
- #define [OsclErrNoResources](#) 114
- #define [OsclErrNotInstalled](#) 115
- #define [OsclErrAlreadyInstalled](#) 116
- #define [OsclErrSystemCallFailed](#) 117
- #define [OsclErrNoHandler](#) 118
- #define [OsclErrThreadContextIncorrect](#) 119
- #define [OSCL\\_ERR\\_NONE](#) OsclErrNone
- #define [OSCL\\_BAD\\_ALLOC\\_EXCEPTION\\_CODE](#) OsclErrNoMemory
- #define [OsclSuccess](#) 0
- #define [OsclPending](#) 1
- #define [OsclFailure](#) -1
- #define [PVERROR\\_IMP\\_JUMPS](#)

- #define **PVError\_DoLeave()** internalLeave \_\_ilv; \_\_ilv.a=0;throw(\_\_ilv)
- #define **\_PV\_TRAP(\_r, \_s)**
- #define **\_PV\_TRAP\_NO\_TLS(\_trapimp, \_r, \_s)**
- #define **OSCL\_JUMP\_MAX\_JUMP\_MARKS** OSCL\_MAX\_TRAP\_LEVELS
- #define **internalLeave (-1)**
- #define **OSCL\_MAX\_TRAP\_LEVELS** 20
- #define **PVERRORTRAP\_REGISTRY\_ID** OSCL\_TLS\_ID\_PVERRORTRAP
- #define **PVERRORTRAP\_REGISTRY** OsclTLSRegistry
- #define **OSCL\_LEAVE(\_leave\_status)** OsclError::Leave(\_leave\_status)

*Use this macro to cause a Leave. It terminates the execution of the current active function.*

- #define **OSCL\_TRY(\_leave\_status, \_statements)** \_PV\_TRAP(\_leave\_status,\_statements)

*This macro will be used to set up a try block.*

- #define **OSCL\_TRY\_NO\_TLS(\_trapimp, \_leave\_status, \_statements)** \_PV\_TRAP\_NO\_TLS(\_-trapimp,\_leave\_status,\_statements)
- #define **OSCL\_FIRST\_CATCH\_ANY(\_leave\_status, \_statements)** if (\_leave\_status!=OsclErrNone) { \_statements; }

*This section defines the macros to be used in the catch block following the try block Use this macro to call a function that handles all exception types thrown in the preceding try block.*

- #define **OSCL\_FIRST\_CATCH(\_leave\_status, \_catch\_value, \_statements)** if (\_leave\_status!=OsclErrNone && \_leave\_status == \_catch\_value){\_statements;}

*Use this macro to define a block of code that catches the first exception type thrown in the preceding try block.*

- #define **OSCL\_CATCH(\_leave\_status, \_catch\_value, \_statements)** else if (\_leave\_status!=OsclErrNone && \_leave\_status == \_catch\_value){\_statements;}

*Use this macro to define a block of code for catching additional exception types.*

- #define **OSCL\_CATCH\_ANY(\_leave\_status, \_statements)** else if (\_leave\_status!=OsclErrNone){ \_-statements; }

*Use this macro to call a function that will catch all remaining exception types.*

- #define **OSCL\_LAST\_CATCH(\_leave\_status)** else if (\_leave\_status!=OsclErrNone){OSCL\_-LEAVE(\_leave\_status);}

*Use this macro if OSCL\_CATCH\_ANY has not been used. It will mark the end of the catch block.*

## Typedefs

- typedef int32 **OsclLeaveCode**
- typedef int32 **OsclReturnCode**
- typedef void(\*) **OsclTrapOperation** )(OsclAny \*)

## Functions

- **OSCL\_IMPORT\_REF** bool **OSCL\_IsErrnoSupported ()**

*This function determines if a particular system saves the error number that occurs on a system call.*

- OSCL\_IMPORT\_REF int [OSCL\\_GetLastError\(\)](#)  
*This function returns the value of the system's global error number variable.*
- OSCL\_IMPORT\_REF bool [OSCL\\_SetLastError\(int newVal\)](#)  
*This function sets the last error code for the system.*
- OSCL\_IMPORT\_REF char \* [OSCL\\_StrError\(int errnum\)](#)  
*This function maps an error number to an error-message string.*

## 5.5.1 Define Documentation

### 5.5.1.1 #define \_PV\_TRAP(\_\_r, \_\_s)

**Value:**

```
__r=OsclErrNone; \
{ \
    OsclErrorTrapImp* __tr=OsclErrorTrapImp::Trap(); \
    if(!__tr){__s;}else{ \
        try{__s;} \
        catch(internalLeave __lv){ \
            __lv.a=__r=__tr->iLeave; } \
            __tr->UnTrap(); } \
    } \
}
```

### 5.5.1.2 #define \_PV\_TRAP\_NO\_TLS(\_\_trapimp, \_\_r, \_\_s)

**Value:**

```
__r=OsclErrNone; \
{ \
    OsclErrorTrapImp* __tr=OsclErrorTrapImp::TrapNoTls(__trapimp); \
    if(!__tr){__s;}else{ \
        try{__s;} \
        catch(internalLeave __lv){ \
            __lv.a=__r=__tr->iLeave; } \
            __tr->UnTrap(); } \
    } \
}
```

### 5.5.1.3 #define internalLeave (-1)

### 5.5.1.4 #define OSCL\_BAD\_ALLOC\_EXCEPTION\_CODE OsclErrNoMemory

### 5.5.1.5 #define OSCL\_CATCH(\_leave\_status, \_catch\_value, \_statements) else if (\_leave\_status!=OsclErrNone && \_leave\_status == \_catch\_value){\_statements;}

Use this macro to define a block of code for catching additional exception types.

OSCL\_FIRST\_CATCH can be used to catch one exception type. Then the OSCL\_CATCH macro can be used to catch each subsequent type. The catch block must end with OSCL\_LAST\_CATCH or OSCL\_CATCH\_ANY

**Parameters:**

*oscl\_leave\_status* is the result of any OSCL\_THROW

*exceptiontype* is the exception handled by this catch block

**5.5.1.6 #define OSCL\_CATCH\_ANY(\_leave\_status, \_statements) else if  
(\_leave\_status!=OsclErrNone){ \_statements;}**

Use this macro to call a function that will catch all remaining exception types.

**Parameters:**

*\_leave\_status*

*\_statements* is a statement or block of statements to handle all remaining exception types. This macro ends the try block.

**5.5.1.7 #define OSCL\_ERR\_NONE OsclErrNone**

For backward compatibility with old definitions

**5.5.1.8 #define OSCL\_FIRST\_CATCH(\_leave\_status, \_catch\_value, \_statements) if  
(\_leave\_status!=OsclErrNone && \_leave\_status == \_catch\_value){\_statements;}**

Use this macro to define a block of code that catches the first exception type thrown in the preceding try block.

**Parameters:**

*oscl\_leave\_status* is the leave code that was returned by OSCL\_THROW

*exceptiontype* is the exception handled by this catch block This macro MUST be used in conjunction with either OSCL\_LAST\_CATCH or OSCL\_CATCH\_ANY

**5.5.1.9 #define OSCL\_FIRST\_CATCH\_ANY(\_leave\_status, \_statements) if  
(\_leave\_status!=OsclErrNone) { \_statements; }**

This section defines the macros to be used in the catch block following the try block Use this macro to call a function that handles all exception types thrown in the preceding try block.

**Parameters:**

*\_leave\_status*

*\_statements* is a statement or block of statements that will catch all the exception types thrown by the preceding try block This is a standalone macro and should not be used with any of the macros above

**5.5.1.10 #define OSCL\_JUMP\_MAX\_JUMP\_MARKS OSCL\_MAX\_TRAP\_LEVELS**

**5.5.1.11 #define OSCL\_LAST\_CATCH(\_leave\_status) else if (\_leave\_status!=OsclErr-  
None){OSCL\_LEAVE(\_leave\_status);}**

Use this macro if OSCL\_CATCH\_ANY has not been used. It will mark the end of the catch block.

**Parameters:**

*\_leave\_status* will be propagated up the call stack. This macro will do an OSCL\_LEAVE if the leave has not been handled by the calls above. This macro ends the try block.

**5.5.1.12 #define OSCL\_LEAVE(\_leave\_status) OsclError::Leave(\_leave\_status)**

Use this macro to cause a Leave. It terminates the execution of the current active function.

It also tries to cleanup the items on the cleanup stack.

**Parameters:**

*oscl\_leave\_status* tells the cause for the Leave

**5.5.1.13 #define OSCL\_MAX\_TRAP\_LEVELS 20****5.5.1.14 #define OSCL\_TRAPSTACK\_POP() OsclError::Pop()****5.5.1.15 #define OSCL\_TRAPSTACK\_POPDEALLOC() OsclError::PopDealloc()****5.5.1.16 #define OSCL\_TRAPSTACK\_PUSH(a) OsclError::PushL(a)**

Cleanup Stack user macros

**5.5.1.17 #define OSCL\_TRY(\_leave\_status, \_statements) \_PV\_TRAP(\_leave\_status,\_statements)**

This macro will be used to set up a try block.

The try block identifies a block of code that might throw exceptions (or leave)

**Parameters:**

*oscl\_leave\_status* oscl\_leave\_status will receive the result of any OSCL\_LEAVE (which will get called from a OSCL\_THROW) on systems that do not support exception handling. This is unused on systems that do support exception handling

*statements* is a statement or block of statements that could throw exceptions and will be executed in the try block



5.5.1.18 #define OSCL\_TRY\_NO\_TLS(\_\_trapimp, \_leave\_status, \_statements)  
  \_\_PV\_TRAP\_NO\_TLS(\_\_trapimp,\_leave\_status,\_statements)

5.5.1.19 #define OsclErrAlreadyExists 106

5.5.1.20 #define OsclErrAlreadyInstalled 116

5.5.1.21 #define OsclErrArgument 104

5.5.1.22 #define OsclErrBadHandle 105

5.5.1.23 #define OsclErrBusy 107

5.5.1.24 #define OsclErrCancelled 102

5.5.1.25 #define OsclErrCorrupt 109

5.5.1.26 #define OsclErrGeneral 100

5.5.1.27 #define OsclErrInvalidState 113

5.5.1.28 #define OsclErrNoHandler 118

5.5.1.29 #define OsclErrNoMemory 101

5.5.1.30 #define OsclErrNone 0

5.5.1.31 #define OsclErrNoResources 114

5.5.1.32 #define OsclErrNotInstalled 115

5.5.1.33 #define OsclErrNotReady 108

5.5.1.34 #define OsclErrNotSupported 103

5.5.1.35 #define OsclErrOverflow 111

5.5.1.36 #define OsclErrSystemCallFailed 117

5.5.1.37 #define OsclErrThreadContextIncorrect 119

5.5.1.38 #define OsclErrTimeout 110

5.5.1.39 #define OsclErrUnderflow 112

5.5.1.40 #define OsclFailure -1

5.5.1.41 #define OsclPending 1

5.5.1.42 #define OsclSuccess 0

5.5.1.43 #define PVError\_DoLeave() internalLeave \_\_ilv; \_\_ilv.a=0; throw(\_\_ilv)

5.5.1.44 #define PVERROR\_IMP\_JUMPS

**5.5.1.45 #define PVERRORTRAP\_REGISTRY OsclTLSRegistry**

**5.5.1.46 #define PVERRORTRAP\_REGISTRY\_ID OSCL\_TLS\_ID\_PVERRORTRAP**

## 5.5.2 Typedef Documentation

**5.5.2.1 typedef int32 OsclLeaveCode**

Leave Codes

**5.5.2.2 typedef int32 OsclReturnCode**

Return Codes

**5.5.2.3 typedef void(\* OsclTrapOperation)(OsclAny\*)**

**OsclTrapItem** may be used in the cleanup stack when a custom cleanup operation is needed.

## 5.5.3 Function Documentation

**5.5.3.1 OSCL\_IMPORT\_REF int OSCL\_GetLastError ()**

This function returns the value of the system's global error number variable.

**Returns:**

Returns 0 for system's that do not have this functionality The value of the error number variable does not change until the user calls SetLastError or if another system call occurs that changes the value

Supported Platforms: Win32/wince, Unix Unsupported Platforms : Symbian

**5.5.3.2 OSCL\_IMPORT\_REF bool OSCL\_IsErrnoSupported ()**

This function determines if a particular system saves the error number that occurs on a system call.

**Returns:**

This method returns false on systems that do not save the error number that occurs on a system call in a global variable. Returns true for systems that do save the error number

**5.5.3.3 OSCL\_IMPORT\_REF bool OSCL\_SetLastError (int *newVal*)**

This function sets the last error code for the system.

**Parameters:**

*newVal* This value represents the new value for the global error number This method can be used to reset the error number after having retrieved it using GetLastError. Supported Platforms: Win32/wince, Unix Unsupported Platforms : Symbian

#### 5.5.3.4 OSCL\_IMPORT\_REF char\* OSCL\_StrError (int *errnum*)

This function maps an error number to an error-message string.

**Parameters:**

*errnum* This value represents the error number to map

**Returns:**

This method returns a pointer to a string containing the system error-message. It returns NULL for systems that do not have this functionality Supported Platforms: Win32/wince, Unix Unsupported Platforms : Symbian

## 5.6 OSCL IO

### Files

- file `oscl_dns.h`

*The file `oscl_socket.h` defines the OSCL DNS APIs.*

- file `oscl_file_cache.h`

*The file `oscl_file_cache.h` defines the class `OsclFileCache`.*

- file `oscl_file_dir_utils.h`

*The file `oscl_file_dir_utils.h` defines some unix-style directory ops.*

- file `oscl_file_find.h`

*The file `oscl_file_find.h` defines the class `Oscl_FileFind`.*

- file `oscl_file_handle.h`

*The file `oscl_file_handle.h` defines the class `Oscl_FileHandle`.*

- file `oscl_file_io.h`

*The file `oscl_file_io.h` defines the class `Oscl_File`. This is the public API to the basic file I/O operations.*

- file `oscl_file_native.h`

*The file `oscl_file_native.h` defines the class `OsclNativeFile`. This is the porting layer for basic file I/O operations.*

- file `oscl_file_server.h`

*The file `oscl_file_server.h` defines the class `Oscl_FileServer`. This is the porting layer for file server implementations.*

- file `oscl_file_stats.h`

*File stats class.*

- file `oscl_file_types.h`

*The file `oscl_file_types.h` defines some constants and types for file I/O implementations. Anything that needs to be shared across implementation modules can go here.*

- file `oscl_socket.h`

*The file `oscl_socket.h` defines the OSCL Socket APIs.*

### Data Structures

- class `Oscl_File`
- class `Oscl_FileFind`
- class `Oscl_FileServer`
- struct `oscl_fsstat`
- struct `oscl_stat_buf`
- class `OsclDNS`
- class `OsclDNSObserver`

- class OsclFileCache
- class OsclFileCacheBuffer
- class OsclFileHandle
- class OsclFileStats
- class OsclFileStatsItem
- class OsclNativeFile
- class OsclNativeFileParams
- class OsclSocketServ
- class OsclTCPSocket
- class OsclUDPSocket

## Defines

- #define TOsclFileOffsetInt32 int32
- #define OSCL\_FILE\_STATS\_LOGGER\_NODE "OsclFileStats"
- #define OSCL\_IO\_FILENAME\_MAXLEN 512
- #define OSCL\_IO\_EXTENSION\_MAXLEN 512
- #define OSCL\_FILE\_WCHAR\_PATH\_DELIMITER \_STRLIT("/")
 • #define OSCL\_FILE\_CHAR\_PATH\_DELIMITER \_STRLIT\_CHAR("/")

## Typedefs

- typedef oscl\_fsstat OSCL\_FSSTAT
- typedef oscl\_stat\_buf OSCL\_STAT\_BUF
- typedef FILE \* TOsclFileHandle

## Enumerations

- enum TPVDNSFx { EPVVDNSGetHostByName }
- enum TPVDNSEvent { EPVVDNSSuccess, EPVVDNSPending, EPVVDNSTimeout, EPVVDNSFailure, EPVVDNSCancel }
- enum OSCL\_FILEMGMT\_PERMS { OSCL\_FILEMGMT\_PERMS\_READ = 0x1, OSCL\_FILEMGMT\_PERMS\_WRITE = 0x2, OSCL\_FILEMGMT\_PERMS\_EXECUTE = 0x4 }
- enum OSCL\_FILEMGMT\_MODES { OSCL\_FILEMGMT\_MODE\_DIR = 0x1 }
- enum OSCL\_FILEMGMT\_ERR\_TYPE { OSCL\_FILEMGMT\_E\_OK = 0, OSCL\_FILEMGMT\_E\_PATH\_TOO\_LONG, OSCL\_FILEMGMT\_E\_PATH\_NOT\_FOUND, OSCL\_FILEMGMT\_E\_ALREADY\_EXISTS, OSCL\_FILEMGMT\_E\_NOT\_EMPTY, OSCL\_FILEMGMT\_E\_PERMISSION\_DENIED, OSCL\_FILEMGMT\_E\_NO\_MATCH, OSCL\_FILEMGMT\_E\_UNKNOWN, OSCL\_FILEMGMT\_E\_SYS\_SPECIFIC, OSCL\_FILEMGMT\_E\_NOT\_IMPLEMENTED }
- enum TOsclFileOp { EOscFileOp\_Open, EOscFileOp\_Close, EOscFileOp\_Read, EOscFileOp\_Write, EOscFileOp\_Seek, EOscFileOp\_Tell, EOscFileOp\_Size, EOscFileOp\_Flush, EOscFileOp\_EndOfFile, EOscFileOp\_NativeOpen, EOscFileOp\_NativeClose, EOscFileOp\_NativeRead, EOscFileOp\_NativeWrite, EOscFileOp\_NativeSeek, EOscFileOp\_NativeTell, EOscFileOp\_NativeSize, EOscFileOp\_NativeFlush, EOscFileOp\_NativeEndOfFile, EOscFileOp\_Last }

## Functions

- OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_getcwd (oscl\_wchar \*path, uint32 size)
- OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_getcwd (char \*path, uint32 size)
- OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_stat (const oscl\_wchar \*path, OSCL\_STAT\_BUF \*statbuf)
- OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_stat (const char \*path, OSCL\_STAT\_BUF \*statbuf)
- OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_mkdir (const oscl\_wchar \*path)
- OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_mkdir (const char \*path)
- OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_rmdir (const oscl\_wchar \*path)
- OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_rmdir (const char \*path)
- OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_chdir (const oscl\_wchar \*path)
- OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_chdir (const char \*path)
- OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_rename (const oscl\_wchar \*oldpath, const oscl\_wchar \*newpath)
- OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_rename (const char \*oldpath, const char \*newpath)
- OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_statsfs (OSCL\_FSSTAT \*stats, const char \*path)
- OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_statsfs (OSCL\_FSSTAT \*stats, const oscl\_wchar \*path)

### 5.6.1 Define Documentation

- 5.6.1.1 #define OSCL\_FILE\_CHAR\_PATH\_DELIMITER \_STRLIT\_CHAR("/")
- 5.6.1.2 #define OSCL\_FILE\_STATS\_LOGGER\_NODE "OsclFileStats"
- 5.6.1.3 #define OSCL\_FILE\_WCHAR\_PATH\_DELIMITER \_STRLIT("/")
- 5.6.1.4 #define OSCL\_IO\_EXTENSION\_MAXLEN 512
- 5.6.1.5 #define OSCL\_IO\_FILENAME\_MAXLEN 512
- 5.6.1.6 #define TOsclFileOffsetInt32 int32

### 5.6.2 Typedef Documentation

- 5.6.2.1 typedef struct oscl\_fsstat OSCL\_FSSTAT
- 5.6.2.2 typedef struct oscl\_stat\_buf OSCL\_STAT\_BUF
- 5.6.2.3 typedef FILE\* TOsclFileHandle

TOsclFileHandle is an OS-native file handle type. With a class-based file API such as Symbian, a class ref is used as a file handle. For most ANSI-style file APIs, a file pointer is used as a file handle.

### 5.6.3 Enumeration Type Documentation

#### 5.6.3.1 enum OSCL\_FILEMGMT\_ERR\_TYPE

Enumeration values:

- `OSCL_FILEMGMT_E_OK`
- `OSCL_FILEMGMT_E_PATH_TOO_LONG`
- `OSCL_FILEMGMT_E_PATH_NOT_FOUND`
- `OSCL_FILEMGMT_E_ALREADY_EXISTS`
- `OSCL_FILEMGMT_E_NOT_EMPTY`
- `OSCL_FILEMGMT_E_PERMISSION_DENIED`
- `OSCL_FILEMGMT_E_NO_MATCH`
- `OSCL_FILEMGMT_E_UNKNOWN`
- `OSCL_FILEMGMT_E_SYS_SPECIFIC`
- `OSCL_FILEMGMT_E_NOT_IMPLEMENTED`

#### 5.6.3.2 enum OSCL\_FILEMGMT\_MODES

Enumeration values:

- `OSCL_FILEMGMT_MODE_DIR`

#### 5.6.3.3 enum OSCL\_FILEMGMT\_PERMS

Enumeration values:

- `OSCL_FILEMGMT_PERMS_READ`
- `OSCL_FILEMGMT_PERMS_WRITE`
- `OSCL_FILEMGMT_PERMS_EXECUTE`

#### 5.6.3.4 enum TOsclFileOp

Enumeration values:

- `EOsclFileOp_Open`
- `EOsclFileOp_Close`
- `EOsclFileOp_Read`
- `EOsclFileOp_Write`
- `EOsclFileOp_Seek`
- `EOsclFileOp_Tell`
- `EOsclFileOp_Size`
- `EOsclFileOp_Flush`
- `EOsclFileOp_EndOfFile`
- `EOsclFileOp_NativeOpen`
- `EOsclFileOp_NativeClose`
- `EOsclFileOp_NativeRead`

**EOsclFileOp\_NativeWrite**  
**EOsclFileOp\_NativeSeek**  
**EOsclFileOp\_NativeTell**  
**EOsclFileOp\_NativeSize**  
**EOsclFileOp\_NativeFlush**  
**EOsclFileOp\_NativeEndOfFile**  
**EOsclFileOp\_Last**

#### 5.6.3.5 enum TPVDNSEvent

**Enumeration values:**

**EPVDNSSuccess**  
**EPVDNSPending**  
**EPVDNSTimeout**  
**EPVDNSFailure**  
**EPVDNSCancel**

#### 5.6.3.6 enum TPVDNSFxn

**Enumeration values:**

**EPVDNSGetHostByName**

### 5.6.4 Function Documentation

#### 5.6.4.1 OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_chdir (const char \**path*)

oscl\_chdir changes the current directory to the path given

**Parameters:**

*character* path the full path of the directory to change to.

**Returns:**

OSCL\_FILEMGMT\_ERR\_TYPE, see enumeration for this type.

#### 5.6.4.2 OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_chdir (const oscl\_wchar \**path*)

oscl\_chdir changes the current directory to the path given

**Parameters:**

*wide* character path the full path of the directory to change to.

**Returns:**

OSCL\_FILEMGMT\_ERR\_TYPE, see enumeration for this type.

**5.6.4.3 OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_getcwd (char \*path, uint32 size)**

oscl\_getcwd function can be used to determine the full path name of the current directory.

**Parameters:**

*pointer* to character buffer to receive the current directory  
*size* size of buffer in characters

**Returns:**

OSCL\_FILEMGMT\_ERR\_TYPE, see enumeration for this type.

**5.6.4.4 OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_getcwd (oscl\_wchar \*path, uint32 size)**

oscl\_getcwd function can be used to determine the full path name of the current directory.

**Parameters:**

*pointer* to wide character buffer to receive the current directory  
*size* size of buffer in wide characters

**Returns:**

OSCL\_FILEMGMT\_ERR\_TYPE, see enumeration for this type.

**5.6.4.5 OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_mkdir (const char \*path)**

oscl\_mkdir function creates a directory in the path given

**Parameters:**

*character* path the full path of the directory to create. if parts of the path do not exist the function will fail

**Returns:**

OSCL\_FILEMGMT\_ERR\_TYPE, see enumeration for this type.

**5.6.4.6 OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_mkdir (const oscl\_wchar \*path)**

oscl\_mkdir function creates a directory in the path given

**Parameters:**

*wide* character path the full path of the directory to create. if parts of the path do not exist the function will fail

**Returns:**

OSCL\_FILEMGMT\_ERR\_TYPE, see enumeration for this type.

**5.6.4.7 OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_rename (const char \*  
*oldpath*, const char \**newpath*)**

oscl\_rmdir removes an empty directory in the path given

**Parameters:**

*character* path the full path of the directory to remove.

**Returns:**

OSCL\_FILEMGMT\_ERR\_TYPE, see enumeration for this type.

**5.6.4.8 OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_rename (const oscl\_wchar  
*\* oldpath*, const oscl\_wchar \**newpath*)**

oscl\_rename function renames a file or directory

**Parameters:**

*wide* character path the full path of the file or directory to rename.

*wide* character path the full path the new name for the directory

**Returns:**

OSCL\_FILEMGMT\_ERR\_TYPE, see enumeration for this type.

**5.6.4.9 OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_rmdir (const char \**path*)**

oscl\_rmdir removes an empty directory in the path given

**Parameters:**

*character* path the full path of the directory to remove.

**Returns:**

OSCL\_FILEMGMT\_ERR\_TYPE, see enumeration for this type.

**5.6.4.10 OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_rmdir (const oscl\_wchar \*  
*path*)**

oscl\_rmdir function removes and empty directory in the path given

**Parameters:**

*wide* character path the full path of the directory to remove.

**Returns:**

OSCL\_FILEMGMT\_ERR\_TYPE, see enumeration for this type.

**5.6.4.11 OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_stat (const char \*path,  
OSCL\_STAT\_BUF \*statbuf)**

oscl\_stat function can be used to determine the size of a file in addition to whether the file exists or not

**Parameters:**

*character* path the full path of the file to stat.

**Returns:**

OSCL\_FILEMGMT\_ERR\_TYPE, see enumeration for this type.

**5.6.4.12 OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_stat (const oscl\_wchar \*  
path, OSCL\_STAT\_BUF \*statbuf)**

oscl\_stat function can be used to determine the size of a file in addition to whether the file exists or not

**Parameters:**

*wide* character path the full path of the file to stat.

**Returns:**

OSCL\_FILEMGMT\_ERR\_TYPE, see enumeration for this type.

**5.6.4.13 OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_statfs (OSCL\_FSSTAT \*  
stats, const oscl\_wchar \*path)**

Oscl\_StatFS function populates a general structure describing free space available on a filesystem

**Parameters:**

*stats* pointer to structure to hold information

*path* located in desired filesystem (utf8) Note: If the OS does not support a particular field in the structure, it is set to -1.

**Returns:**

OSCL\_FILEMGMT\_ERR\_TYPE, see enumeration for this type.

**5.6.4.14 OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_statfs (OSCL\_FSSTAT \*  
stats, const char \*path)**

Oscl\_StatFS function populates a general structure describing free space available on a filesystem

**Parameters:**

*stats* pointer to structure to hold information

*path* located in desired filesystem (utf8) Note: If the OS does not support a particular field in the structure, it is set to -1.

**Returns:**

OSCL\_FILEMGMT\_ERR\_TYPE, see enumeration for this type.

## 5.7 OSCL Proc

### Files

- file [oscl\\_aostatus.h](#)  
*Some basic types used with active objects.*
- file [oscl\\_double\\_list.h](#)  
*Internal use types for scheduler.*
- file [oscl\\_scheduler\\_ao.h](#)  
*Oscl Scheduler user execution object classes.*
- file [oscl\\_scheduler\\_aobase.h](#)  
*Oscl Scheduler internal active object classes.*
- file [oscl\\_scheduler\\_readyq.h](#)  
*ready q types for oscl scheduler*
- file [oscl\\_scheduler\\_threadcontext.h](#)  
*Thread context functions needed by oscl scheduler.*
- file [oscl\\_scheduler\\_tuneables.h](#)  
*Tuneable settings for Oscl Scheduler.*
- file [oscl\\_scheduler\\_types.h](#)  
*Scheduler common types include file.*

### Data Structures

- class [OsclActiveObject](#)
- class [OsclAOStatus](#)
- class [OsclDoubleLink](#)
- class [OsclDoubleList](#)
- class [OsclDoubleListBase](#)
- class [OsclDoubleRunner](#)
- class [OsclExecScheduler](#)
- class [OsclExecSchedulerBase](#)
- class [OsclExecSchedulerCommonBase](#)
- class [OsclPriorityLink](#)
- class [OsclPriorityList](#)
- class [OsclReadyAlloc](#)
- class [OsclReadyCompare](#)
- class [OsclReadyQ](#)
- class [OsclScheduler](#)
- class [OsclSchedulerObserver](#)
- class [OsclTimerCompare](#)
- class [OsclTimerObject](#)

- class OsclTimerQ
- class PVActiveBase
- class PVActiveStats
- class PVSchedulerStopper
- class PVThreadContext
- class TReadyQueLink

## Defines

- #define QUE\_ITER\_BEGIN(\_type, \_qname)
- #define QUE\_ITER\_END(\_qname)
- #define PVSCHEDNAMELEN 30
- #define OSCL\_ZEROIZE(ptr, size) oscl\_memset(ptr, 0, size)
- #define PVEXECNAMELEN 30
- #define PV\_SCHED\_ENABLE\_AO\_STATS 1
- #define PV\_SCHED\_ENABLE\_LOOP\_STATS 0
- #define PV\_SCHED\_ENABLE\_PERF\_LOGGING 1
- #define PV\_SCHED\_ENABLE\_THREAD\_CONTEXT\_CHECKS 1
- #define PV\_SCHED\_LOG\_Q 0
- #define PV\_SCHED\_CHECK\_Q 0
- #define PV\_SCHED\_FAIR\_SCHEDULING 1
- #define OSCL\_PERF\_SUMMARY\_LOGGING 0

## Typedefs

- typedef PVActiveBase \* TOsclReady

## Enumerations

- enum TPVThreadContext { EPVThreadContext\_InThread, EPVThreadContext\_OsclThread, EPVThreadContext\_NonOsclThread, EPVThreadContext\_Undetermined }

## Functions

- template<class T, class S> T \* OsclPtrAdd (T \*aPtr, S aVal)
- template<class T, class S> T \* OsclPtrSub (T \*aPtr, S aVal)

## Variables

- const int32 OSCL\_REQUEST\_ERR\_NONE = 0
- const int32 OSCL\_REQUEST\_PENDING = (-0x7fffffff)
- const int32 OSCL\_REQUEST\_ERR\_CANCEL = (-1)
- const int32 OSCL\_REQUEST\_ERR\_GENERAL = (-2)

## 5.7.1 Define Documentation

**5.7.1.1 #define OSCL\_PERF\_SUMMARY\_LOGGING 0**

**5.7.1.2 #define OSCL\_ZEROIZE(ptr, size) oscl\_memset(ptr, 0, size)**

This file defines the [PVActiveBase](#) class, which is a common base for All PV ExecObjs on all platforms.

**5.7.1.3 #define PV\_SCHED\_CHECK\_Q 0**

**5.7.1.4 #define PV\_SCHED\_ENABLE\_AO\_STATS 1**

**5.7.1.5 #define PV\_SCHED\_ENABLE\_LOOP\_STATS 0**

**5.7.1.6 #define PV\_SCHED\_ENABLE\_PERF\_LOGGING 1**

**5.7.1.7 #define PV\_SCHED\_ENABLE\_THREAD\_CONTEXT\_CHECKS 1**

**5.7.1.8 #define PV\_SCHED\_FAIR\_SCHEDULING 1**

**5.7.1.9 #define PV\_SCHED\_LOG\_Q 0**

**5.7.1.10 #define PVEEXECNAMELEN 30**

**5.7.1.11 #define PVSCEDNAMELEN 30**

PV Scheduler class

**5.7.1.12 #define QUE\_ITER\_BEGIN(\_type, \_qname)**

**Value:**

```
if (!_qname.IsEmpty())\
{\
    OsclDoubleRunner <_type> iter(_qname);\
    _type *item;\
    for (iter.SetToHead(); ;iter++)\
    {\
        item=iter;\
```

**5.7.1.13 #define QUE\_ITER\_END(\_qname)**

**Value:**

```
if (_qname.IsTail(item))\
    break;\
}\
```

## 5.7.2 Typedef Documentation

### 5.7.2.1 `typedef PVActiveBase* TOsclReady`

## 5.7.3 Enumeration Type Documentation

### 5.7.3.1 `enum TPVThreadContext`

Thread context type

Enumeration values:

`EPVThreadContext_InThread`  
`EPVThreadContext_OsclThread`  
`EPVThreadContext_NonOsclThread`  
`EPVThreadContext_Undetermined`

## 5.7.4 Function Documentation

### 5.7.4.1 `template<class T, class S> T* OsclPtrAdd (T * aPtr, S aVal) [inline]`

### 5.7.4.2 `template<class T, class S> T* OsclPtrSub (T * aPtr, S aVal) [inline]`

## 5.7.5 Variable Documentation

### 5.7.5.1 `const int32 OSCL_REQUEST_ERR_CANCEL = (-1)`

### 5.7.5.2 `const int32 OSCL_REQUEST_ERR_GENERAL = (-2)`

### 5.7.5.3 `const int32 OSCL_REQUEST_ERR_NONE = 0`

### 5.7.5.4 `const int32 OSCL_REQUEST_PENDING = (-0x7fffffff)`

## 5.8 OSCL Init

### Files

- file [oscl\\_init.h](#)  
*Global oscl initialization.*

### Data Structures

- class [OsclInit](#)
- class [OsclSelect](#)

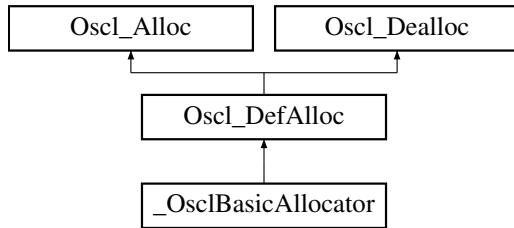
# Chapter 6

## oscl Data Structure Documentation

### 6.1 \_OsclBasicAllocator Class Reference

```
#include <oscl_base_alloc.h>
```

Inheritance diagram for \_OsclBasicAllocator::



#### Public Methods

- [OsclAny \\* allocate \(const uint32 size\)](#)
- [void deallocate \(OsclAny \\*p\)](#)
- [virtual ~\\_OsclBasicAllocator \(\)](#)

#### 6.1.1 Detailed Description

A basic allocator that does not rely on other modules. There is no memory auditing or exception generation.

Note: this allocator is for internal use by Oscl code only. Higher level code should use [OsclMemAllocator](#) defined in "[oscl\\_mem.h](#)".

## 6.1.2 Constructor & Destructor Documentation

6.1.2.1 `virtual _OsclBasicAllocator::~_OsclBasicAllocator () [inline, virtual]`

## 6.1.3 Member Function Documentation

6.1.3.1 `OsclAny* _OsclBasicAllocator::allocate (const uint32 size) [inline, virtual]`

Implements [Oscl\\_DefAlloc](#).

6.1.3.2 `void _OsclBasicAllocator::deallocate (OsclAny *p) [inline, virtual]`

Implements [Oscl\\_DefAlloc](#).

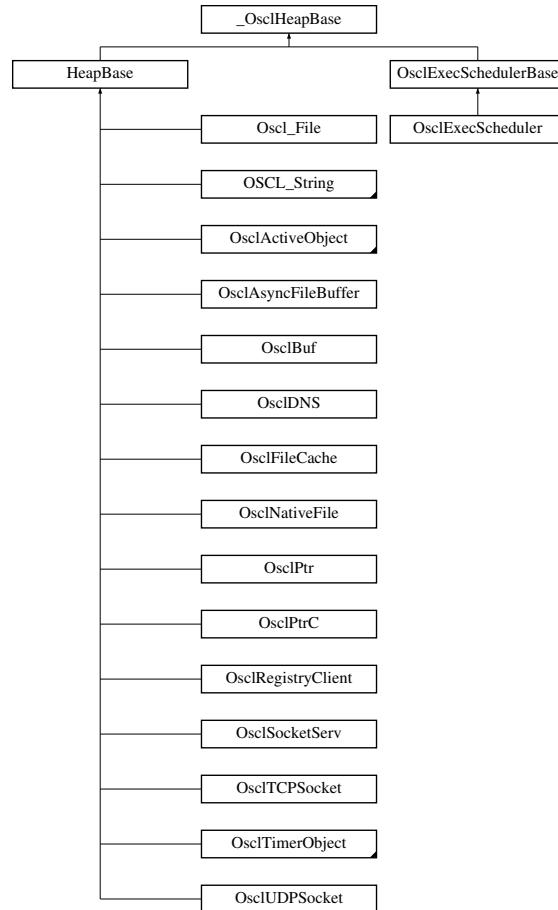
The documentation for this class was generated from the following file:

- [oscl\\_base\\_alloc.h](#)

## 6.2 \_OsclHeapBase Class Reference

```
#include <oscl_heapbase.h>
```

Inheritance diagram for \_OsclHeapBase::



### Public Methods

- virtual ~\_OsclHeapBase ()

### Protected Methods

- [\\_OsclHeapBase \(\)](#)
- [\\_OsclHeapBase \(const \\_OsclHeapBase &\)](#)

### Friends

- class [PVCleanupStack](#)

### 6.2.1 Detailed Description

\_OsclHeapBase is used as the base for cleanup stack items with virtual destructor.

### 6.2.2 Constructor & Destructor Documentation

6.2.2.1 `virtual _OsclHeapBase::~_OsclHeapBase () [inline, virtual]`

6.2.2.2 `_OsclHeapBase::_OsclHeapBase () [inline, protected]`

6.2.2.3 `_OsclHeapBase::_OsclHeapBase (const _OsclHeapBase &) [inline, protected]`

### 6.2.3 Friends And Related Function Documentation

6.2.3.1 `friend class PVCleanupStack [friend]`

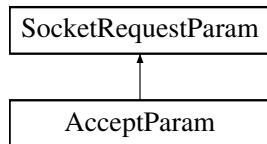
The documentation for this class was generated from the following file:

- [oscl\\_heapbase.h](#)

## 6.3 AcceptParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for AcceptParam::



### Public Methods

- [AcceptParam \(OsclSocketI &aBlankSocket\)](#)

### Data Fields

- [OsclSocketI \\* iBlankSocket](#)

#### 6.3.1 Constructor & Destructor Documentation

6.3.1.1 [AcceptParam::AcceptParam \(OsclSocketI & aBlankSocket\) \[inline\]](#)

#### 6.3.2 Field Documentation

6.3.2.1 [OsclSocketI\\* AcceptParam::iBlankSocket](#)

The documentation for this class was generated from the following file:

- [oscl\\_socket\\_request.h](#)

## 6.4 allocator Class Reference

```
#include <oscl_mem_mempool.h>
```

### 6.4.1 Detailed Description

A memory allocator class which allocates and deallocates from a fixed size memory pool; The memory pool is a multiple of fixed chunk size and does not grow. All allocation size must be the same as this chunk size.

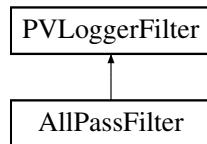
The documentation for this class was generated from the following file:

- [oscl\\_mem\\_mempool.h](#)

## 6.5 AllPassFilter Class Reference

```
#include <pvlogger_accessories.h>
```

Inheritance diagram for AllPassFilter::



### Public Types

- [typedef PVLoggerFilter::message\\_id\\_type message\\_id\\_type](#)
- [typedef PVLoggerFilter::log\\_level\\_type log\\_level\\_type](#)
- [typedef PVLoggerFilter::filter\\_status\\_type filter\\_status\\_type](#)

### Public Methods

- [AllPassFilter \(\)](#)
- [virtual ~AllPassFilter \(\)](#)
- [filter\\_status\\_type FilterString \(char \\*tag, message\\_id\\_type msgID, log\\_level\\_type level\)](#)
- [filter\\_status\\_type FilterOpaqueMessge \(char \\*tag, message\\_id\\_type msgID, log\\_level\\_type level\)](#)

#### 6.5.1 Detailed Description

Example filter that allows all messages to be logged.

#### 6.5.2 Member Typedef Documentation

##### 6.5.2.1 [typedef PVLoggerFilter::filter\\_status\\_type AllPassFilter::filter\\_status\\_type](#)

Reimplemented from [PVLoggerFilter](#).

##### 6.5.2.2 [typedef PVLoggerFilter::log\\_level\\_type AllPassFilter::log\\_level\\_type](#)

Reimplemented from [PVLoggerFilter](#).

##### 6.5.2.3 [typedef PVLoggerFilter::message\\_id\\_type AllPassFilter::message\\_id\\_type](#)

Reimplemented from [PVLoggerFilter](#).

### 6.5.3 Constructor & Destructor Documentation

**6.5.3.1** `AllPassFilter::AllPassFilter () [inline]`

**6.5.3.2** `virtual AllPassFilter::~AllPassFilter () [inline, virtual]`

### 6.5.4 Member Function Documentation

**6.5.4.1** `filter_status_type AllPassFilter::FilterOpaqueMessge (char * tag, message_id_type msgID, log_level_type level) [inline, virtual]`

Implements [PVLoggerFilter](#).

**6.5.4.2** `filter_status_type AllPassFilter::FilterString (char * tag, message_id_type msgID, log_level_type level) [inline, virtual]`

Implements [PVLoggerFilter](#).

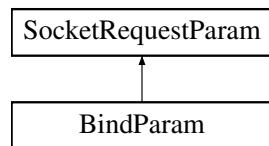
The documentation for this class was generated from the following file:

- [pvlogger\\_accessories.h](#)

## 6.6 BindParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for BindParam::



### Public Methods

- [BindParam \(OsclNetworkAddress &anAddr\)](#)

### Data Fields

- [OsclNetworkAddress iAddr](#)

#### 6.6.1 Constructor & Destructor Documentation

6.6.1.1 [BindParam::BindParam \(OsclNetworkAddress & anAddr\) \[inline\]](#)

#### 6.6.2 Field Documentation

6.6.2.1 [OsclNetworkAddress BindParam::iAddr](#)

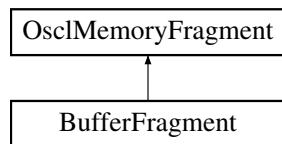
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_request.h](#)

## 6.7 BufferFragment Class Reference

```
#include <oscl_media_data.h>
```

Inheritance diagram for BufferFragment::



The documentation for this class was generated from the following file:

- [oscl\\_media\\_data.h](#)

## 6.8 BufferMgr Class Reference

```
#include <oscl_media_data.h>
```

### Public Methods

- virtual void [BufferReleased](#) (void \*ptr, [BufferState](#) \*state=NULL)=0
- virtual [~BufferMgr](#) ()

#### 6.8.1 Constructor & Destructor Documentation

**6.8.1.1 virtual BufferMgr::~BufferMgr () [inline, virtual]**

#### 6.8.2 Member Function Documentation

**6.8.2.1 virtual void BufferMgr::BufferReleased (void \*ptr, BufferState \* state = NULL) [pure virtual]**

The documentation for this class was generated from the following file:

- [oscl\\_media\\_data.h](#)

## 6.9 BufferState Class Reference

```
#include <oscl_media_data.h>
```

### Public Methods

- [BufferState \(BufferFreeFuncPtr the\\_free\\_function, void \\*bufptr=0\)](#)
- [BufferState \(BufferMgr \\*the\\_buf\\_mgr=0, void \\*bufptr=0\)](#)
- [void increment\\_refcnt \(\)](#)
- [void decrement\\_refcnt \(\)](#)
- [void bind \(void \\*in\\_ptr, BufferFreeFuncPtr in\\_free\\_function\)](#)
- [void bind \(void \\*in\\_ptr, BufferMgr \\*in\\_buf\\_mgr\)](#)
- [void \\* get\\_ptr \(\)](#)
- [int32 getRefCount \(\)](#)
- [BufferFreeFuncPtr get\\_free\\_function \(\)](#)
- [BufferMgr \\* get\\_buf\\_mgr \(\)](#)
- [void reset \(\)](#)

#### 6.9.1 Constructor & Destructor Documentation

**6.9.1.1 [BufferState::BufferState \(BufferFreeFuncPtr the\\_free\\_function, void \\* bufptr = 0\)](#) [inline]**

**6.9.1.2 [BufferState::BufferState \(BufferMgr \\* the\\_buf\\_mgr = 0, void \\* bufptr = 0\)](#) [inline]**

#### 6.9.2 Member Function Documentation

**6.9.2.1 [void BufferState::bind \(void \\* in\\_ptr, BufferMgr \\* in\\_buf\\_mgr\)](#) [inline]**

**6.9.2.2 [void BufferState::bind \(void \\* in\\_ptr, BufferFreeFuncPtr in\\_free\\_function\)](#) [inline]**

**6.9.2.3 [void BufferState::decrement\\_refcnt \(\)](#) [inline]**

**6.9.2.4 [BufferMgr\\* BufferState::get\\_buf\\_mgr \(\)](#) [inline]**

**6.9.2.5 [BufferFreeFuncPtr BufferState::get\\_free\\_function \(\)](#) [inline]**

**6.9.2.6 [void\\* BufferState::get\\_ptr \(\)](#) [inline]**

**6.9.2.7 [int32 BufferState::getRefCount \(\)](#) [inline]**

**6.9.2.8 [void BufferState::increment\\_refcnt \(\)](#) [inline]**

**6.9.2.9 [void BufferState::reset \(\)](#) [inline]**

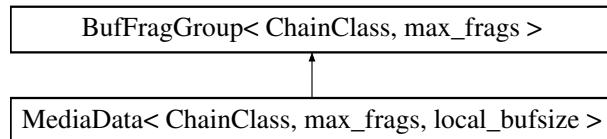
The documentation for this class was generated from the following file:

- [oscl\\_media\\_data.h](#)

## 6.10 BufFragGroup< ChainClass, max\_frags > Class Template Reference

```
#include <oscl_media_data.h>
```

Inheritance diagram for BufFragGroup< ChainClass, max\_frags >::



### Public Methods

- [BufFragGroup \(\)](#)
- virtual [~BufFragGroup \(\)](#)
- int32 [GetMaxFrags \(\) const](#)
- int32 [GetNumFrags \(\) const](#)
- uint32 [GetLength \(\) const](#)
- [BufferFragment \\* GetFragment \(const int32 idx\)](#)
- [BufferState \\* GetBufferState \(const int32 idx\)](#)
- void [AppendNext \(ChainClass \\*next\\_ptr\)](#)
- ChainClass \* [GetNext \(\) const](#)

### Protected Methods

- virtual void [Clear \(\)](#)
- [BufFragStatusClass::status\\_t AddFragment \(const BufferFragment &frag, BufferState \\*in\\_buffer\\_state, int32 location\\_offset=max\\_frags\)](#)

### Protected Attributes

- [BufferFragment fragments \[max\\_frags\]](#)
- [BufferState \\* buffer\\_states \[max\\_frags\]](#)
- [ChainClass \\* next](#)
- uint32 [num.fragments](#)
- uint32 [length](#)

```
template<class ChainClass, uint32 max_frags> class BufFragGroup< ChainClass, max_frags >
```

### 6.10.1 Constructor & Destructor Documentation

6.10.1.1 `template<class ChainClass, uint32 max_frags> BufFragGroup< ChainClass, max_frags >::BufFragGroup () [inline]`

6.10.1.2 `template<class ChainClass, uint32 max_frags> virtual BufFragGroup< ChainClass, max_frags >::~BufFragGroup () [inline, virtual]`

### 6.10.2 Member Function Documentation

6.10.2.1 `template<class ChainClass, uint32 max_frags> BufFragStatusClass::status\_t BufFragGroup< ChainClass, max_frags >::AddFragment (const BufferFragment & frag, BufferState * in_buffer_state, int32 location_offset = max_frags) [inline, protected]`

6.10.2.2 `template<class ChainClass, uint32 max_frags> void BufFragGroup< ChainClass, max_frags >::AppendNext (ChainClass * next_ptr) [inline]`

6.10.2.3 `template<class ChainClass, uint32 max_frags> virtual void BufFragGroup< ChainClass, max_frags >::Clear () [inline, protected, virtual]`

Reimplemented in [MediaData< ChainClass, max\\_frags, local\\_bufsize >](#).

**6.10.2.4** template<class ChainClass, uint32 max\_frags> uint32 BufFragGroup< ChainClass, max\_frags >::GetLength () const [inline]

**6.10.2.5** template<class ChainClass, uint32 max\_frags> int32 BufFragGroup< ChainClass, max\_frags >::GetMaxFrags () const [inline]

**6.10.2.6** template<class ChainClass, uint32 max\_frags> ChainClass\* BufFragGroup< ChainClass, max\_frags >::GetNext () const [inline]

**6.10.2.7** template<class ChainClass, uint32 max\_frags> int32 BufFragGroup< ChainClass, max\_frags >::GetNumFrags () const [inline]

### 6.10.3 Field Documentation

**6.10.3.1** template<class ChainClass, uint32 max\_frags> BufferState\* BufFragGroup< ChainClass, max\_frags >::buffer\_states[max\_frags] [protected]

**6.10.3.2** template<class ChainClass, uint32 max\_frags> BufferFragment BufFragGroup< ChainClass, max\_frags >::fragments[max\_frags] [protected]

**6.10.3.3** template<class ChainClass, uint32 max\_frags> uint32 BufFragGroup< ChainClass, max\_frags >::length [protected]

**6.10.3.4** template<class ChainClass, uint32 max\_frags> ChainClass\* BufFragGroup< ChainClass, max\_frags >::next [protected]

**6.10.3.5** template<class ChainClass, uint32 max\_frags> uint32 BufFragGroup< ChainClass, max\_frags >::num\_fragments [protected]

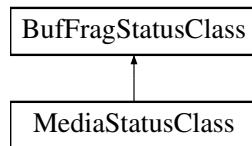
The documentation for this class was generated from the following file:

- [oscl\\_media\\_data.h](#)

## 6.11 BufFragStatusClass Class Reference

```
#include <oscl_media_status.h>
```

Inheritance diagram for BufFragStatusClass::



### Public Types

- enum `status_t` { `BFG_SUCCESS` = 0, `TOO_MANY_FRAGS` = 1, `NOT_ENOUGH_SPACE` = 2, `EMPTY_FRAGMENT` = 3, `NULL_INPUT` = 4, `FIXED_FRAG_LOC_FULL` = 5, `INTERNAL_ERROR`, `INVALID_ID` }

#### 6.11.1 Member Enumeration Documentation

##### 6.11.1.1 enum BufFragStatusClass::status\_t

Enumeration values:

`BFG_SUCCESS`  
`TOO_MANY_FRAGS`  
`NOT_ENOUGH_SPACE`  
`EMPTY_FRAGMENT`  
`NULL_INPUT`  
`FIXED_FRAG_LOC_FULL`  
`INTERNAL_ERROR`  
`INVALID_ID`

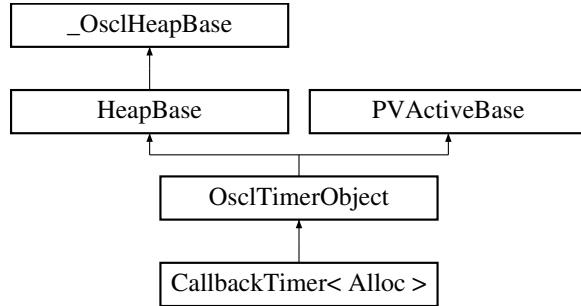
The documentation for this class was generated from the following file:

- [oscl\\_media\\_status.h](#)

## 6.12 CallbackTimer< Alloc > Class Template Reference

```
#include <oscl_timer.h>
```

Inheritance diagram for CallbackTimer< Alloc >::



### Public Methods

- [CallbackTimer \(CallbackTimerObserver &aContainer, const char \\*name, int32 aPriority=OsclActiveObject::EPriorityNominal\)](#)
- [~CallbackTimer \(\)](#)
- [void Run \(\)](#)

```
template<class Alloc> class CallbackTimer< Alloc >
```

#### 6.12.1 Constructor & Destructor Documentation

**6.12.1.1 template<class Alloc> CallbackTimer< Alloc >::CallbackTimer (CallbackTimerObserver & aContainer, const char \* name, int32 aPriority = OsclActiveObject::EPriorityNominal) [inline]**

**6.12.1.2 template<class Alloc> CallbackTimer< Alloc >::~CallbackTimer () [inline]**

#### 6.12.2 Member Function Documentation

**6.12.2.1 template<class Alloc> void CallbackTimer< Alloc >::Run () [inline, virtual]**

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's WaitForAnyRequest() function completes.

Before calling this active object's [Run\(\)](#) function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request
2. marked this active object's request as complete (i.e. the request is no longer outstanding)

[Run\(\)](#) runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls ExecError() to handle the leave.

Note that once the active scheduler's Start() function has been called, all user code is run under one of the program's active object's [Run\(\)](#) or [RunError\(\)](#) functions.

Implements [PVActiveBase](#).

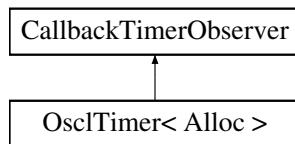
The documentation for this class was generated from the following file:

- [oscl\\_timer.h](#)

## 6.13 CallbackTimerObserver Class Reference

```
#include <oscl_timer.h>
```

Inheritance diagram for CallbackTimerObserver::



### Public Methods

- virtual void [TimerBaseElapsed \(\)=0](#)
- virtual [~CallbackTimerObserver \(\)](#)

#### 6.13.1 Constructor & Destructor Documentation

**6.13.1.1 virtual CallbackTimerObserver::~CallbackTimerObserver () [inline, virtual]**

#### 6.13.2 Member Function Documentation

**6.13.2.1 virtual void CallbackTimerObserver::TimerBaseElapsed () [pure virtual]**

Implemented in [OsclTimer< Alloc >](#).

The documentation for this class was generated from the following file:

- [oscl\\_timer.h](#)

## 6.14 CFastRep Class Reference

```
#include <oscl_string_rep.h>
```

### Public Methods

- [CFastRep \(\)](#)
- OSCL\_IMPORT\_REF void [set\\_w](#) (char \*cp, uint32 len, uint32 maxlen)
- OSCL\_IMPORT\_REF void [set\\_w \(oscl\\_wchar](#) \*cp, uint32 len, uint32 maxlen)
- OSCL\_IMPORT\_REF void [set\\_r](#) (const char \*cp, uint32 len)
- OSCL\_IMPORT\_REF void [set\\_r \(const oscl\\_wchar](#) \*cp, uint32 len)
- OSCL\_IMPORT\_REF void [append](#) (const char \*cp, uint32 len)
- OSCL\_IMPORT\_REF void [append \(const oscl\\_wchar](#) \*cp, uint32 len)

### Data Fields

- uint32 [maxsize](#)
- uint32 [size](#)
- [OsclAny \\* buffer](#)
- bool [writable](#)

#### 6.14.1 Detailed Description

For internal use only– fast string representation

## 6.14.2 Constructor & Destructor Documentation

6.14.2.1 `CFastRep::CFastRep () [inline]`

## 6.14.3 Member Function Documentation

6.14.3.1 `OSCL_IMPORT_REF void CFastRep::append (const oscl_wchar * cp, uint32 len)`

6.14.3.2 `OSCL_IMPORT_REF void CFastRep::append (const char * cp, uint32 len)`

6.14.3.3 `OSCL_IMPORT_REF void CFastRep::set_r (const oscl_wchar * cp, uint32 len)`

6.14.3.4 `OSCL_IMPORT_REF void CFastRep::set_r (const char * cp, uint32 len)`

6.14.3.5 `OSCL_IMPORT_REF void CFastRep::set_w (oscl_wchar * cp, uint32 len, uint32 maxlen)`

6.14.3.6 `OSCL_IMPORT_REF void CFastRep::set_w (char * cp, uint32 len, uint32 maxlen)`

## 6.14.4 Field Documentation

6.14.4.1 `OsclAny* CFastRep::buffer`

6.14.4.2 `uint32 CFastRep::maxsize`

6.14.4.3 `uint32 CFastRep::size`

6.14.4.4 `bool CFastRep::writable`

The documentation for this class was generated from the following file:

- `oscl_string_rep.h`

## 6.15 CHeapRep Class Reference

```
#include <oscl_string_rep.h>
```

### Public Methods

- [CHeapRep \(\)](#)
- OSCL\_IMPORT\_REF bool [set](#) (uint32, const char \*, [Oscl\\_DefAlloc](#) &)
- OSCL\_IMPORT\_REF bool [set](#) (uint32, const [oscl\\_wchar](#) \*, [Oscl\\_DefAlloc](#) &)
- OSCL\_IMPORT\_REF bool [append](#) (uint32, const char \*, uint32, const char \*, [Oscl\\_DefAlloc](#) &)
- OSCL\_IMPORT\_REF bool [append](#) (uint32, const [oscl\\_wchar](#) \*, uint32, const [oscl\\_wchar](#) \*, [Oscl\\_DefAlloc](#) &)
- OSCL\_IMPORT\_REF void [add\\_ref](#) ()
- OSCL\_IMPORT\_REF void [remove\\_ref](#) ([Oscl\\_DefAlloc](#) &)

### Static Public Methods

- OSCL\_IMPORT\_REF void [set\\_rep](#) (CHheapRep \*&, [Oscl\\_DefAlloc](#) &, const char \*, uint32)
- OSCL\_IMPORT\_REF void [set\\_rep](#) (CHheapRep \*&, [Oscl\\_DefAlloc](#) &, const [oscl\\_wchar](#) \*, uint32)
- OSCL\_IMPORT\_REF void [append\\_rep](#) (CHheapRep \*&, [Oscl\\_DefAlloc](#) &, const char \*, uint32)
- OSCL\_IMPORT\_REF void [append\\_rep](#) (CHheapRep \*&, [Oscl\\_DefAlloc](#) &, const [oscl\\_wchar](#) \*, uint32)
- OSCL\_IMPORT\_REF void [assign](#) (CHheapRep \*&, CHheapRep \*, [Oscl\\_DefAlloc](#) &)

### Data Fields

- uint32 [refcount](#)
- [OsclAny](#) \* [buffer](#)
- uint32 [maxsize](#)
- uint32 [size](#)

#### 6.15.1 Detailed Description

For internal use only– heap string representation

## 6.15.2 Constructor & Destructor Documentation

6.15.2.1 `CHheapRep::CHheapRep () [inline]`

## 6.15.3 Member Function Documentation

6.15.3.1 `OSCL_IMPORT_REF void CHheapRep::add_ref ()`

6.15.3.2 `OSCL_IMPORT_REF bool CHheapRep::append (uint32, const oscl_wchar *, uint32, const oscl_wchar *, Oscl_DefAlloc &)`

6.15.3.3 `OSCL_IMPORT_REF bool CHheapRep::append (uint32, const char *, uint32, const char *, Oscl_DefAlloc &)`

6.15.3.4 `OSCL_IMPORT_REF void CHheapRep::append_rep (CHheapRep *&, Oscl_DefAlloc &, const oscl_wchar *, uint32) [static]`

6.15.3.5 `OSCL_IMPORT_REF void CHheapRep::append_rep (CHheapRep *&, Oscl_DefAlloc &, const char *, uint32) [static]`

6.15.3.6 `OSCL_IMPORT_REF void CHheapRep::assign (CHheapRep *&, CHheapRep *, Oscl_DefAlloc &) [static]`

6.15.3.7 `OSCL_IMPORT_REF void CHheapRep::remove_ref (Oscl_DefAlloc &)`

6.15.3.8 `OSCL_IMPORT_REF bool CHheapRep::set (uint32, const oscl_wchar *, Oscl_DefAlloc &)`

6.15.3.9 `OSCL_IMPORT_REF bool CHheapRep::set (uint32, const char *, Oscl_DefAlloc &)`

6.15.3.10 `OSCL_IMPORT_REF void CHheapRep::set_rep (CHheapRep *&, Oscl_DefAlloc &, const oscl_wchar *, uint32) [static]`

6.15.3.11 `OSCL_IMPORT_REF void CHheapRep::set_rep (CHheapRep *&, Oscl_DefAlloc &, const char *, uint32) [static]`

## 6.15.4 Field Documentation

6.15.4.1 `OsclAny* CHheapRep::buffer`

6.15.4.2 `uint32 CHheapRep::maxsize`

6.15.4.3 `uint32 CHheapRep::refcount`

6.15.4.4 `uint32 CHheapRep::size`

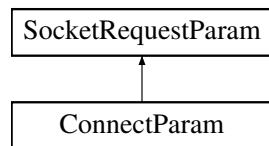
The documentation for this class was generated from the following file:

- `oscl_string_rep.h`

## 6.16 ConnectParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for ConnectParam::



### Public Methods

- [ConnectParam \(OsclNetworkAddress &anAddr\)](#)

### Data Fields

- [OsclNetworkAddress iAddr](#)

#### 6.16.1 Constructor & Destructor Documentation

**6.16.1.1 ConnectParam::ConnectParam (OsclNetworkAddress & *anAddr*) [inline]**

#### 6.16.2 Field Documentation

**6.16.2.1 OsclNetworkAddress ConnectParam::iAddr**

The documentation for this class was generated from the following file:

- [oscl\\_socket\\_request.h](#)

## 6.17 CStackRep Class Reference

```
#include <oscl_string_rep.h>
```

### Public Methods

- [CStackRep \(\)](#)
- [OSCL\\_IMPORT\\_REF void set \(const char \\*cp, uint32 len\)](#)
- [OSCL\\_IMPORT\\_REF void set \(const oscl\\_wchar \\*cp, uint32 len\)](#)
- [OSCL\\_IMPORT\\_REF void append \(const char \\*cp, uint32 len\)](#)
- [OSCL\\_IMPORT\\_REF void append \(const oscl\\_wchar \\*cp, uint32 len\)](#)

### Data Fields

- [uint32 maxsize](#)
- [uint32 size](#)
- [OsclAny \\* buffer](#)

#### 6.17.1 Detailed Description

For internal use only– stack string representation

#### 6.17.2 Constructor & Destructor Documentation

##### 6.17.2.1 CStackRep::CStackRep () [inline]

#### 6.17.3 Member Function Documentation

##### 6.17.3.1 OSCL\_IMPORT\_REF void CStackRep::append (const oscl\_wchar \* cp, uint32 len)

##### 6.17.3.2 OSCL\_IMPORT\_REF void CStackRep::append (const char \* cp, uint32 len)

##### 6.17.3.3 OSCL\_IMPORT\_REF void CStackRep::set (const oscl\_wchar \* cp, uint32 len)

##### 6.17.3.4 OSCL\_IMPORT\_REF void CStackRep::set (const char \* cp, uint32 len)

#### 6.17.4 Field Documentation

##### 6.17.4.1 OsclAny\* CStackRep::buffer

##### 6.17.4.2 uint32 CStackRep::maxsize

##### 6.17.4.3 uint32 CStackRep::size

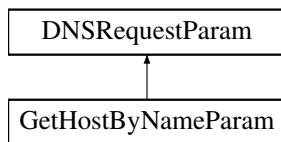
The documentation for this class was generated from the following file:

- [oscl\\_string\\_rep.h](#)

## 6.18 DNSRequestParam Class Reference

```
#include <oscl_dns_param.h>
```

Inheritance diagram for DNSRequestParam::



### Public Methods

- virtual ~DNSRequestParam ()
- void RemoveRef ()
- void InThread ()
- virtual void Destroy ()=0

### Data Fields

- TPVDNSFx<sub>n</sub> iFx<sub>n</sub>
- OsclDNSRequest \* iDNSRequest

### Protected Methods

- DNSRequestParam (TPVDNSFx<sub>n</sub> aFx<sub>n</sub>)

### Protected Attributes

- uint32 iRefCount

#### 6.18.1 Constructor & Destructor Documentation

6.18.1.1 virtual DNSRequestParam::~DNSRequestParam () [inline, virtual]

6.18.1.2 DNSRequestParam::DNSRequestParam (TPVDNSFx<sub>n</sub> aFx<sub>n</sub>) [protected]

#### 6.18.2 Member Function Documentation

6.18.2.1 virtual void DNSRequestParam::Destroy () [pure virtual]

Implemented in [GetHostByNameParam](#).

**6.18.2.2 void DNSRequestParam::InThread ()**

**6.18.2.3 void DNSRequestParam::RemoveRef ()**

### **6.18.3 Field Documentation**

**6.18.3.1 OsclDNSRequest\* DNSRequestParam::iDNSRequest**

**6.18.3.2 TPVDNSFxn DNSRequestParam::iFxn**

**6.18.3.3 uint32 DNSRequestParam::iRefCount [protected]**

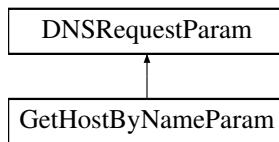
The documentation for this class was generated from the following file:

- [oscl\\_dns\\_param.h](#)

## 6.19 GetHostByNameParam Class Reference

```
#include <oscl_dns_param.h>
```

Inheritance diagram for GetHostByNameParam::



### Public Methods

- void [Destroy \(\)](#)
- [~GetHostByNameParam \(\)](#)

### Static Public Methods

- [GetHostByNameParam \\* Create \(const char \\*name, OsclNetworkAddress \\*&addr\)](#)

### Data Fields

- [char \\* iName](#)
- [OsclNetworkAddress \\* iAddr](#)

#### 6.19.1 Constructor & Destructor Documentation

##### 6.19.1.1 [GetHostByNameParam::~GetHostByNameParam \(\)](#)

#### 6.19.2 Member Function Documentation

##### 6.19.2.1 [GetHostByNameParam\\* GetHostByNameParam::Create \(const char \\* name, OsclNetworkAddress \\*& addr\) \[static\]](#)

##### 6.19.2.2 [void GetHostByNameParam::Destroy \(\) \[virtual\]](#)

Implements [DNSRequestParam](#).

#### 6.19.3 Field Documentation

##### 6.19.3.1 [OsclNetworkAddress\\* GetHostByNameParam::iAddr](#)

##### 6.19.3.2 [char\\* GetHostByNameParam::iName](#)

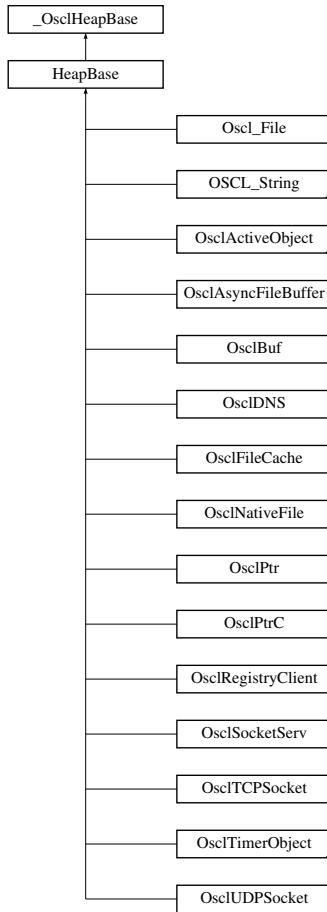
The documentation for this class was generated from the following file:

- [oscl\\_dns\\_param.h](#)

## 6.20 HeapBase Class Reference

```
#include <oscl_mem.h>
```

Inheritance diagram for HeapBase::



### Public Methods

- [HeapBase \(\)](#)
- virtual [~HeapBase \(\)](#)

#### 6.20.1 Detailed Description

HeapBase is the base class for all classes that allocates memory.

HeapBase has overloaded new and delete operators.

Derived from [\\_OsclHeapBase](#) providing CBase\* alike pointer and virtual destructor for cleanupstack to Push and Pop for cleanup when leave occurs.

HeapBase has a virtual destructor which calls the destructor of all the derived classes.

## 6.20.2 Constructor & Destructor Documentation

**6.20.2.1 HeapBase::HeapBase () [inline]**

**6.20.2.2 virtual HeapBase::~HeapBase () [inline, virtual]**

The documentation for this class was generated from the following file:

- [oscl\\_mem.h](#)

## 6.21 internalLeave Class Reference

```
#include <oscl_error_imp_cppexceptions.h>
```

### Data Fields

- int a

#### 6.21.1 Field Documentation

##### 6.21.1.1 int internalLeave::a

The documentation for this class was generated from the following file:

- [oscl\\_error\\_imp\\_cppexceptions.h](#)

## 6.22 LinkedListElement< LLClass > Class Template Reference

```
#include <oscl_linked_list.h>
```

### Public Methods

- [LinkedListElement \(LLClass in\\_data\)](#)

### Data Fields

- [LinkedListElement< LLClass > \\* next](#)
- [LLClass data](#)

#### 6.22.1 Detailed Description

```
template<class LLClass> class LinkedListElement< LLClass >
```

Linked List Element Class

#### 6.22.2 Constructor & Destructor Documentation

```
6.22.2.1 template<class LLClass> LinkedListElement< LLClass >::LinkedListElement  
(LLClass in_data) [inline]
```

#### 6.22.3 Field Documentation

```
6.22.3.1 template<class LLClass> LLClass LinkedListElement< LLClass >::data
```

```
6.22.3.2 template<class LLClass> LinkedListElement<LLClass>*>* LinkedListElement<  
LLClass >::next
```

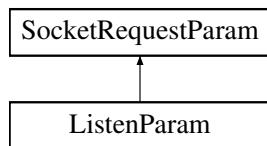
The documentation for this class was generated from the following file:

- [oscl\\_linked\\_list.h](#)

## 6.23 ListenParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for ListenParam::



### Public Methods

- [ListenParam \(uint32 aSize\)](#)

### Data Fields

- uint32 [iQSize](#)

#### 6.23.1 Constructor & Destructor Documentation

6.23.1.1 [ListenParam::ListenParam \(uint32 aSize\) \[inline\]](#)

#### 6.23.2 Field Documentation

##### 6.23.2.1 [uint32 ListenParam::iQSize](#)

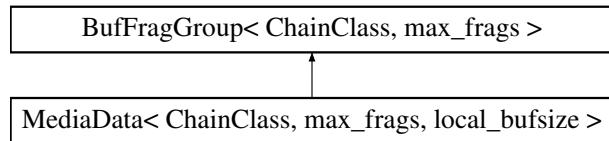
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_request.h](#)

## 6.24 `MediaData< ChainClass, max_frags, local_bufsize >` Class Template Reference

```
#include <oscl_media_data.h>
```

Inheritance diagram for `MediaData< ChainClass, max_frags, local_bufsize >`::



### Public Methods

- `MediaData ()`
- virtual `~MediaData ()`
- `uint32 GetLocalBufsize () const`
- `MediaTimestamp GetTimestamp () const`
- `void SetTimestamp (MediaTimestamp in_timestamp)`
- `uint32 GetAvailableBufferSize () const`
- `MediaStatusClass::status_t GetLocalFragment (BufferFragment &fragment)`
- virtual void `Clear ()`
- `bool IsLocalData (const OsclMemoryFragment &frag) const`
- `int GetMediaSize () const`
- `BufferFragment * GetMediaFragment (const uint32 idx)`
- `uint32 GetNumMediaFrags (const uint32 idx) const`

### Protected Methods

- `MediaStatusClass::status_t AddLocalFragment (const BufferFragment &frag, int32 location_offset)`

### Protected Attributes

- `MediaTimestamp timestamp`
- `uint8 localbuf [local_bufsize]`
- `uint32 available_localbuf`
- `int num_reserved_fragments`

template<class ChainClass, uint32 max\_frags, uint32 local\_bufsize> class MediaData< ChainClass, max\_frags, local\_bufsize >

### 6.24.1 Constructor & Destructor Documentation

6.24.1.1 template<class ChainClass, uint32 max\_frags, uint32 local\_bufsize> MediaData< ChainClass, max\_frags, local\_bufsize >::MediaData () [inline]

6.24.1.2 template<class ChainClass, uint32 max\_frags, uint32 local\_bufsize> virtual MediaData< ChainClass, max\_frags, local\_bufsize >::~MediaData () [inline, virtual]

### 6.24.2 Member Function Documentation

6.24.2.1 template<class ChainClass, uint32 max\_frags, uint32 local\_bufsize> MediaStatusClass::status\_t MediaData< ChainClass, max\_frags, local\_bufsize >::AddLocalFragment (const BufferFragment &frag, int32 location\_offset) [inline, protected]

6.24.2.2 template<class ChainClass, uint32 max\_frags, uint32 local\_bufsize> virtual void MediaData< ChainClass, max\_frags, local\_bufsize >::Clear () [inline, virtual]

Reimplemented from [BufFragGroup< ChainClass, max\\_frags >](#).

## 6.24 MediaData< ChainClass, max\_frags, local\_bufsize > Class Template Reference

---

- 6.24.2.3 template<class ChainClass, uint32 max\_frags, uint32 local\_bufsize> uint32 MediaData<ChainClass, max\_frags, local\_bufsize >::GetAvailableBufferSize () const [inline]
- 6.24.2.4 template<class ChainClass, uint32 max\_frags, uint32 local\_bufsize> uint32 MediaData<ChainClass, max\_frags, local\_bufsize >::GetLocalBufsize () const [inline]
- 6.24.2.5 template<class ChainClass, uint32 max\_frags, uint32 local\_bufsize>  
**MediaStatusClass::status\_t** MediaData<ChainClass, max\_frags, local\_bufsize >::GetLocalFragment (**BufferFragment** & *fragment*) [inline]
- 6.24.2.6 template<class ChainClass, uint32 max\_frags, uint32 local\_bufsize> **BufferFragment\*** MediaData<ChainClass, max\_frags, local\_bufsize >::GetMediaFragment (const uint32 *idx*) [inline]
- 6.24.2.7 template<class ChainClass, uint32 max\_frags, uint32 local\_bufsize> int MediaData<ChainClass, max\_frags, local\_bufsize >::GetMediaSize () const [inline]
- 6.24.2.8 template<class ChainClass, uint32 max\_frags, uint32 local\_bufsize> uint32 MediaData<ChainClass, max\_frags, local\_bufsize >::GetNumMediaFrags (const uint32 *idx*) const [inline]
- 6.24.2.9 template<class ChainClass, uint32 max\_frags, uint32 local\_bufsize> **MediaTimestamp** MediaData<ChainClass, max\_frags, local\_bufsize >::GetTimestamp () const [inline]
- 6.24.2.10 template<class ChainClass, uint32 max\_frags, uint32 local\_bufsize> bool MediaData<ChainClass, max\_frags, local\_bufsize >::IsLocalData (const **OsclMemoryFragment** & *frag*) const [inline]
- 6.24.2.11 template<class ChainClass, uint32 max\_frags, uint32 local\_bufsize> void MediaData<ChainClass, max\_frags, local\_bufsize >::SetTimestamp (**MediaTimestamp** *in\_timestamp*) [inline]

### 6.24.3 Field Documentation

- 6.24.3.1 template<class ChainClass, uint32 max\_frags, uint32 local\_bufsize> uint32 MediaData<ChainClass, max\_frags, local\_bufsize >::available\_localbuf [protected]
- 6.24.3.2 template<class ChainClass, uint32 max\_frags, uint32 local\_bufsize> uint8 MediaData<ChainClass, max\_frags, local\_bufsize >::localbuf[local\_bufsize] [protected]
- 6.24.3.3 template<class ChainClass, uint32 max\_frags, uint32 local\_bufsize> int MediaData<ChainClass, max\_frags, local\_bufsize >::num\_reserved.fragments [protected]
- 6.24.3.4 template<class ChainClass, uint32 max\_frags, uint32 local\_bufsize> **MediaTimestamp** MediaData<ChainClass, max\_frags, local\_bufsize >::timestamp [protected]

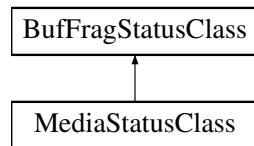
The documentation for this class was generated from the following file:

- [oscl\\_media\\_data.h](#)

## 6.25 MediaStatusClass Class Reference

```
#include <oscl_media_status.h>
```

Inheritance diagram for MediaStatusClass::



The documentation for this class was generated from the following file:

- [oscl\\_media\\_status.h](#)

## 6.26 MemAllocator< T > Class Template Reference

```
#include <oscl_media_data.h>
```

### Public Types

- `typedef T * pointer`

### Public Methods

- `virtual pointer allocate (void *hint=0, const int num_reserved_frags=1)=0`
- `virtual void deallocate (pointer p)=0`
- `virtual ~MemAllocator ()`

```
template<class T> class MemAllocator< T >
```

#### 6.26.1 Member Typedef Documentation

##### 6.26.1.1 template<class T> `typedef T* MemAllocator< T >::pointer`

#### 6.26.2 Constructor & Destructor Documentation

##### 6.26.2.1 template<class T> `virtual MemAllocator< T >::~MemAllocator () [inline, virtual]`

#### 6.26.3 Member Function Documentation

##### 6.26.3.1 template<class T> `virtual pointer MemAllocator< T >::allocate (void * hint = 0, const int num_reserved_frags = 1) [pure virtual]`

##### 6.26.3.2 template<class T> `virtual void MemAllocator< T >::deallocate (pointer p) [pure virtual]`

The documentation for this class was generated from the following file:

- [oscl\\_media\\_data.h](#)

## 6.27 MM\_AllocBlockFence Struct Reference

```
#include <oscl_mem_audit_internals.h>
```

### Public Methods

- [MM\\_AllocBlockFence \(\)](#)
- [void fill\\_fence \(\)](#)
- [bool check\\_fence \(\)](#)

### Data Fields

- [uint8 pad \[COMPUTE\\_MEM\\_ALIGN\\_SIZE\(sizeof\(MM\\_AllocBlockHdr\), MIN\\_FENCE\\_SIZE, MEM\\_ALIGN\\_SIZE\)\]](#)

#### 6.27.1 Constructor & Destructor Documentation

[6.27.1.1 MM\\_AllocBlockFence::MM\\_AllocBlockFence \(\) \[inline\]](#)

#### 6.27.2 Member Function Documentation

[6.27.2.1 bool MM\\_AllocBlockFence::check\\_fence \(\) \[inline\]](#)

[6.27.2.2 void MM\\_AllocBlockFence::fill\\_fence \(\) \[inline\]](#)

#### 6.27.3 Field Documentation

[6.27.3.1 uint8 MM\\_AllocBlockFence::pad\[COMPUTE\\_MEM\\_ALIGN\\_SIZE\(sizeof\(MM\\_AllocBlockHdr\), MIN\\_FENCE\\_SIZE, MEM\\_ALIGN\\_SIZE\)\]](#)

The documentation for this struct was generated from the following file:

- [oscl\\_mem\\_audit\\_internals.h](#)

## 6.28 MM\_AllocBlockHdr Struct Reference

```
#include <oscl_mem_audit_internals.h>
```

### Public Methods

- `bool isAllocNodePtr ()`
- `void setAllocNodeFlag ()`
- `MM_AllocBlockHdr ()`
- `MM_AllocBlockHdr (void *ptr, uint32 inSize)`

### Data Fields

- `void * pNode`
- `uint32 size`
- `void * pRootNode`
- `uint32 pad`

### Static Public Attributes

- `const uint32 ALLOC_NODE_FLAG = 0x80000000`

#### 6.28.1 Constructor & Destructor Documentation

**6.28.1.1 `MM_AllocBlockHdr::MM_AllocBlockHdr () [inline]`**

**6.28.1.2 `MM_AllocBlockHdr::MM_AllocBlockHdr (void *ptr, uint32 inSize) [inline]`**

#### 6.28.2 Member Function Documentation

**6.28.2.1 `bool MM_AllocBlockHdr::isAllocNodePtr () [inline]`**

**6.28.2.2 `void MM_AllocBlockHdr::setAllocNodeFlag () [inline]`**

#### 6.28.3 Field Documentation

**6.28.3.1 `uint32 MM_AllocBlockHdr::pad`**

**6.28.3.2 `void* MM_AllocBlockHdr::pNode`**

**6.28.3.3 `void* MM_AllocBlockHdr::pRootNode`**

**6.28.3.4 `uint32 MM_AllocBlockHdr::size`**

The documentation for this struct was generated from the following file:

- `oscl_mem_audit_internals.h`

## 6.29 MM\_AllocInfo Struct Reference

```
#include <oscl_mem_audit.h>
```

### Public Methods

- [MM\\_AllocInfo \(\)](#)
- [~MM\\_AllocInfo \(\)](#)
- [void \\* operator new \(oscl\\_memsize\\_t size\)](#)
- [void \\* operator new \(oscl\\_memsize\\_t size, MM\\_AllocInfo \\*ptr\)](#)
- [void operator delete \(void \\*ptr\) throw \(\)](#)

### Data Fields

- [uint32 allocNum](#)
- [char \\* pFileName](#)
- [uint32 lineNo](#)
- [uint32 size](#)
- [void \\* pMemBlock](#)
- [OsclMemStatsNode \\* pStatsNode](#)
- [bool bSetFailure](#)

## 6.29.1 Constructor & Destructor Documentation

6.29.1.1 `MM_AllocInfo::MM_AllocInfo () [inline]`

6.29.1.2 `MM_AllocInfo::~MM_AllocInfo () [inline]`

## 6.29.2 Member Function Documentation

6.29.2.1 `void MM_AllocInfo::operator delete (void *ptr) throw () [inline]`

6.29.2.2 `void* MM_AllocInfo::operator new (oscl_memsize_t size, MM_AllocInfo *ptr) [inline]`

6.29.2.3 `void* MM_AllocInfo::operator new (oscl_memsize_t size) [inline]`

## 6.29.3 Field Documentation

6.29.3.1 `uint32 MM_AllocInfo::allocNum`

6.29.3.2 `bool MM_AllocInfo::bSetFailure`

6.29.3.3 `uint32 MM_AllocInfo::lineNo`

6.29.3.4 `char* MM_AllocInfo::pFileName`

6.29.3.5 `void* MM_AllocInfo::pMemBlock`

6.29.3.6 `OsclMemStatsNode* MM_AllocInfo::pStatsNode`

6.29.3.7 `uint32 MM_AllocInfo::size`

The documentation for this struct was generated from the following file:

- [oscl\\_mem\\_audit.h](#)

## 6.30 MM\_AllocNode Struct Reference

```
#include <oscl_mem_audit.h>
```

### Public Methods

- [MM\\_AllocNode \(\)](#)
- [~MM\\_AllocNode \(\)](#)
- [void \\* operator new \(oscl\\_memsize\\_t size\)](#)
- [void \\* operator new \(oscl\\_memsize\\_t size, MM\\_AllocNode \\*ptr\)](#)
- [void operator delete \(void \\*ptr\) throw \(\)](#)

### Data Fields

- [MM\\_AllocInfo \\* pAllocInfo](#)
- [MM\\_AllocNode \\* pPrev](#)
- [MM\\_AllocNode \\* pNext](#)

#### 6.30.1 Constructor & Destructor Documentation

**6.30.1.1** [MM\\_AllocNode::MM\\_AllocNode \(\) \[inline\]](#)

**6.30.1.2** [MM\\_AllocNode::~MM\\_AllocNode \(\) \[inline\]](#)

#### 6.30.2 Member Function Documentation

**6.30.2.1** [void MM\\_AllocNode::operator delete \(void \\*ptr\) throw \(\) \[inline\]](#)

**6.30.2.2** [void\\* MM\\_AllocNode::operator new \(oscl\\_memsize\\_t size, MM\\_AllocNode \\*ptr\) \[inline\]](#)

**6.30.2.3** [void\\* MM\\_AllocNode::operator new \(oscl\\_memsize\\_t size\) \[inline\]](#)

#### 6.30.3 Field Documentation

**6.30.3.1** [MM\\_AllocInfo\\* MM\\_AllocNode::pAllocInfo](#)

**6.30.3.2** [MM\\_AllocNode\\* MM\\_AllocNode::pNext](#)

**6.30.3.3** [MM\\_AllocNode\\* MM\\_AllocNode::pPrev](#)

The documentation for this struct was generated from the following file:

- [oscl\\_mem\\_audit.h](#)

## 6.31 MM\_AllocQueryInfo Struct Reference

```
#include <oscl_mem_audit.h>
```

### Data Fields

- uint32 [allocNum](#)
- char [fileName](#) [MM\_ALLOC\_MAX\_QUERY\_FILENAME\_LEN]
- uint32 [lineNo](#)
- uint32 [size](#)
- const void \* [pMemBlock](#)
- char [tag](#) [MM\_ALLOC\_MAX\_QUERY\_TAG\_LEN]

### 6.31.1 Field Documentation

**6.31.1.1 uint32 MM\_AllocQueryInfo::allocNum**

**6.31.1.2 char MM\_AllocQueryInfo::fileName[MM\_ALLOC\_MAX\_QUERY\_FILENAME\_LEN]**

**6.31.1.3 uint32 MM\_AllocQueryInfo::lineNo**

**6.31.1.4 const void\* MM\_AllocQueryInfo::pMemBlock**

**6.31.1.5 uint32 MM\_AllocQueryInfo::size**

**6.31.1.6 char MM\_AllocQueryInfo::tag[MM\_ALLOC\_MAX\_QUERY\_TAG\_LEN]**

The documentation for this struct was generated from the following file:

- [oscl\\_mem\\_audit.h](#)

## 6.32 MM\_Audit\_Imp Class Reference

```
#include <oscl_mem_audit.h>
```

### Public Methods

- [MM\\_Audit\\_Imp \(\)](#)
- [~MM\\_Audit\\_Imp \(\)](#)
- [OSCL\\_IMPORT\\_REF void \\* MM\\_allocate \(const OsclMemStatsNode \\*statsNode, uint32 sizeIn, const char \\*pFileName, uint32 lineNumber, bool allocNodeTracking=false\)](#)
- [OSCL\\_IMPORT\\_REF bool MM\\_deallocate \(void \\*pMemBlockIn\)](#)
- [OSCL\\_IMPORT\\_REF MM\\_Stats\\_t \\* MM\\_GetStats \(const char \\*const tagIn\)](#)
- [OSCL\\_IMPORT\\_REF uint32 MM\\_GetStatsInDepth \(const char \\*tagIn, MM\\_Stats\\_CB \\*array\\_ptr, uint32 max\\_nodes\)](#)
- [OSCL\\_IMPORT\\_REF uint32 MM\\_GetTreeNodes \(const char \\*tagIn\)](#)
- [OSCL\\_IMPORT\\_REF bool MM\\_AddTag \(const char \\*tagIn\)](#)
- [OSCL\\_IMPORT\\_REF const OsclMemStatsNode \\* MM\\_GetTagName \(const char \\*tagIn\)](#)
- [OSCL\\_IMPORT\\_REF const OsclMemStatsNode \\* MM\\_GetExistingTag \(const char \\*tagIn\)](#)
- [OSCL\\_IMPORT\\_REF const OsclMemStatsNode \\* MM\\_GetRootNode \(\)](#)
- [OSCL\\_IMPORT\\_REF MM\\_AllocQueryInfo \\* MM\\_CreateAllocNodeInfo \(uint32 max\\_array\\_size\)](#)
- [OSCL\\_IMPORT\\_REF void MM\\_ReleaseAllocNodeInfo \(MM\\_AllocQueryInfo \\*info\)](#)
- [OSCL\\_IMPORT\\_REF uint32 MM\\_GetAllocNodeInfo \(MM\\_AllocQueryInfo \\*output\\_array, uint32 max\\_array\\_size, uint32 offset\)](#)
- [OSCL\\_IMPORT\\_REF bool MM\\_Validate \(const void \\*ptrIn\)](#)
- [uint32 MM\\_GetAllocNo \(void\)](#)
- [void MM\\_GetOverheadStats \(MM\\_AuditOverheadStats &stats\)](#)
- [uint32 MM\\_GetNumAllocNodes \(\)](#)
- [uint32 MM\\_GetMode \(void\)](#)
- [uint8 MM\\_GetPrefillPattern \(void\)](#)
- [uint32 MM\\_GetPostfillPattern \(void\)](#)
- [OSCL\\_IMPORT\\_REF void MM\\_SetMode \(uint32 inMode\)](#)
- [OSCL\\_IMPORT\\_REF void MM\\_SetPrefillPattern \(uint8 pattern\)](#)
- [OSCL\\_IMPORT\\_REF void MM\\_SetPostfillPattern \(uint8 pattern\)](#)
- [OSCL\\_IMPORT\\_REF void MM\\_SetTagLevel \(uint32 level\)](#)
- [OSCL\\_IMPORT\\_REF bool MM\\_SetFailurePoint \(const char \\*tagIn, uint32 alloc\\_number\)](#)
- [OSCL\\_IMPORT\\_REF void MM\\_UnsetFailurePoint \(const char \\*tagIn\)](#)
- [MM\\_AllocNode \\* addAllocNode \(void \\*pMem, uint32 sizeIn, OsclMemStatsNode \\*pStatsNode, const char \\*pFileName, uint32 lineNumber\)](#)
- [OsclMemStatsNode \\* removeAllocNode \(void \\*pMemBlockIn, uint32 &size\)](#)
- [void removeALLAllocNodes \(\)](#)
- [OsclMemStatsNode \\* createStatsNode \(const char \\*tagIn\)](#)
- [bool updateStatsNode \(OsclMemStatsNode \\*pCurrStatsNode, const MM\\_Stats\\_t &pDelta, bool bAdd\)](#)
- [bool updateStatsNodeInFailure \(const char \\*tagIn\)](#)
- [bool updateStatsNodeInFailure \(OsclMemStatsNode \\*pStatsNode\)](#)
- [bool pruneSubtree \(OsclMemStatsNode \\*pNode\)](#)
- [bool pruneSubtree \(const char \\*tagIn\)](#)
- [void retrieveParentTag \(char \\*tag\)](#)
- [int32 retrieveParentTagLength \(const char \\*tag, int32 bound\)](#)
- [void makeValidTag \(const char \\*tagIn, MMAuditCharAutoPtr &autoPtr\)](#)

- uint32 [getTagActualSize](#) (const char \*tagIn)
- bool [isSetFailure](#) (const char \*tagIn)
- bool [isSetFailure](#) (OsclMemStatsNode \*statsNode)
- bool [validate\\_all\\_heap](#) ()

## Static Public Methods

- bool [validate](#) (void \*ptrIn)
- OsclMemAudit \* [getAuditRoot](#) (void \*ptrIn)
- uint32 [getSize](#) (void \*ptrIn)

### 6.32.1 Constructor & Destructor Documentation

#### 6.32.1.1 MM\_Audit\_Imp::MM\_Audit\_Imp ()

Constructor, create the root node in statistics table

#### 6.32.1.2 MM\_Audit\_Imp::~MM\_Audit\_Imp ()

A destructor, remove all the nodes in allocation and statistics table

### 6.32.2 Member Function Documentation

#### 6.32.2.1 MM\_AllocNode\* MM\_Audit\_Imp::addAllocNode (void \* pMem, uint32 sizeIn, OsclMemStatsNode \* pStatsNode, const char \* pFileName, uint32 lineNumber)

##### Returns:

true if operation succeeds;

#### 6.32.2.2 OsclMemStatsNode\* MM\_Audit\_Imp::createStatsNode (const char \* tagIn)

##### Returns:

true if operation succeeds;

#### 6.32.2.3 OsclMemAudit\* MM\_Audit\_Imp::getAuditRoot (void \* ptrIn) [static]

##### Returns:

audit root pointer.

#### 6.32.2.4 uint32 MM\_Audit\_Imp::getSize (void \* ptrIn) [static]

##### Returns:

original block size. leaves if bad pointer.

**6.32.2.5 uint32 MM\_Audit\_Imp::getTagActualSize (const char \* tagIn)****Returns:**

the size of the truncated tag; 0 means NO truncation

**6.32.2.6 bool MM\_Audit\_Imp::isSetFailure (OsclMemStatsNode \* statsNode)****6.32.2.7 bool MM\_Audit\_Imp::isSetFailure (const char \* tagIn)****Returns:**

true if operation succeeds;

**6.32.2.8 void MM\_Audit\_Imp::makeValidTag (const char \* tagIn, MMAuditCharAutoPtr & autoptr)****Returns:**

a valid tag; NULL will be converted into root tag

**6.32.2.9 OSCL\_IMPORT\_REF bool MM\_Audit\_Imp::MM\_AddTag (const char \* tagIn) [inline]**

API to add a node and zero out its counters; Note that this tag should be re-used

**Parameters:**

*tagIn* input tag

**Returns:**

true if operation succeeds;

**6.32.2.10 OSCL\_IMPORT\_REF void\* MM\_Audit\_Imp::MM\_allocate (const OsclMemStatsNode \* statsNode, uint32 sizeIn, const char \* pFileName, uint32 lineNumber, bool allocNodeTracking = false)**

The following are APIs t \_\_nothrow\_ / const \_\_nothrow\_

**Returns:**

the memory pointer if operation succeeds.

**6.32.2.11 OSCL\_IMPORT\_REF MM\_AllocQueryInfo\* MM\_Audit\_Imp::MM\_CreateAllocNode-Info (uint32 max\_array\_size)**

These APIs will allocate and release space for alloc node info, to be used with the MM\_GetAllocNodeInfo API.

**6.32.2.12 OSCL\_IMPORT\_REF bool MM\_Audit\_Imp::MM\_deallocate (void \* pMemBlockIn)****Returns:**

true if operation succeeds;

**6.32.2.13 uint32 MM\_Audit\_Imp::MM\_GetAllocNo (void) [inline]**

API to get the current allocation number

**Returns:**

the current allocation number

**6.32.2.14 OSCL\_IMPORT\_REF uint32 MM\_Audit\_Imp::MM\_GetAllocNodeInfo  
(MM\_AllocQueryInfo \* output\_array, uint32 max\_array\_size, uint32 offset)**

API to query the list of alloc nodes. It copies the information into the provided output array.

**Parameters:**

*output\_array* the array where the data will be written

*max\_array\_size* the max number of output array elements

*offset* the offset into the alloc node list from which the data should begin.

**Returns:**

the number of valid nodes in the output array

**6.32.2.15 OSCL\_IMPORT\_REF const OsclMemStatsNode\* MM\_Audit\_Imp::MM\_GetExisting-  
Tag (const char \* tagIn)**

API to add a node and zero out its counters; Note that this tag should be re-used

**Parameters:**

*tagIn* input tag

**Returns:**

true if operation succeeds;

**6.32.2.16 uint32 MM\_Audit\_Imp::MM\_GetMode (void) [inline]**

API to get the operating mode of the mm\_audit class.

**6.32.2.17 uint32 MM\_Audit\_Imp::MM\_GetNumAllocNodes () [inline]**

API to get the number of allocation nodes (records) for allocations that are being tracked individually.

**6.32.2.18 void MM\_Audit\_Imp::MM\_GetOverheadStats (MM\_AuditOverheadStats & stats)  
[inline]**

API to get the overhead statistics for the memory used by the mm\_audit class.

**6.32.2.19 uint32 MM\_Audit\_Imp::MM\_GetPostfillPattern (void) [inline]**

API to get the postfill pattern. The pattern is used to fill the memory before freeing it.

**6.32.2.20 uint8 MM\_Audit\_Imp::MM\_GetPrefillPattern (void) [inline]**

API to get the prefill pattern. The pattern is used to fill the memory before returning it to the caller.

**6.32.2.21 OSCL\_IMPORT\_REF const OsclMemStatsNode\* MM\_Audit\_Imp::MM\_GetRootNode () [inline]****6.32.2.22 OSCL\_IMPORT\_REF MM\_Stats\_t\* MM\_Audit\_Imp::MM\_GetStats (const char \*const tagIn)**

API to get memory statistics through context string(tag)

**Returns:**

statistics pointer if operation succeeds

**6.32.2.23 OSCL\_IMPORT\_REF uint32 MM\_Audit\_Imp::MM\_GetStatsInDepth (const char \* tagIn, MM\_Stats\_CB \* array\_ptr, uint32 max\_nodes)**

API to get memory statistics in detail through context string(tag) including its subtree

**Returns:**

statistics pointer array and actual number of nodes if operation succeeds

**6.32.2.24 OSCL\_IMPORT\_REF const OsclMemStatsNode\* MM\_Audit\_Imp::MM\_GetTagName (const char \* tagIn)**

API to add a node and zero out its counters; Note that this tag should be re-used

**Parameters:**

*tagIn* input tag

**Returns:**

pointer to [OsclMemStatsNode](#) which should be passed to MM\_allocate

**6.32.2.25 OSCL\_IMPORT\_REF uint32 MM\_Audit\_Imp::MM\_GetTreeNodes (const char \* tagIn)**

API to get the number of tree nodes including the tag node and its subtree

**Parameters:**

*tagIn* input tag

**Returns:**

the number of tree nodes ; 0 means no tag node

**6.32.2.26 OSCL\_IMPORT\_REF void MM\_Audit\_Imp::MM\_ReleaseAllocNodeInfo  
([MM\\_AllocQueryInfo](#) \* *info*)**

**6.32.2.27 OSCL\_IMPORT\_REF bool MM\_Audit\_Imp::MM\_SetFailurePoint (const char \* *tagIn*,  
uint32 *alloc\_number*)**

API to insert allocation failure deterministically according to allocation number associated with tag

**Parameters:**

*tagIn* input tag

*alloc\_number* allocation number associated with tag

**Returns:**

true if operation succeeds;

**6.32.2.28 OSCL\_IMPORT\_REF void MM\_Audit\_Imp::MM\_SetMode (uint32 *inMode*)**

API to set the operating mode of the mm\_audit class.

**6.32.2.29 OSCL\_IMPORT\_REF void MM\_Audit\_Imp::MM\_SetPostfillPattern (uint8 *pattern*)**

API to set the postfill pattern.

**6.32.2.30 OSCL\_IMPORT\_REF void MM\_Audit\_Imp::MM\_SetPrefillPattern (uint8 *pattern*)**

API to set the prefill pattern.

**6.32.2.31 OSCL\_IMPORT\_REF void MM\_Audit\_Imp::MM\_SetTagLevel (uint32 *level*)**

API to set the maximum tag level,i.e. tag level for a.b.c.d = 4

**Parameters:**

*level* input tag level to be set

**6.32.2.32 OSCL\_IMPORT\_REF void MM\_Audit\_Imp::MM\_UnsetFailurePoint (const char \*  
*tagIn*)**

API to cancel the allocation failure point associated with tag

**Parameters:**

*tagIn* input tag

**6.32.2.33 OSCL\_IMPORT\_REF bool MM\_Audit\_Imp::MM\_Validate (const void \* *ptrIn*)**

API to check the input pointer is a valid pointer to a chunk of memory

**Parameters:**

*ptrIn* input pointer to be validated

**Returns:**

true if operation succeeds;

**6.32.2.34** `bool MM_Audit_Imp::pruneSubtree (const char * tagIn)`

**6.32.2.35** `bool MM_Audit_Imp::pruneSubtree (OsclMemStatsNode * pNode)`

**Returns:**

true if operation succeeds;

**6.32.2.36** `void MM_Audit_Imp::removeALLAllocNodes ()`

**6.32.2.37** `OsclMemStatsNode* MM_Audit_Imp::removeAllocNode (void * pMemBlockIn, uint32 & size)`

**Returns:**

true if operation succeeds;

**6.32.2.38** `void MM_Audit_Imp::retrieveParentTag (char * tag)`

**6.32.2.39** `int32 MM_Audit_Imp::retrieveParentTagLength (const char * tag, int32 bound)`

**Returns:**

the length of a immediate parent tag for the input tag

**6.32.2.40** `bool MM_Audit_Imp::updateStatsNode (OsclMemStatsNode * pCurrStatsNode, const MM_Stats_t & pDelta, bool bAdd)`

**Returns:**

true if operation succeeds;

**6.32.2.41** `bool MM_Audit_Imp::updateStatsNodeInFailure (OsclMemStatsNode * pStatsNode)`

**6.32.2.42** `bool MM_Audit_Imp::updateStatsNodeInFailure (const char * tagIn)`

**Returns:**

true if operation succeeds;

**6.32.2.43** `bool MM_Audit_Imp::validate (void * ptrIn) [static]`

**Returns:**

true if operation succeeds;

**6.32.2.44 bool MM\_Audit\_Imp::validate\_all\_heap ()****Returns:**

true if operation succeeds;

The documentation for this class was generated from the following file:

- [oscl\\_mem\\_audit.h](#)

## 6.33 MM\_AuditOverheadStats Struct Reference

```
#include <oscl_mem_audit.h>
```

### Data Fields

- uint32 [per\\_allocation\\_overhead](#)
- uint32 [stats\\_overhead](#)

#### 6.33.1 Field Documentation

**6.33.1.1 uint32 MM\_AuditOverheadStats::per\_allocation\_overhead**

**6.33.1.2 uint32 MM\_AuditOverheadStats::stats\_overhead**

The documentation for this struct was generated from the following file:

- [oscl\\_mem\\_audit.h](#)

## 6.34 MM\_FailInsertParam Struct Reference

```
#include <oscl_mem_audit.h>
```

### Public Methods

- [MM\\_FailInsertParam \(\)](#)
- [void reset \(\)](#)
- [void \\* operator new \(oscl\\_memsize\\_t size\)](#)
- [void \\* operator new \(oscl\\_memsize\\_t size, MM\\_FailInsertParam \\*ptr\)](#)
- [void operator delete \(void \\*ptr\) throw \(\)](#)

### Data Fields

- [uint32 nAllocNum](#)
- [uint16 xsubi \[3\]](#)

#### 6.34.1 Constructor & Destructor Documentation

**6.34.1.1 [MM\\_FailInsertParam::MM\\_FailInsertParam \(\) \[inline\]](#)**

#### 6.34.2 Member Function Documentation

**6.34.2.1 [void MM\\_FailInsertParam::operator delete \(void \\*ptr\) throw \(\) \[inline\]](#)**

**6.34.2.2 [void\\* MM\\_FailInsertParam::operator new \(oscl\\_memsize\\_t size, MM\\_FailInsertParam \\*ptr\) \[inline\]](#)**

**6.34.2.3 [void\\* MM\\_FailInsertParam::operator new \(oscl\\_memsize\\_t size\) \[inline\]](#)**

**6.34.2.4 [void MM\\_FailInsertParam::reset \(\) \[inline\]](#)**

#### 6.34.3 Field Documentation

**6.34.3.1 [uint32 MM\\_FailInsertParam::nAllocNum](#)**

**6.34.3.2 [uint16 MM\\_FailInsertParam::xsubi\[3\]](#)**

The documentation for this struct was generated from the following file:

- [oscl\\_mem\\_audit.h](#)

## 6.35 MM\_Stats\_CB Struct Reference

```
#include <oscl_mem_audit.h>
```

### Public Methods

- [MM\\_Stats\\_CB \(\)](#)
- [void \\* operator new \(oscl\\_memsize\\_t size\)](#)
- [void \\* operator new \(oscl\\_memsize\\_t size, MM\\_Stats\\_CB \\*ptr\)](#)
- [void operator delete \(void \\*ptr\) throw \(\)](#)

### Data Fields

- [const char \\* tag](#)
- [const MM\\_Stats\\_t \\* pStats](#)
- [uint32 num\\_child\\_nodes](#)

#### 6.35.1 Constructor & Destructor Documentation

**6.35.1.1 [MM\\_Stats\\_CB::MM\\_Stats\\_CB \(\) \[inline\]](#)**

#### 6.35.2 Member Function Documentation

**6.35.2.1 [void MM\\_Stats\\_CB::operator delete \(void \\*ptr\) throw \(\) \[inline\]](#)**

**6.35.2.2 [void\\* MM\\_Stats\\_CB::operator new \(oscl\\_memsize\\_t size, MM\\_Stats\\_CB \\*ptr\) \[inline\]](#)**

**6.35.2.3 [void\\* MM\\_Stats\\_CB::operator new \(oscl\\_memsize\\_t size\) \[inline\]](#)**

#### 6.35.3 Field Documentation

**6.35.3.1 [uint32 MM\\_Stats\\_CB::num\\_child\\_nodes](#)**

**6.35.3.2 [const MM\\_Stats\\_t\\* MM\\_Stats\\_CB::pStats](#)**

**6.35.3.3 [const char\\* MM\\_Stats\\_CB::tag](#)**

The documentation for this struct was generated from the following file:

- [oscl\\_mem\\_audit.h](#)

## 6.36 MM\_Stats\_t Struct Reference

```
#include <oscl_mem_audit.h>
```

### Public Methods

- [MM\\_Stats\\_t \(\)](#)
- [MM\\_Stats\\_t \(uint32 sizeIn\)](#)
- [void reset \(\)](#)
- [void update \(const MM\\_Stats\\_t &delta, bool add\)](#)
- [void \\* operator new \(oscl\\_memsize\\_t size\)](#)
- [void \\* operator new \(oscl\\_memsize\\_t size, MM\\_Stats\\_t \\*ptr\)](#)
- [void operator delete \(void \\*ptr\) throw \(\)](#)

### Data Fields

- [uint32 numBytes](#)
- [uint32 peakNumBytes](#)
- [uint32 numAllocs](#)
- [uint32 peakNumAllocs](#)
- [uint32 numAllocFails](#)
- [uint32 totalNumAllocs](#)
- [uint32 totalNumBytes](#)

### 6.36.1 Constructor & Destructor Documentation

6.36.1.1 `MM_Stats_t::MM_Stats_t () [inline]`

6.36.1.2 `MM_Stats_t::MM_Stats_t (uint32 sizeIn) [inline]`

### 6.36.2 Member Function Documentation

6.36.2.1 `void MM_Stats_t::operator delete (void *ptr) throw () [inline]`

6.36.2.2 `void* MM_Stats_t::operator new (oscl_memsize_t size, MM_Stats_t *ptr) [inline]`

6.36.2.3 `void* MM_Stats_t::operator new (oscl_memsize_t size) [inline]`

6.36.2.4 `void MM_Stats_t::reset () [inline]`

6.36.2.5 `void MM_Stats_t::update (const MM_Stats_t & delta, bool add) [inline]`

### 6.36.3 Field Documentation

6.36.3.1 `uint32 MM_Stats_t::numAllocFails`

6.36.3.2 `uint32 MM_Stats_t::numAllocs`

6.36.3.3 `uint32 MM_Stats_t::numBytes`

6.36.3.4 `uint32 MM_Stats_t::peakNumAllocs`

6.36.3.5 `uint32 MM_Stats_t::peakNumBytes`

6.36.3.6 `uint32 MM_Stats_t::totalNumAllocs`

6.36.3.7 `uint32 MM_Stats_t::totalNumBytes`

The documentation for this struct was generated from the following file:

- [oscl\\_mem\\_audit.h](#)

## 6.37 NTPTime Class Reference

The NTPTime class represents a time value as the number of seconds since 0h (UTC) Jan. 1, 1900.

```
#include <oscl_time.h>
```

### Public Methods

- **OSCL\_COND\_IMPORT\_REF NTPTime ()**  
*The default constructor creates an NTPTime instance representing the current system time.*
- **OSCL\_COND\_IMPORT\_REF NTPTime (const NTPTime &src)**  
*Copy constructor to create a new NTPTime from an existing one.*
- **OSCL\_COND\_IMPORT\_REF NTPTime (const uint32 seconds)**  
*Construct an NTPTime from a uint32.*
- **OSCL\_COND\_IMPORT\_REF NTPTime (const int32 seconds)**  
*Construct an NTPTime from a int.*
- **OSCL\_COND\_IMPORT\_REF NTPTime (const TimeValue &t)**  
*Construct a NTPTime instance from a TimeValue instance.*
- **OSCL\_COND\_IMPORT\_REF NTPTime (const uint64 value)**  
*Construct a NTPTime instance from a uint64 value.*
- **OSCL\_COND\_IMPORT\_REF NTPTime & operator= (uint32 newval)**  
*The assignment operator for a 32 bit integer.*
- **OSCL\_COND\_IMPORT\_REF NTPTime & operator= (uint64 newval)**  
*The assignment operator for a 64 bit integer.*
- **OSCL\_COND\_IMPORT\_REF NTPTime & operator+= (uint64 val)**  
*The += operator is used to add a 64 bit 32.32 value to an existing NTPTime value.*
- **OSCL\_COND\_IMPORT\_REF NTPTime operator- (const NTPTime &npt) const**  
*The - operator allows subtraction of one NTPTime value from another. This is useful to measure an interval.*
- **void set\_from\_system\_time (const uint32 systemtime)**  
*This method converts a 32-bit system time to NTP time.*
- **OSCL\_COND\_IMPORT\_REF uint32 get\_middle32 () const**  
*Grab the middle 32 bits of the 64 bit 32.32 representation.*
- **OSCL\_COND\_IMPORT\_REF uint32 get\_upper32 () const**  
*This method returns the upper 32 bits of the 32.32 representation.*
- **OSCL\_COND\_IMPORT\_REF uint32 get\_lower32 () const**  
*This method returns the lower 32 bits of the 32.32 representation.*

- int32 [to\\_system\\_time \(\) const](#)

*This method converts the ntp time value to system time.*

- OSCL\_COND\_IMPORT\_REF [uint64 get\\_value \(\) const](#)

*This method returns the 32.32 ntp representation.*

- OSCL\_IMPORT\_REF int [set\\_to\\_current\\_time \(\)](#)

*This method sets the 32.32 representation to the current system time value.*

### 6.37.1 Detailed Description

The NTPTime class represents a time value as the number of seconds since 0h (UTC) Jan. 1, 1900.

The NTPTime class: Conversion to/from Unix (epoch at 0h Jan. 1, 1970) amount to addition/subtraction of 2208988800. A single 64 bit value is used to represent the time. This value represents the number of seconds since 0h (UTC) Jan. 1, 1900. There is an implied binary point between the upper 32 bits and lower 32 bits (this is referred to as a 32.32 fractional representation). For example a binary value of 00000000 00000000 00000011 10000000 00000000 00000000 00000000 represents 3.5 seconds since Jan 1, 1900.

### 6.37.2 Constructor & Destructor Documentation

#### 6.37.2.1 OSCL\_COND\_IMPORT\_REF NTPTime::NTPTime ()

The default constructor creates an NTPTime instance representing the current system time.

#### 6.37.2.2 OSCL\_COND\_IMPORT\_REF NTPTime::NTPTime (const NTPTime & src)

Copy constructor to create a new NTPTime from an existing one.

#### 6.37.2.3 OSCL\_COND\_IMPORT\_REF NTPTime::NTPTime (const uint32 seconds)

Construct an NTPTime from a uint32.

##### Parameters:

**seconds** The uint32 input represents the number of seconds since Jan. 1, 1900.

#### 6.37.2.4 OSCL\_COND\_IMPORT\_REF NTPTime::NTPTime (const int32 seconds)

Construct an NTPTime from a int.

##### Parameters:

**seconds** The int input represents the number of seconds since Jan. 1, 1900.

### 6.37.2.5 OSCL\_COND\_IMPORT\_REF NTPTime::NTPTime (const TimeValue & t)

Construct a NTPTime instance from a TimeValue instance.

This constructor creates an NTPTime value representing the same absolute time as the TimeValue parameter.

**Parameters:**

*t* A reference to a TimeValue object.

### 6.37.2.6 OSCL\_COND\_IMPORT\_REF NTPTime::NTPTime (const uint64 value)

Construct a NTPTime instance from a uint64 value.

**Parameters:**

*value* A 64 bit integer argument which is used as the ntp 32.32 fractional representation.

## 6.37.3 Member Function Documentation

### 6.37.3.1 OSCL\_COND\_IMPORT\_REF uint32 NTPTime::get\_lower32 ()

This method returns the lower 32 bits of the 32.32 representation.

### 6.37.3.2 OSCL\_COND\_IMPORT\_REF uint32 NTPTime::get\_middle32 ()

Grab the middle 32 bits of the 64 bit 32.32 representation.

### 6.37.3.3 OSCL\_COND\_IMPORT\_REF uint32 NTPTime::get\_upper32 ()

This method returns the upper 32 bits of the 32.32 representation.

### 6.37.3.4 OSCL\_COND\_IMPORT\_REF uint64 NTPTime::get\_value ()

This method returns the 32.32 ntp representation.

### 6.37.3.5 OSCL\_COND\_IMPORT\_REF NTPTime& NTPTime::operator+= (uint64 val)

The += operator is used to add a 64 bit 32.32 value to an existing NTPTime value.

**Parameters:**

*val* The 64 bit 32.32 value to add to this object's value.

### 6.37.3.6 OSCL\_COND\_IMPORT\_REF NTPTime NTPTime::operator- (const NTPTime & npt) const

The - operator allows subtraction of one NTPTime value from another. This is useful to measure an interval.

**Parameters:**

*npt* A reference to the NTPTime object to be subtracted from this one.

**6.37.3.7 OSCL\_COND\_IMPORT\_REF NTPTime& NTPTime::operator= (*uint64 newval*)**

The assignment operator for a 64 bit integer.

**Parameters:**

*newval* A 64 bit value which represents the 32.32 fractional representation of the ntp time.

**6.37.3.8 OSCL\_COND\_IMPORT\_REF NTPTime& NTPTime::operator= (*uint32 newval*)**

The assignment operator for a 32 bit integer.

**Parameters:**

*newval* A 32 bit integer representing the upper 32 bits of the 32.32 NTP time (e.g. the number of whole seconds since Jan 1, 1900 UTC).

**6.37.3.9 void NTPTime::set\_from\_system\_time (*const uint32 systemtime*)**

This method converts a 32-bit system time to NTP time.

This method sets the value of the NTPTime instance to the absolute time represented by the 32 bit input argument.

**Parameters:**

*systemtime* This 32-bit value is interpreted as the number of seconds since the unix epoch Jan. 1 1970.

**6.37.3.10 OSCL\_IMPORT\_REF int NTPTime::set\_to\_current\_time ()**

This method sets the 32.32 representation to the current system time value.

**6.37.3.11 int32 NTPTime::to\_system\_time ()**

This method converts the ntp time value to system time.

This method returns a 32 bit value representing the same absolute time as the NTP time value, but expressed as whole seconds since the unix epoch. The fractional part of the ntp value is discarded.

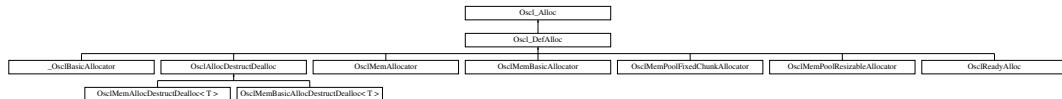
The documentation for this class was generated from the following file:

- [oscl\\_time.h](#)

## 6.38 Oscl\_Alloc Class Reference

```
#include <oscl_defalloc.h>
```

Inheritance diagram for Oscl\_Alloc::



### Public Methods

- virtual [OsclAny \\* allocate \(const uint32 size\)=0](#)
- virtual [OsclAny \\* allocate\\_fl \(const uint32 size, const char \\*file\\_name, const int line\\_num\)](#)

#### 6.38.1 Member Function Documentation

##### 6.38.1.1 virtual [OsclAny\\* Oscl\\_Alloc::allocate \(const uint32 size\)](#) [pure virtual]

Implemented in [\\_OsclBasicAllocator](#), [Oscl\\_DefAlloc](#), [OsclMemAllocator](#), [OsclMemBasicAllocator](#), [OsclMemAllocDestructDealloc< T >](#), [OsclMemBasicAllocDestructDealloc< T >](#), [OsclMemPoolFixedChunkAllocator](#), [OsclMemPoolResizableAllocator](#), and [OsclReadyAlloc](#).

##### 6.38.1.2 virtual [OsclAny\\* Oscl\\_Alloc::allocate\\_fl \(const uint32 size, const char \\*file\\_name, const int line\\_num\)](#) [inline, virtual]

Reimplemented in [Oscl\\_DefAlloc](#), [OsclMemAllocator](#), [OsclMemAllocDestructDealloc< T >](#), and [OsclReadyAlloc](#).

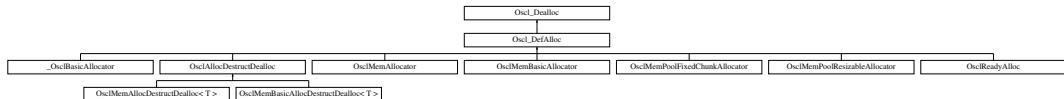
The documentation for this class was generated from the following file:

- [oscl\\_defalloc.h](#)

## 6.39 Oscl\_Dealloc Class Reference

```
#include <oscl_defalloc.h>
```

Inheritance diagram for Oscl\_Dealloc::



### Public Methods

- virtual void [deallocate \(OsclAny \\*p\)=0](#)

#### 6.39.1 Member Function Documentation

##### 6.39.1.1 virtual void Oscl\_Dealloc::deallocate (OsclAny \*p) [pure virtual]

Implemented in [\\_OsclBasicAllocator](#), [Oscl\\_DefAlloc](#), [OsclMemAllocator](#), [OsclMemBasicAllocator](#), [OsclMemAllocDestructDealloc< T >](#), [OsclMemBasicAllocDestructDealloc< T >](#), [OsclMemPoolFixedChunkAllocator](#), [OsclMemPoolResizableAllocator](#), and [OsclReadyAlloc](#).

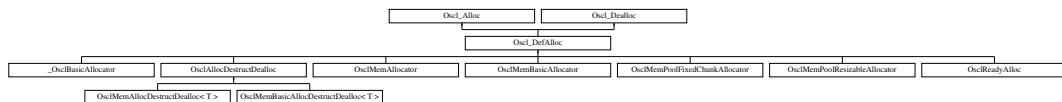
The documentation for this class was generated from the following file:

- [oscl\\_defalloc.h](#)

## 6.40 Oscl\_DefAlloc Class Reference

```
#include <oscl_defalloc.h>
```

Inheritance diagram for Oscl\_DefAlloc::



### Public Methods

- virtual [OsclAny \\* allocate \(const uint32 size\)=0](#)
- virtual [OsclAny \\* allocate\\_fl \(const uint32 size, const char \\*file\\_name, const int line\\_num\)](#)
- virtual void [deallocate \(OsclAny \\*p\)=0](#)

#### 6.40.1 Member Function Documentation

##### 6.40.1.1 virtual [OsclAny\\* Oscl\\_DefAlloc::allocate \(const uint32 size\)](#) [pure virtual]

Implements [Oscl\\_Alloc](#).

Implemented in [\\_OsclBasicAllocator](#), [OsclMemAllocator](#), [OsclMemBasicAllocator](#), [OsclMemAlloc-DestructDealloc< T >](#), [OsclMemBasicAllocDestructDealloc< T >](#), [OsclMemPoolFixedChunkAllocator](#), [OsclMemPoolResizableAllocator](#), and [OsclReadyAlloc](#).

##### 6.40.1.2 virtual [OsclAny\\* Oscl\\_DefAlloc::allocate\\_fl \(const uint32 size, const char \\*file\\_name, const int line\\_num\)](#) [inline, virtual]

Reimplemented from [Oscl\\_Alloc](#).

Reimplemented in [OsclMemAllocator](#), [OsclMemAllocDestructDealloc< T >](#), and [OsclReadyAlloc](#).

##### 6.40.1.3 virtual void [Oscl\\_DefAlloc::deallocate \(OsclAny \\*p\)](#) [pure virtual]

Implements [Oscl\\_Dealloc](#).

Implemented in [\\_OsclBasicAllocator](#), [OsclMemAllocator](#), [OsclMemBasicAllocator](#), [OsclMemAlloc-DestructDealloc< T >](#), [OsclMemBasicAllocDestructDealloc< T >](#), [OsclMemPoolFixedChunkAllocator](#), [OsclMemPoolResizableAllocator](#), and [OsclReadyAlloc](#).

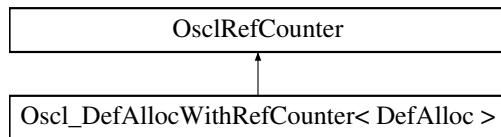
The documentation for this class was generated from the following file:

- [oscl\\_defalloc.h](#)

## 6.41 Oscl\_DefAllocWithRefCounter< DefAlloc > Class Template Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for Oscl\_DefAllocWithRefCounter< DefAlloc >::



### Public Methods

- void [Delete](#) ()
- void [addRef](#) ()
- void [removeRef](#) ()
- uint32 [getCount](#) ()

### Static Public Methods

- Oscl\_DefAllocWithRefCounter \* [New](#) ()

#### 6.41.1 Detailed Description

**template<class DefAlloc> class Oscl\_DefAllocWithRefCounter< DefAlloc >**

Implementation of an [Oscl\\_DefAlloc](#) class with a built-in ref counter.

#### 6.41.2 Member Function Documentation

**6.41.2.1 template<class DefAlloc> void Oscl\_DefAllocWithRefCounter< DefAlloc >::addRef () [inline, virtual]**

Add to the reference count

Implements [OsclRefCounter](#).

**6.41.2.2 template<class DefAlloc> void Oscl\_DefAllocWithRefCounter< DefAlloc >::Delete () [inline]**

Delete object

**6.41.2.3 template<class DefAlloc> uint32 Oscl\_DefAllocWithRefCounter< DefAlloc >::getCount () [inline, virtual]**

Gets the current number of references

Implements [OsclRefCounter](#).

**6.41.2.4 template<class DefAlloc> Oscl\_DefAllocWithRefCounter\*<br/>Oscl\_DefAllocWithRefCounter< DefAlloc >::New () [inline, static]**

Create object

**6.41.2.5 template<class DefAlloc> void Oscl\_DefAllocWithRefCounter< DefAlloc >::removeRef<br/>() [inline, virtual]**

Delete from reference count

Implements [OsclRefCounter](#).

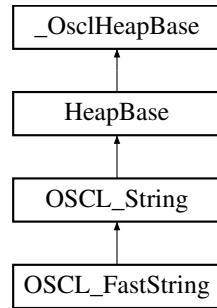
The documentation for this class was generated from the following file:

- [oscl\\_refcounter.h](#)

## 6.42 OSCL\_FastString Class Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL\_FastString::



### Public Types

- `typedef OSCL_String::chartype chartype`

### Public Methods

- `OSCL_IMPORT_REF OSCL_FastString()`
- `OSCL_IMPORT_REF OSCL_FastString(const OSCL_FastString &src)`
- `OSCL_IMPORT_REF OSCL_FastString(const chartype *cstr)`
- `OSCL_IMPORT_REF OSCL_FastString(chartype *buf, uint32 maxlen)`
- `OSCL_IMPORT_REF ~OSCL_FastString()`
- `OSCL_IMPORT_REF uint32 get_size() const`
- `OSCL_IMPORT_REF uint32 get_maxsize() const`
- `OSCL_IMPORT_REF const chartype * get_cstr() const`
- `OSCL_IMPORT_REF chartype * get_str() const`
- `OSCL_IMPORT_REF OSCL_FastString & operator=(const OSCL_FastString &src)`
- `OSCL_IMPORT_REF OSCL_FastString & operator=(const chartype *cstr)`
- `OSCL_IMPORT_REF void set(chartype *cstr, uint32 maxlen)`
- `OSCL_IMPORT_REF void set_length()`

### Friends

- class `OSCL_String`

#### 6.42.1 Detailed Description

`OSCL_FastString` is a simple string class, compatible with regular character array strings.

This class does not allocate internal memory for the string but acts as a container for a user-defined buffer. This means no copying of the string is done and provides a faster way of manipulating strings. Depending on initialization, this container provides either read-only or read-write access to the string.

Implementation assumes the input string is null-terminated.

**Parameters:**

*C*: type of character.

## 6.42.2 Member Typedef Documentation

### 6.42.2.1 `typedef OSCL_String::chartype OSCL_FastString::chartype`

Reimplemented from [OSCL\\_String](#).

## 6.42.3 Constructor & Destructor Documentation

### 6.42.3.1 `OSCL_IMPORT_REF OSCL_FastString::OSCL_FastString()`

Default constructor.

### 6.42.3.2 `OSCL_IMPORT_REF OSCL_FastString::OSCL_FastString(const OSCL_FastString & src)`

Creates a fast string that contains a copy of the input string. The string inherits the writable-ness of the source string.

**Parameters:**

*src*: input string.

### 6.42.3.3 `OSCL_IMPORT_REF OSCL_FastString::OSCL_FastString(const chartype * cstr)`

Create the string and initialize it to contain the input string. The string is not writable.

**am:** null-terminated string.

### 6.42.3.4 `OSCL_IMPORT_REF OSCL_FastString::OSCL_FastString(chartype * buf, uint32 maxlen)`

Create the string and initialize it to contain the input string. The string is writable.

**Parameters:**

*cp*: null-terminated string.

*maxlen*: maximum size of storage at *cp*, not incl null terminator. If input string is not null-terminated, the function leaves.

### 6.42.3.5 `OSCL_IMPORT_REF OSCL_FastString::~OSCL_FastString()`

## 6.42.4 Member Function Documentation

### 6.42.4.1 `OSCL_IMPORT_REF const chartype* OSCL_FastString::get_cstr() [virtual]`

This function returns the C-style string for read access.

Implements [OSCL\\_String](#).

**6.42.4.2 OSCL\_IMPORT\_REF uint32 OSCL\_FastString::get\_maxsize () [virtual]**

This function returns the maximum available storage size, not including null terminator. The maximum size may be larger than the current string size.

Implements [OSCL\\_String](#).

**6.42.4.3 OSCL\_IMPORT\_REF uint32 OSCL\_FastString::get\_size () [virtual]**

Pure virtuals from [OSCL\\_String](#)

Implements [OSCL\\_String](#).

**6.42.4.4 OSCL\_IMPORT\_REF chartype\* OSCL\_FastString::get\_str () [virtual]**

This function returns the C-style string for write access. If the string is not writable it returns NULL.

Implements [OSCL\\_String](#).

**6.42.4.5 OSCL\_IMPORT\_REF OSCL\_FastString& OSCL\_FastString::operator= (const chartype \* cstr)**

Assignment operator

**am:** null-terminated string

Reimplemented from [OSCL\\_String](#).

**6.42.4.6 OSCL\_IMPORT\_REF OSCL\_FastString& OSCL\_FastString::operator= (const OSCL\_FastString & src)**

Assignment operators

**6.42.4.7 OSCL\_IMPORT\_REF void OSCL\_FastString::set (chartype \* cstr, uint32 maxlen)**

This function can be used to reassign the string to a new writable string. If input string is not null-terminated, the function leaves.

**6.42.4.8 OSCL\_IMPORT\_REF void OSCL\_FastString::set\_length ()**

This function can be used to refresh the string size in case the contents of the string buffer have been modified since the container was created.

**6.42.5 Friends And Related Function Documentation****6.42.5.1 friend class OSCL\_String [friend]**

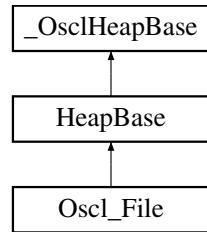
The documentation for this class was generated from the following file:

- [oscl\\_string\\_containers.h](#)

## 6.43 Oscl\_File Class Reference

```
#include <oscl_file_io.h>
```

Inheritance diagram for Oscl\_File::



### Public Types

- enum `seek_type` { `SEEKSET`, `SEEKCUR`, `SEEKEND` }
- enum `mode_type` { `MODE_READ` = 0x0001, `MODE_READWRITE` = 0x0002, `MODE_APPEND` = 0x0004, `MODE_BINARY` = 0x0008, `MODE_TEXT` = 0x0010, `MODE_READ_PLUS` = 0x0020 }
- enum `TSymbianAccessMode` { `ESymbianAccessMode_Rfile` = 0, `ESymbianAccessMode_RfileBuf` = 1 }

### Public Methods

- `OSCL_IMPORT_REF Oscl_File()`
- `OSCL_IMPORT_REF Oscl_File(uint32 aCacheSize)`
- `OSCL_IMPORT_REF Oscl_File(uint32 aCacheSize, OsclFileHandle *aFileHandle)`
- `OSCL_IMPORT_REF ~Oscl_File()`
- `OSCL_IMPORT_REF void SetPVCacheSize(uint32 aSize)`
- `void AddFixedCache(const OsclFixedCacheParam &aParam)`
- `void RemoveFixedCache(const TOsclFileOffset &aPos)`
- `void SetCacheObserver(OsclCacheObserver *aObs)`
- `OSCL_IMPORT_REF void SetNativeAccessMode(uint32 aMode)`
- `OSCL_IMPORT_REF void SetNativeBufferSize(int32 aSize)`
- `OSCL_IMPORT_REF void SetAsyncReadBufferSize(uint32 aSize)`
- `OSCL_IMPORT_REF int32 SetFileHandle(OsclFileHandle *aHandle)`
- `OSCL_IMPORT_REF int32 Open(const char *filename, uint32 mode, Oscl_FileServer &fileserv)`
- `OSCL_IMPORT_REF int32 Open(const oscl_wchar *filename, uint32 mode, Oscl_FileServer &fileserv)`
- `OSCL_IMPORT_REF uint32 Read(OsclAny *buffer, uint32 size, uint32 numelements)`
- `OSCL_IMPORT_REF uint32 Write(const OsclAny *buffer, uint32 size, uint32 numelements)`
- `OSCL_IMPORT_REF int32 Seek(TOscOfFileOffset offset, seek_type origin)`
- `OSCL_IMPORT_REF TOscOfFileOffset Tell()`
- `OSCL_IMPORT_REF int32 Close()`
- `OSCL_IMPORT_REF int32 Flush()`
- `OSCL_IMPORT_REF int32 EndOfFile()`
- `OSCL_IMPORT_REF int32 GetError()`
- `OsclFileHandle * Handle()`

- OSCL\_IMPORT\_REF TOsclFileOffset Size ()
- OSCL\_IMPORT\_REF void SetLoggingEnable (bool aEnable)
- OSCL\_IMPORT\_REF void SetSummaryStatsLoggingEnable (bool aEnable)

## Friends

- class OsclFileCache
- class OsclFileCacheBuffer
- class asynccfilereadwrite\_test
- class largeasynccfilereadwrite\_test
- class asynccfilereadcancel\_test

### 6.43.1 Member Enumeration Documentation

#### 6.43.1.1 enum Oscl\_File::mode\_type

##### Enumeration values:

**MODE\_READ** Opens a file for reading. The file must exist.

**MODE\_READWRITE** Opens the file for reading and writing. If the file exists, its contents will be overwritten unless APPEND mode is specified. If the file does not exist, it will be created.

**MODE\_APPEND** Specifies all write operations to occur at the end of the file. The file pointer can be moved with the Seek command, but will always be moved to the end of the file for write commands.

**MODE\_BINARY** Opens the file in 'binary' mode. This is the default.

**MODE\_TEXT** Opens the file in 'text' mode. The default mode is 'binary'.

**MODE\_READ\_PLUS** Open a file for reading and writing. The file must exist. The default mode is 'binary'.

#### 6.43.1.2 enum Oscl\_File::seek\_type

##### Enumeration values:

**SEEKSET** Beginning of file

**SEEKCUR** Current position of file pointer

**SEEKEND** End of file

#### 6.43.1.3 enum Oscl\_File::TSymbianAccessMode

Defines mode options for SetNativeAccessMode on Symbian.

##### Enumeration values:

**ESymbianAccessMode\_Rfile**

**ESymbianAccessMode\_RfileBuf**

### 6.43.2 Constructor & Destructor Documentation

#### 6.43.2.1 OSCL\_IMPORT\_REF Oscl\_File::Oscl\_File ()

Constructor

#### 6.43.2.2 OSCL\_IMPORT\_REF Oscl\_File::Oscl\_File (uint32 *aCacheSize*)

Deprecated Constructor, present for back-compatibility.

**Parameters:**

*aCacheSize*: sets native buffer size, and when pv cache is enabled, also sets pv cache size.

#### 6.43.2.3 OSCL\_IMPORT\_REF Oscl\_File::Oscl\_File (uint32 *aCacheSize*, **OsclFileHandle** \* *aFileHandle*)

Deprecated Constructor, present for back-compatibility.

**Parameters:**

*aCacheSize*: sets native buffer size, and when pv cache is enabled, also sets pv cache size.

*aFileHandle*: open file handle.

#### 6.43.2.4 OSCL\_IMPORT\_REF Oscl\_File::~Oscl\_File ()

Destructor

### 6.43.3 Member Function Documentation

#### 6.43.3.1 void Oscl\_File::AddFixedCache (const **OsclFixedCacheParam** & *aParam*) [inline]

AddFixedCache adds a fixed cache. The fixed cache will be used on the next opportunity. The fixed cache must not overlap with any other fixed cache.

**Parameters:**

*aParam*: Cache location and size.

#### 6.43.3.2 OSCL\_IMPORT\_REF int32 Oscl\_File::Close ()

The File Close operation Closes the file after flushing any remaining data in the buffers.

Note: If the file object was opened with an external file handle, then Close will simply flush the file. The file will remain open.

**Returns:**

returns 0 if successful, and a non-zero value otherwise

#### 6.43.3.3 OSCL\_IMPORT\_REF int32 Oscl\_File::EndOfFile ()

The File EOF(end of file) operation returns a nonzero value after the first read operation that attempts to read past the end of the file

**Returns:**

#### 6.43.3.4 OSCL\_IMPORT\_REF int32 Oscl\_File::Flush ()

The File Flush operation On an output stream OSCL\_FileFlush causes any buffered but unwritten data to be written to the file.

**Returns:**

returns 0 if successful, and a non-zero value otherwise

#### 6.43.3.5 OSCL\_IMPORT\_REF int32 Oscl\_File::GetError ()

The File Error operation If no error has occurred on stream, returns 0. Otherwise, it returns a nonzero value

**Returns:**

#### 6.43.3.6 OsclFileHandle\* Oscl\_File::Handle () [inline]

Retrieve the file handle.

**Returns:**

file handle

#### 6.43.3.7 OSCL\_IMPORT\_REF int32 Oscl\_File::Open (const oscl\_wchar \*filename, uint32 mode, Oscl\_FileServer &fileserv)

Opens a file.

Note: when an external file handle is used, Open will attach to the file handle and initialize cacheing features, but will not do a native file open.

**Parameters:**

*filename* name of file to open (Unicode)

*mode* combination of open mode flags

*fileserv* fileserv to use

**Returns:**

returns 0 if successful and a non-zero value otherwise

#### 6.43.3.8 OSCL\_IMPORT\_REF int32 Oscl\_File::Open (const char \*filename, uint32 mode, Oscl\_FileServer &fileserv)

Opens a file.

Note: when an external file handle is used, Open will attach to the file handle and initialize cacheing features, but will not do a native file open.

**Parameters:**

*filename* name of file to open (Utf8)

*mode* combination of open mode flags

*fileserv* fileserv to use

**Returns:**

returns 0 if successful and a non-zero value otherwise

#### 6.43.3.9 OSCL\_IMPORT\_REF uint32 Oscl\_File::Read (*OsclAny \* buffer, uint32 size, uint32 numelements*)

The File Read operation Reads from the file into the buffer a maximum of 'numelements' of size 'size'.

**Parameters:**

*buffer* pointer to buffer of type void

*size* element size in bytes

*numelements* max number of elements to read

**Returns:**

returns the number of full elements actually read, which may be less than count if an error occurs or if the end of the file is encountered before reaching count. Use the CheckEndOfFile or GetError function to distinguish a read error from an end-of-file condition.

#### 6.43.3.10 void Oscl\_File::RemoveFixedCache (const **TOsclFileOffset & aPos**) [ inline ]

RemoveFixedCache removes a fixed cache.

**Parameters:**

*aPos*: Cache location and size.

#### 6.43.3.11 OSCL\_IMPORT\_REF int32 Oscl\_File::Seek (*TOsclFileOffset offset, seek\_type origin*)

The File Seek operation Sets the position for file pointer

**Parameters:**

*offset* offset from the specified origin.

*origin* starting point

**Returns:**

returns 0 on success, and a non-zero value otherwise

#### 6.43.3.12 OSCL\_IMPORT\_REF void Oscl\_File::SetAsyncReadBufferSize (uint32 *aSize*)

SetAsyncReadBufferSize configures the asynchronous background read function. May not be available on all platforms.

This should be called before opening the file. If used when the file is open, the option will not take effect until the next Open.

Note: if asynchronous read is not available on the platform, this call will have no effect.

**Parameters:**

*aSize*: buffer size in bytes. Zero disables the feature.

6.43.3.13 void Oscl\_File::SetCacheObserver ([OsclCacheObserver](#) \* *aObs*) [inline]

6.43.3.14 OSCL\_IMPORT\_REF int32 Oscl\_File::SetFileHandle ([OsclFileHandle](#) \* *aHandle*)

SetFileHandle adds an open file handle to the Oscl\_File object. The Oscl\_File object will use that handle to access the file.

This call is not available when the Oscl\_File object is already open.

Note: This feature is used in Symbian with the MMF framework. The MMF framework provides an open RFile handle to access content. When using RFileBuf access mode with an RFile handle, the RFileBuf will be attached to the open RFile handle.

To use the external file handle, the caller starts with a native file handle to an open file. The caller must wrap the native file handle in an [OsclFileHandle](#) object, pass the [OsclFileHandle](#) pointer to SetFileHandle, call [Oscl\\_File::Open](#), then proceed to use the Oscl\_File object, finally calling [Oscl\\_File::Close](#). In this usage mode, [Oscl\\_File::Open](#) and [Oscl\\_File::Close](#) do not actually call native file open and close. It is assumed that the caller will close the original native file handle after usage is complete.

**Parameters:**

*aHandle*: container for an open file handle.

**Returns:**

returns 0 if successful, non-zero if error.

6.43.3.15 OSCL\_IMPORT\_REF void Oscl\_File::SetLoggingEnable (bool *aEnable*)

SetLoggingEnable configures the [PVLogger](#) output for this file. This will enable full logging of each API entry and exit using the logger object "Oscl\_File", plus full logging of native operation entry & exit using logger object "[OsclNativeFile](#)".

**Parameters:**

*aEnable*: true to enable, false to disable logging.

6.43.3.16 OSCL\_IMPORT\_REF void Oscl\_File::SetNativeAccessMode (uint32 *aMode*)

SetNativeAccessMode allows switching between different native file access modes, when available.

Note: for Symbian, use the TSymbianAccessMode values to choose the mode. If multiple access modes are not available on the platform, this call will have no effect.

**Parameters:**

*aMode*: access mode.

6.43.3.17 OSCL\_IMPORT\_REF void Oscl\_File::SetNativeBufferSize (int32 *aSize*)

SetNativeBufferSize configures the native file buffering feature, when available.

This should be called before opening the file. If used when the file is open, the option will not take effect until the next Open.

Note: For Symbian, this sets the RFileBuf cache size. If native buffering is not available on the platform, this call will have no effect.

**Parameters:**

*aSize*: native buffer size in bytes. Zero disables the feature.

**6.43.3.18 OSCL\_IMPORT\_REF void Oscl\_File::SetPVCacheSize (uint32 *aSize*)**

SetPVCacheSize configures the read/write cache.

This should be called before opening the file. If used when the file is open, the option will not take effect until the next Open.

**Parameters:**

*aSize*: cache size in bytes. Zero disables the cache.

**6.43.3.19 OSCL\_IMPORT\_REF void Oscl\_File::SetSummaryStatsLoggingEnable (bool *aEnable*)**

SetSummaryStatsLoggingEnable configures the [PVLogger](#) output for this file. This will enable summary statistics logging only, using the logger object "[OsclFileStats](#)".

**Parameters:**

*aEnable*: true to enable, false to disable stats logging.

**6.43.3.20 OSCL\_IMPORT\_REF TOsclFileOffset Oscl\_File::Size ()**

Get the file size in bytes.

**Returns:**

- The size of the file, or -1 on error.

**6.43.3.21 OSCL\_IMPORT\_REF TOsclFileOffset Oscl\_File::Tell ()**

The File Tell operation Returns the current file position for file specified by fp

**6.43.3.22 OSCL\_IMPORT\_REF uint32 Oscl\_File::Write (const OsclAny \* *buffer*, uint32 *size*, uint32 *numelements*)**

The File Write operation Writes from the buffer '*numelements*' objects of size '*size*'

**Parameters:**

*buffer* pointer to buffer of type void

*size* element size in bytes

*numelements* number of elements to write

**Returns:**

The number of elements written

#### 6.43.4 Friends And Related Function Documentation

6.43.4.1 **friend class `asyncfilereadcancel_test` [friend]**

6.43.4.2 **friend class `asyncfilereadwrite_test` [friend]**

6.43.4.3 **friend class `largeasyncfilereadwrite_test` [friend]**

6.43.4.4 **friend class `OsclFileCache` [friend]**

6.43.4.5 **friend class `OsclFileCacheBuffer` [friend]**

The documentation for this class was generated from the following file:

- [oscl\\_file\\_io.h](#)

## 6.44 Oscl\_File::OsclCacheObserver Class Reference

```
#include <oscl_file_io.h>
```

### Public Methods

- virtual **OsclFileCacheBuffer \* ChooseCurCache (OsclFileCache &aContext, TOsclFileOffset aPos)=0**

#### 6.44.1 Detailed Description

For defining a cache observer. Cache observer can implement customized cache schemes by replacing the SetCachePosition routine.

#### 6.44.2 Member Function Documentation

##### 6.44.2.1 virtual **OsclFileCacheBuffer\* Oscl\_File::OsclCacheObserver::ChooseCurCache (OsclFileCache & aContext, TOsclFileOffset aPos) [pure virtual]**

The documentation for this class was generated from the following file:

- [oscl\\_file\\_io.h](#)

## 6.45 Oscl\_File::OsclFixedCacheParam Class Reference

```
#include <oscl_file_io.h>
```

### Public Methods

- bool [Contains \(TOsclFileOffset pos\) const](#)

### Data Fields

- [TOsclFileOffset iFilePosition](#)
- [uint32 iSize](#)

#### 6.45.1 Detailed Description

Parameters for defining a fixed cache

#### 6.45.2 Member Function Documentation

**6.45.2.1 bool Oscl\_File::OsclFixedCacheParam::Contains (TOsclFileOffset *pos*) const  
[inline]**

#### 6.45.3 Field Documentation

**6.45.3.1 TOsclFileOffset Oscl\_File::OsclFixedCacheParam::iFilePosition**

**6.45.3.2 uint32 Oscl\_File::OsclFixedCacheParam::iSize**

The documentation for this class was generated from the following file:

- [oscl\\_file\\_io.h](#)

## 6.46 Oscl\_FileFind Class Reference

```
#include <oscl_file_find.h>
```

### Public Types

- enum `error_type` { `E_OK` = 0, `E_INVALID_STATE`, `E_INVALID_ARG`, `E_PATH_TOO_LONG`, `E_PATH_NOT_FOUND`, `E_NO_MATCH`, `E_BUFFER_TOO_SMALL`, `E_NOT_IMPLEMENTED`, `E_MEMORY_ERROR`, `E_OTHER` }
- enum `element_type` { `FILE_TYPE` = 0, `DIR_TYPE`, `INVALID_TYPE` }

### Public Methods

- OSCL\_IMPORT\_REF const char \* `FindFirst` (const char \*directory, const char \*pattern, char \*buf, uint32 buflen)
- OSCL\_IMPORT\_REF const `oscl_wchar` \* `FindFirst` (const `oscl_wchar` \*directory, const `oscl_wchar` \*pattern, `oscl_wchar` \*buf, uint32 buflen)
- OSCL\_IMPORT\_REF char \* `FindNext` (char \*buf, uint32 buflen)
- OSCL\_IMPORT\_REF `oscl_wchar` \* `FindNext` (`oscl_wchar` \*buf, uint32 buflen)
- OSCL\_IMPORT\_REF void `Close` ()
- OSCL\_IMPORT\_REF `element_type` `GetElementType` ()
- OSCL\_IMPORT\_REF `error_type` `GetLastError` ()
- OSCL\_IMPORT\_REF `Oscl_FileFind` ()
- OSCL\_IMPORT\_REF `~Oscl_FileFind` ()

### 6.46.1 Detailed Description

`Oscl_FileFind` class defines the generic way of finding filesystem elements that match a pattern within a directory

### 6.46.2 Member Enumeration Documentation

#### 6.46.2.1 enum Oscl\_FileFind::element\_type

Enumeration values:

`FILE_TYPE`

`DIR_TYPE`

`INVALID_TYPE`

#### 6.46.2.2 enum Oscl\_FileFind::error\_type

Enumeration values:

`E_OK`

`E_INVALID_STATE`

`E_INVALID_ARG`

`E_PATH_TOO_LONG`

**E\_PATH\_NOT\_FOUND**  
**E\_NO\_MATCH**  
**E\_BUFFER\_TOO\_SMALL**  
**E\_NOT\_IMPLEMENTED**  
**E\_MEMORY\_ERROR**  
**E\_OTHER**

### 6.46.3 Constructor & Destructor Documentation

#### 6.46.3.1 OSCL\_IMPORT\_REF Oscl\_FileFind::Oscl\_FileFind ()

constructor.

**Returns:**

none

#### 6.46.3.2 OSCL\_IMPORT\_REF Oscl\_FileFind::~Oscl\_FileFind ()

destructor. will deallocate open handles if necessary

**Returns:**

none

### 6.46.4 Member Function Documentation

#### 6.46.4.1 OSCL\_IMPORT\_REF void Oscl\_FileFind::Close ()

closes the handle to directory.

**Returns:**

none

#### 6.46.4.2 OSCL\_IMPORT\_REF const oscl\_wchar\* Oscl\_FileFind::FindFirst (const oscl\_wchar \* directory, const oscl\_wchar \* pattern, oscl\_wchar \* buf, uint32 buflen)

Opens a directory for reading.

**Parameters:**

*directory* directory to search (utf16).

*pattern* wildcard pattern filter (utf16). passing NULL, results in a universal match.

*buf* buffer for returned pathname (utf16).

*buflen* size in wide characters of buf. If buf is not large enough to hold the returned string, NULL is returned, and GetLastError is set to E\_BUFFER\_TOO\_SMALL.

**Returns:**

returns a pointer to buffer supplied, which contains the pathname of the next found element, or NULL otherwise. On a NULL return value, [GetLastError\(\)](#) returns a more detailed error.

**6.46.4.3 OSCL\_IMPORT\_REF const char\* Oscl\_FileFind::FindFirst (const char \* *directory*, const char \* *pattern*, char \* *buf*, uint32 *buflen*)**

Finds first element matching the pattern.

**Parameters:**

*directory* directory to search (utf8).

*pattern* wildcard pattern filter (utf8). passing NULL, results in a universal match.

*buf* buffer for returned pathname (utf8).

*buflen* size in wide characters of buf. If buf is not large enough to hold the returned string, NULL is returned, and GetLastError is set to E\_BUFFER\_TOO\_SMALL.

**Returns:**

returns a pointer to buffer supplied, which contains the pathname of the next found element, or NULL otherwise. On a NULL return value, [GetLastError\(\)](#) returns a more detailed error.

**6.46.4.4 OSCL\_IMPORT\_REF oscl\_wchar\* Oscl\_FileFind::FindNext (oscl\_wchar \* *buf*, uint32 *buflen*)**

Reads the next element in a directory. Note: the pointer returned by this function is not persistent and should be stored. Its scope is limited to the lifetime of the class.

**Parameters:**

*buf* buffer to hold directory name(utf16)

*buflen* size in wide characters of buf. If buf is not large enough to hold the returned string, NULL is returned, and GetLastError is set to E\_BUFFER\_TOO\_SMALL.

**Returns:**

returns a pointer to buffer supplied, which contains the pathname of the next found element, or NULL otherwise. On a NULL return value, [GetLastError\(\)](#) returns a more detailed error.

**6.46.4.5 OSCL\_IMPORT\_REF char\* Oscl\_FileFind::FindNext (char \* *buf*, uint32 *buflen*)**

Reads the next element in the directory. Note: the pointer returned by this function is not persistent and should be stored. Its scope is limited to the lifetime of the class.

**Parameters:**

*buf* buffer to hold directory name(utf8)

*buflen* size in wide characters of buf. If buf is not large enough to hold the returned string, NULL is returned, and GetLastError is set to E\_BUFFER\_TOO\_SMALL.

**Returns:**

returns a pointer to buffer supplied, which contains the pathname of the next found element, or NULL otherwise. On a NULL return value, [GetLastError\(\)](#) returns a more detailed error.

**6.46.4.6 OSCL\_IMPORT\_REF element\_type Oscl\_FileFind::GetElementType ()**

Returns the element type for the last element returned

**Returns:**

see enumeration above for more info.

#### 6.46.4.7 OSCL\_IMPORT\_REF [error\\_type](#) Oscl\_FileFind::GetLastError ()

Returns the error code for the last operation.

**Returns:**

see enumeration above for more info.

The documentation for this class was generated from the following file:

- [oscl\\_file\\_find.h](#)

## 6.47 Oscl\_FileServer Class Reference

```
#include <oscl_file_server.h>
```

### Public Methods

- OSCL\_IMPORT\_REF [Oscl\\_FileServer\(\)](#)
- OSCL\_IMPORT\_REF [~Oscl\\_FileServer\(\)](#)
- OSCL\_IMPORT\_REF int32 [Connect\(\)](#)
- OSCL\_IMPORT\_REF int32 [Close\(\)](#)
- OSCL\_IMPORT\_REF int32 [Oscl\\_DeleteFile\(const char \\*filename\)](#)
- OSCL\_IMPORT\_REF int32 [Oscl\\_DeleteFile\(const oscl\\_wchar \\*filename\)](#)

### Friends

- class [Oscl\\_File](#)
- class [OsclNativeFile](#)

#### 6.47.1 Constructor & Destructor Documentation

##### 6.47.1.1 OSCL\_IMPORT\_REF Oscl\_FileServer::Oscl\_FileServer()

Constructor

##### 6.47.1.2 OSCL\_IMPORT\_REF Oscl\_FileServer::~Oscl\_FileServer()

Destructor

#### 6.47.2 Member Function Documentation

##### 6.47.2.1 OSCL\_IMPORT\_REF int32 Oscl\_FileServer::Close()

Closes a file server.

**Returns:**

returns 0 on success and a non-zero value otherwise

##### 6.47.2.2 OSCL\_IMPORT\_REF int32 Oscl\_FileServer::Connect()

Connects the server. This must be called before a file server can be used.

**Returns:**

returns 0 on success and a non-zero value otherwise

**6.47.2.3 OSCL\_IMPORT\_REF int32 Oscl\_FileServer::Oscl\_DeleteFile (const oscl\_wchar \*  
*filename*)**

Deletes a file from the filesystem

**Parameters:**

*filename* name of the file to delete (Unicode)

**Returns:**

returns 0 if successful, and a non-zero value otherwise.

**6.47.2.4 OSCL\_IMPORT\_REF int32 Oscl\_FileServer::Oscl\_DeleteFile (const char \**filename*)**

Deletes a file from the filesystem \*

**Parameters:**

*filename* name of the file to delete (Utf8)

**Returns:**

returns 0 if successful, and a non-zero value otherwise.

**6.47.3 Friends And Related Function Documentation****6.47.3.1 friend class Oscl\_File [friend]****6.47.3.2 friend class OsclNativeFile [friend]**

The documentation for this class was generated from the following file:

- [oscl\\_file\\_server.h](#)

## 6.48 oscl\_fsstat Struct Reference

```
#include <oscl_file_dir_utils.h>
```

### Data Fields

- [uint64 freebytes](#)
- [uint64 totalbytes](#)

#### 6.48.1 Field Documentation

##### 6.48.1.1 [uint64 oscl\\_fsstat::freebytes](#)

##### 6.48.1.2 [uint64 oscl\\_fsstat::totalbytes](#)

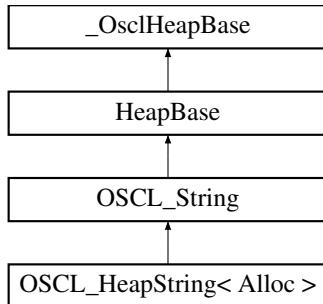
The documentation for this struct was generated from the following file:

- [oscl\\_file\\_dir\\_utils.h](#)

## 6.49 OSCL\_HeapString< Alloc > Class Template Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL\_HeapString< Alloc >::



### Public Types

- `typedef OSCL_String::chartype chartype`

### Public Methods

- `OSCL_HeapString()`
- `OSCL_HeapString(const OSCL_HeapString &src)`
- `OSCL_HeapString(const OSCL_String &src)`
- `OSCL_HeapString(const chartype *cstr)`
- `OSCL_HeapString(const chartype *buf, uint32 length)`
- `~OSCL_HeapString()`
- `uint32 get_size() const`
- `uint32 get_maxsize() const`
- `const chartype * get_cstr() const`
- `chartype * get_str() const`
- `OSCL_HeapString & operator=(const OSCL_HeapString &src)`
- `OSCL_HeapString & operator=(const OSCL_String &src)`
- `OSCL_HeapString & operator=(const chartype *cstr)`
- `void set(const chartype *buf, uint32 length)`

### Friends

- class `OSCL_String`

#### 6.49.1 Detailed Description

`template<class Alloc> class OSCL_HeapString< Alloc >`

`OSCL_HeapString` is a simple string class, compatible with regular character array strings.

The string array is variable length, is allocated from the heap, and is modifiable. A copy-on-write mechanism is used to minimize unnecessary copying when multiple instances of a string are created for reading.

Allocated memory is automatically freed by the class destructor when the last string referencing the memory is destroyed.

The class HAS NO thread synchronization built-in, so it is NOT MT-SAFE. External locks should be used if the class is to be shared across threads.

**Parameters:**

*Alloc*: memory allocator, derived from [Oscl\\_DefAlloc](#).

## 6.49.2 Member Typedef Documentation

### 6.49.2.1 template<class Alloc> typedef OSCL\_String::chartype OSCL\_HeapString< Alloc >::chartype

Reimplemented from [OSCL\\_String](#).

## 6.49.3 Friends And Related Function Documentation

### 6.49.3.1 template<class Alloc> friend class OSCL\_String [friend]

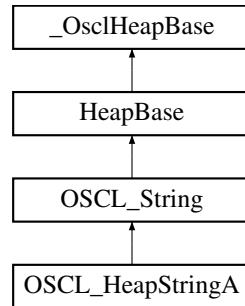
The documentation for this class was generated from the following file:

- [oscl\\_string\\_containers.h](#)

## 6.50 OSCL\_HeapStringA Class Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL\_HeapStringA::



### Public Types

- `typedef OSCL_String::chartype chartype`

### Public Methods

- `OSCL_IMPORT_REF OSCL_HeapStringA ()`
- `OSCL_IMPORT_REF OSCL_HeapStringA (Oscl_DefAlloc *alloc, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_HeapStringA (const OSCL_HeapStringA &src)`
- `OSCL_IMPORT_REF OSCL_HeapStringA (const OSCL_HeapStringA &src, Oscl_DefAlloc *alloc, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_HeapStringA (const OSCL_String &src, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_HeapStringA (const chartype *cstr, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_HeapStringA (const chartype *buf, uint32 length, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF ~OSCL_HeapStringA ()`
- `OSCL_IMPORT_REF uint32 get_size () const`
- `OSCL_IMPORT_REF uint32 get_maxsize () const`
- `OSCL_IMPORT_REF const chartype * get_cstr () const`
- `OSCL_IMPORT_REF chartype * get_str () const`
- `OSCL_IMPORT_REF OSCL_HeapStringA & operator= (const OSCL_HeapStringA &src)`
- `OSCL_IMPORT_REF OSCL_HeapStringA & operator= (const OSCL_String &src)`
- `OSCL_IMPORT_REF OSCL_HeapStringA & operator= (const chartype *cstr)`
- `OSCL_IMPORT_REF void set (const chartype *buf, uint32 length)`

### Friends

- class `OSCL_String`

### 6.50.1 Detailed Description

OSCL\_HeapStringA is a simple string class, compatible with regular character array strings. It is similar to [OSCL\\_HeapString](#), except that the allocator is passed at run-time instead of compile-time. The allocator pointer is passed in the constructor, and may be a reference-counted object. If the allocator is not a reference-counted object then it must persist over the lifetime of all OSCL\_HeapStringA objects that use it. If no allocator is provided, then an [OsclMemAllocator](#) will be used.

The string array is variable length, is allocated from the heap, and is modifiable. A copy-on-write mechanism is used to minimize unnecessary copying when multiple instances of a string are created for reading. Allocated memory is automatically freed by the class destructor when the last string referencing the memory is destroyed.

The class HAS NO thread synchronization built-in, so it is NOT MT-SAFE. External locks should be used if the class is to be shared across threads.

### 6.50.2 Member Typedef Documentation

#### 6.50.2.1 `typedef OSCL_String::chartype OSCL_HeapStringA::chartype`

Reimplemented from [OSCL\\_String](#).

### 6.50.3 Constructor & Destructor Documentation

#### 6.50.3.1 `OSCL_IMPORT_REF OSCL_HeapStringA::OSCL_HeapStringA()`

The default constructor creates an empty string.

**am:** (optional) allocator or reference-counted allocator.

**am:** (optional) reference counter associated with allocator object.

If no allocator is provided, this object will use an [OsclMemAllocator](#).

#### 6.50.3.2 `OSCL_IMPORT_REF OSCL_HeapStringA::OSCL_HeapStringA (Oscl_DefAlloc * alloc, OsclRefCounter * ref = NULL)`

#### 6.50.3.3 `OSCL_IMPORT_REF OSCL_HeapStringA::OSCL_HeapStringA (const OSCL_HeapStringA & src)`

Creates a heap string that contains a copy of the input string.

**Parameters:**

*src*: input string.

**am:** (optional) allocator or reference-counted allocator.

**am:** (optional) reference counter associated with allocator object.

If no allocator is provided, this object will use an [OsclMemAllocator](#).

- 6.50.3.4 OSCL\_IMPORT\_REF OSCL\_HeapStringA::OSCL\_HeapStringA (const OSCL\_HeapStringA & src, Oscl\_DefAlloc \* alloc, OsclRefCounter \* ref = NULL)**
- 6.50.3.5 OSCL\_IMPORT\_REF OSCL\_HeapStringA::OSCL\_HeapStringA (const OSCL\_String & src, Oscl\_DefAlloc \* alloc = NULL, OsclRefCounter \* ref = NULL)**
- 6.50.3.6 OSCL\_IMPORT\_REF OSCL\_HeapStringA::OSCL\_HeapStringA (const chartype \* cstr, Oscl\_DefAlloc \* alloc = NULL, OsclRefCounter \* ref = NULL)**

Creates a heap string that contains a copy of the input string.

**Parameters:**

*cp*: null-terminated string.

**am: (optional) allocator or reference-counted allocator.**

**am: (optional) reference counter associated with allocator object.**

If no allocator is provided, this object will use an [OsclMemAllocator](#).

- 6.50.3.7 OSCL\_IMPORT\_REF OSCL\_HeapStringA::OSCL\_HeapStringA (const chartype \* buf, uint32 length, Oscl\_DefAlloc \* alloc = NULL, OsclRefCounter \* ref = NULL)**

Creates a heap string that contains a copy of the input string or character array.

**Parameters:**

*src*: character array, not necessarily null-terminated.

*length*: number of characters to copy.

**am: (optional) allocator or reference-counted allocator.**

**am: (optional) reference counter associated with allocator object.**

If no allocator is provided, this object will use an [OsclMemAllocator](#).

- 6.50.3.8 OSCL\_IMPORT\_REF OSCL\_HeapStringA::~OSCL\_HeapStringA ()**

## 6.50.4 Member Function Documentation

- 6.50.4.1 OSCL\_IMPORT\_REF const chartype\* OSCL\_HeapStringA::get\_cstr () [virtual]**

This function returns the C-style string for read access.

Implements [OSCL\\_String](#).

- 6.50.4.2 OSCL\_IMPORT\_REF uint32 OSCL\_HeapStringA::get\_maxsize () [virtual]**

This function returns the maximum available storage size, not including null terminator. The maximum size may be larger than the current string size.

Implements [OSCL\\_String](#).

**6.50.4.3 OSCL\_IMPORT\_REF uint32 OSCL\_HeapStringA::get\_size () [virtual]**

Pure virtuals from [OSCL\\_String](#)

Implements [OSCL\\_String](#).

**6.50.4.4 OSCL\_IMPORT\_REF chartype\* OSCL\_HeapStringA::get\_str () [virtual]**

This function returns the C-style string for write access. If the string is not writable it returns NULL.

Implements [OSCL\\_String](#).

**6.50.4.5 OSCL\_IMPORT\_REF OSCL\_HeapStringA& OSCL\_HeapStringA::operator= (const chartype \* cstr)**

Assignment operator

**am:** null-terminated string

Reimplemented from [OSCL\\_String](#).

**6.50.4.6 OSCL\_IMPORT\_REF OSCL\_HeapStringA& OSCL\_HeapStringA::operator= (const OSCL\_String & src)**

Assignment operator

Reimplemented from [OSCL\\_String](#).

**6.50.4.7 OSCL\_IMPORT\_REF OSCL\_HeapStringA& OSCL\_HeapStringA::operator= (const OSCL\_HeapStringA & src)**

Assignment operators

**6.50.4.8 OSCL\_IMPORT\_REF void OSCL\_HeapStringA::set (const chartype \* buf, uint32 length)**

Set the contents of this string to a new string or character array.

**Parameters:**

*buf*: string or character array.

*length*: number of characters to copy.

**6.50.5 Friends And Related Function Documentation****6.50.5.1 friend class OSCL\_String [friend]**

The documentation for this class was generated from the following file:

- [oscl\\_string\\_containers.h](#)

## 6.51 Oscl\_Int64\_Utils Class Reference

The Oscl\_Int64\_Utils class provides a wrapper for commonly used int64/uint64 operations.

```
#include <oscl_int64_utils.h>
```

### Static Public Methods

- OSCL\_IMPORT\_REF void `set_int64` (`int64` &`input_value`, const `int32` `upper`, const `int32` `lower`)
- OSCL\_IMPORT\_REF `int32` `get_int64_upper32` (const `int64` &`input_value`)
- OSCL\_IMPORT\_REF `int32` `get_int64_lower32` (const `int64` &`input_value`)
- OSCL\_IMPORT\_REF `int32` `get_int64_middle32` (const `int64` &`input_value`)
- OSCL\_IMPORT\_REF void `set_uint64` (`uint64` &`input_value`, const `uint32` `upper`, const `uint32` `lower`)
- OSCL\_IMPORT\_REF `uint32` `get_uint64_upper32` (const `uint64` &`input_value`)
- OSCL\_IMPORT\_REF `uint32` `get_uint64_lower32` (const `uint64` &`input_value`)
- OSCL\_IMPORT\_REF `uint32` `get_uint64_middle32` (const `uint64` &`input_value`)

### 6.51.1 Detailed Description

The Oscl\_Int64\_Utils class provides a wrapper for commonly used int64/uint64 operations.

The Oscl\_Int64\_Utils class:

Provides a wrapper for commonly used operations to mask the differences between OSes that have an int64/uint64 class instead of a 64-bit integer.

## 6.51.2 Member Function Documentation

- 6.51.2.1 **OSCL\_IMPORT\_REF int32 Oscl\_Int64\_Utils::get\_int64\_lower32 (const int64 & *input\_value*) [static]**
- 6.51.2.2 **OSCL\_IMPORT\_REF int32 Oscl\_Int64\_Utils::get\_int64\_middle32 (const int64 & *input\_value*) [static]**
- 6.51.2.3 **OSCL\_IMPORT\_REF int32 Oscl\_Int64\_Utils::get\_int64\_upper32 (const int64 & *input\_value*) [static]**
- 6.51.2.4 **OSCL\_IMPORT\_REF uint32 Oscl\_Int64\_Utils::get\_uint64\_lower32 (const uint64 & *input\_value*) [static]**
- 6.51.2.5 **OSCL\_IMPORT\_REF uint32 Oscl\_Int64\_Utils::get\_uint64\_middle32 (const uint64 & *input\_value*) [static]**
- 6.51.2.6 **OSCL\_IMPORT\_REF uint32 Oscl\_Int64\_Utils::get\_uint64\_upper32 (const uint64 & *input\_value*) [static]**
- 6.51.2.7 **OSCL\_IMPORT\_REF void Oscl\_Int64\_Utils::set\_int64 (int64 & *input\_value*, const int32 *upper*, const int32 *lower*) [static]**
- 6.51.2.8 **OSCL\_IMPORT\_REF void Oscl\_Int64\_Utils::set\_uint64 (uint64 & *input\_value*, const uint32 *upper*, const uint32 *lower*) [static]**

The documentation for this class was generated from the following file:

- [oscl\\_int64\\_utils.h](#)

## 6.52 Oscl\_Less< T > Struct Template Reference

```
#include <oscl_map.h>
```

### Public Methods

- bool [operator\(\)](#) (const T &x, const T &y) const

```
template<class T> struct Oscl_Less< T >
```

#### 6.52.1 Member Function Documentation

**6.52.1.1 template<class T> bool Oscl\_Less< T >::operator() (const T & x, const T & y) const [inline]**

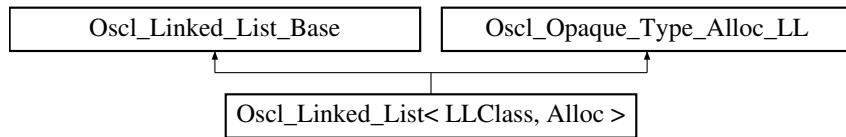
The documentation for this struct was generated from the following file:

- [oscl\\_map.h](#)

## 6.53 Oscl\_Linked\_List< LLClass, Alloc > Class Template Reference

```
#include <oscl_linked_list.h>
```

Inheritance diagram for Oscl\_Linked\_List< LLClass, Alloc >::



### Public Methods

- [Oscl\\_Linked\\_List \(\)](#)
- [~Oscl\\_Linked\\_List \(\)](#)
- int32 [dequeue\\_element \(LLClass &element\)](#)
- int32 [get\\_first \(LLClass &ele\)](#)
- int32 [get\\_next \(LLClass &ele\)](#)
- int32 [check\\_list \(\)](#)
- int32 [get\\_num\\_elements \(\)](#)
- int32 [add\\_element \(LLClass &new\\_element\)](#)
- int32 [add\\_to\\_front \(const LLClass &new\\_element\)](#)
- int32 [get\\_element \(int32 index, LLClass &element\)](#)
- int32 [remove\\_element \(const LLClass &data\\_to\\_remove\)](#)
- int32 [get\\_index \(const LLClass &data\)](#)
- int32 [remove\\_element \(const int32 index\\_to\\_remove\)](#)
- int32 [move\\_to\\_end \(const LLClass &data\\_to\\_move\)](#)
- int32 [move\\_to\\_front \(const LLClass &data\\_to\\_move\)](#)

#### 6.53.1 Detailed Description

`template<class LLClass, class Alloc> class Oscl_Linked_List< LLClass, Alloc >`

Oscl Linked List Class

#### 6.53.2 Constructor & Destructor Documentation

**6.53.2.1 `template<class LLClass, class Alloc> Oscl_Linked_List< LLClass, Alloc >::Oscl_Linked_List () [inline]`**

Initialized the protected variables of list.

**6.53.2.2 `template<class LLClass, class Alloc> Oscl_Linked_List< LLClass, Alloc >::~Oscl_Linked_List () [inline]`**

The destructor.

### 6.53.3 Member Function Documentation

#### 6.53.3.1 template<class LLClass, class Alloc> int32 Oscl\_Linked\_List< LLClass, Alloc >::add\_element (LLClass & new\_element) [inline]

Adds new element to the list.if list is already there then it adds element at end of list otherwise it create the list and add the element as first element of list.

**Parameters:**

*new\_element* the element to be add in the list.

**Returns:**

32-bit integer on the success returns 1.

#### 6.53.3.2 template<class LLClass, class Alloc> int32 Oscl\_Linked\_List< LLClass, Alloc >::add\_to\_front (const LLClass & new\_element) [inline]

Adds new element at the start of the list.if list is already exist then it adds element at start of list otherwise it create the list and add the element as first element of list.

**Parameters:**

*new\_element* the element to be add in the list.

**Returns:**

32-bit integer on the success returns 1.

#### 6.53.3.3 template<class LLClass, class Alloc> int32 Oscl\_Linked\_List< LLClass, Alloc >::check\_list () [inline]

Debug routine: Checks the list for elements.

**Returns:**

32-bit integer, if node count is equal to number of node added to the list then returns 1 otherwise returns 0.

Reimplemented from [Oscl\\_Linked\\_List\\_Base](#).

#### 6.53.3.4 template<class LLClass, class Alloc> int32 Oscl\_Linked\_List< LLClass, Alloc >::dequeue\_element (LLClass & element) [inline]

#### 6.53.3.5 template<class LLClass, class Alloc> int32 Oscl\_Linked\_List< LLClass, Alloc >::get\_element (int32 index, LLClass & element) [inline]

Search and returns the element in the list for passed index.

**Parameters:**

*index, element* The index is the count for the node.

**Returns:**

32-bit integer on success returns 1 otherwise returns 0.

**6.53.3.6 template<class LLClass, class Alloc> int32 Oscl\_Linked\_List< LLClass, Alloc >::get\_first (LLClass & *ele*) [inline]**

Return the first element of list in passed parameter,

**Parameters:**

*ele* return the value of first element of list in this parameter

**Returns:**

32-bit interger,If first element found, it returns 1 otherwise it returns 0

**6.53.3.7 template<class LLClass, class Alloc> int32 Oscl\_Linked\_List< LLClass, Alloc >::get\_index (const LLClass & *data*) [inline]**

Returns the index for requested element.

**Parameters:**

*data* the element for which index to be return.

**Returns:**

32-bit integer if data is found in the list it returns index otherwise it returns -1.

**6.53.3.8 template<class LLClass, class Alloc> int32 Oscl\_Linked\_List< LLClass, Alloc >::get\_next (LLClass & *ele*) [inline]**

Return the next element of list in passed parameter,

**Parameters:**

*ele* return the value of next element of list in this parameter

**Returns:**

32-bit interger ,if next element is found in list,it returns 1 otherwise it returns 0

**6.53.3.9 template<class LLClass, class Alloc> int32 Oscl\_Linked\_List< LLClass, Alloc >::get\_num\_elements () [inline]**

Get number of elements in the list.

**Returns:**

32-bit integer, number of elements in list.

**6.53.3.10 template<class LLClass, class Alloc> int32 Oscl\_Linked\_List< LLClass, Alloc >::move\_to\_end (const LLClass & *data\_to\_move*) [inline]**

Moves the element to end of the list

**Parameters:**

*data\_to\_move*

**Returns:**

On success returns 1 otherwise returns 0.

**6.53.3.11 template<class LLClass, class Alloc> int32 Oscl\_Linked\_List< LLClass, Alloc >::move\_to\_front (const LLClass & *data\_to\_move*) [inline]**

Moves the element to front of the list

**Parameters:**

*data\_to\_move*

**Returns:**

On success returns 1 otherwise returns 0.

**6.53.3.12 template<class LLClass, class Alloc> int32 Oscl\_Linked\_List< LLClass, Alloc >::remove\_element (const int32 *index\_to\_remove*) [inline]**

Removes the element for requested index.

**Parameters:**

*index\_to\_remove*

**Returns:**

on success return 1 otherwise return 0.

Reimplemented from [Oscl\\_Linked\\_List\\_Base](#).

**6.53.3.13 template<class LLClass, class Alloc> int32 Oscl\_Linked\_List< LLClass, Alloc >::remove\_element (const LLClass & *data\_to\_remove*) [inline]**

Removes the element from the list.

**Parameters:**

*data\_to\_remove*

**Returns:**

32-bit integer on if element fount in the list returns 1 otherwise returns 0.

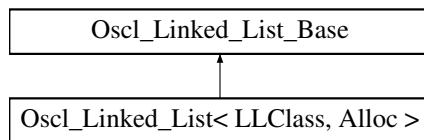
The documentation for this class was generated from the following file:

- [oscl\\_linked\\_list.h](#)

## 6.54 Oscl\_Linked\_List\_Base Class Reference

```
#include <oscl_linked_list.h>
```

Inheritance diagram for Oscl\_Linked\_List\_Base::



### Protected Methods

- virtual ~Oscl\_Linked\_List\_Base ()
- OSCL\_IMPORT\_REF void construct (Oscl\_Opaque\_Type\_Alloc\_LL \*op)
- OSCL\_IMPORT\_REF void destroy ()
- OSCL\_IMPORT\_REF int32 get\_first (OsclAny \*ele)
- OSCL\_IMPORT\_REF int32 get\_next (OsclAny \*ele)
- OSCL\_IMPORT\_REF int32 check\_list ()
- OSCL\_IMPORT\_REF int32 add\_element (OsclAny \*new\_element)
- OSCL\_IMPORT\_REF int32 add\_to\_front (const OsclAny \*new\_element)
- OSCL\_IMPORT\_REF int32 get\_element (int32 index, OsclAny \*element)
- OSCL\_IMPORT\_REF int32 remove\_element (const OsclAny \*data\_to\_remove)
- OSCL\_IMPORT\_REF int32 get\_index (const OsclAny \*data)
- OSCL\_IMPORT\_REF int32 remove\_element (const int32 index\_to\_remove)
- OSCL\_IMPORT\_REF int32 move\_to\_end (const OsclAny \*data\_to\_move)
- OSCL\_IMPORT\_REF int32 move\_to\_front (const OsclAny \*data\_to\_move)

### Protected Attributes

- OsclAny \* head
- OsclAny \* tail
- OsclAny \* iterator
- int32 num\_elements
- uint32 sizeof\_T

#### 6.54.1 Detailed Description

Oscl Linked List Base Class. A non-templated base class is used to avoid large inline functions in the [Oscl\\_Linked\\_List](#) implementation.

## 6.54.2 Constructor & Destructor Documentation

**6.54.2.1** `virtual Oscl_Linked_List_Base::~Oscl_Linked_List_Base ()` [inline, protected, virtual]

## 6.54.3 Member Function Documentation

**6.54.3.1** `OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::add_element (OsclAny * new_element)` [protected]

Adds new element to the list.if list is already there then it adds element at end of list otherwise it create the list and add the element as first element of list.

**Parameters:**

*new\_element* the element to be add in the list.

**Returns:**

32-bit integer on the success returns 1.

**6.54.3.2** `OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::add_to_front (const OsclAny * new_element)` [protected]

Adds new element at the start of the list.if list is already exist then it adds element at start of list otherwise it create the list and add the element as first element of list.

**Parameters:**

*new\_element* the element to be add in the list.

**Returns:**

32-bit integer on the success returns 1.

**6.54.3.3** `OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::check_list ()` [protected]

Debug routine: Checks the list for elements.

**Returns:**

32-bit integer, if node count is equal to number of node added to the list then returns 1 otherwise returns 0.

Reimplemented in [Oscl\\_Linked\\_List< LLClass, Alloc >](#).

**6.54.3.4** `OSCL_IMPORT_REF void Oscl_Linked_List_Base::construct (Oscl_Opaque_Type_Alloc_LL * op)` [protected]

**6.54.3.5** `OSCL_IMPORT_REF void Oscl_Linked_List_Base::destroy ()` [protected]

**6.54.3.6** `OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::get_element (int32 index, OsclAny * element)` [protected]

Search and returns the element in the list for passed index.

**Parameters:**

*index, element* The index is the count for the node.

**Returns:**

32-bit integer on success returns 1 otherwise returns 0.

**6.54.3.7 OSCL\_IMPORT\_REF int32 Oscl\_Linked\_List\_Base::get\_first (OsclAny \* *ele*)  
[protected]**

Return the first element of list in passed parameter,

**Parameters:**

*ele* return the value of first element of list in this parameter

**Returns:**

32-bit interger,If first element found, it returns 1 otherwise it returns 0

**6.54.3.8 OSCL\_IMPORT\_REF int32 Oscl\_Linked\_List\_Base::get\_index (const OsclAny \* *data*)  
[protected]**

Returns the index for requested element.

**Parameters:**

*data* the element for which index to be return.

**Returns:**

32-bit integer if data is found in the list it returns index otherwise it returns -1.

**6.54.3.9 OSCL\_IMPORT\_REF int32 Oscl\_Linked\_List\_Base::get\_next (OsclAny \* *ele*)  
[protected]**

Return the next element of list in passed parameter,

**Parameters:**

*ele* return the value of next element of list in this parameter

**Returns:**

32-bit interger ,if next element is found in list,it returns 1 otherwise it returns 0

**6.54.3.10 OSCL\_IMPORT\_REF int32 Oscl\_Linked\_List\_Base::move\_to\_end (const OsclAny \*  
*data\_to\_move*) [protected]**

Moves the element to end of the list

**Parameters:**

*data\_to\_move*

**Returns:**

On success returns 1 otherwise returns 0.

**6.54.3.11 OSCL\_IMPORT\_REF int32 Oscl\_Linked\_List\_Base::move\_to\_front (const OsclAny \*  
*data\_to\_move*) [protected]**

Moves the element to front of the list

**Parameters:**

*data\_to\_move*

**Returns:**

On success returns 1 otherwise returns 0.

**6.54.3.12 OSCL\_IMPORT\_REF int32 Oscl\_Linked\_List\_Base::remove\_element (const int32  
*index\_to\_remove*) [protected]**

Removes the element for requested index.

**Parameters:**

*index\_to\_remove*

**Returns:**

on success return 1 otherwise return 0.

Reimplemented in [Oscl\\_Linked\\_List< LLClass, Alloc >](#).

**6.54.3.13 OSCL\_IMPORT\_REF int32 Oscl\_Linked\_List\_Base::remove\_element (const OsclAny \*  
*data\_to\_remove*) [protected]**

Removes the element from the list.

**Parameters:**

*data\_to\_remove*

**Returns:**

32-bit integer on if element fount in the list returns 1 otherwise returns 0.

## 6.54.4 Field Documentation

**6.54.4.1 OsclAny\* Oscl\_Linked\_List\_Base::head [protected]****6.54.4.2 OsclAny\* Oscl\_Linked\_List\_Base::iterator [protected]****6.54.4.3 int32 Oscl\_Linked\_List\_Base::num\_elements [protected]****6.54.4.4 uint32 Oscl\_Linked\_List\_Base::sizeof\_T [protected]****6.54.4.5 OsclAny\* Oscl\_Linked\_List\_Base::tail [protected]**

The documentation for this class was generated from the following file:

- [oscl\\_linked\\_list.h](#)

## 6.55 Oscl\_Map< Key, T, Alloc, Compare > Class Template Reference

```
#include <oscl_map.h>
```

### Public Types

- `typedef Key key_type`
- `typedef Compare key_compare`
- `typedef Oscl_Pair< const Key, T > value_type`
- `typedef Oscl_Map< Key, T, Alloc, Compare > self`
- `typedef rep_type::pointer pointer`
- `typedef rep_type::reference reference`
- `typedef rep_type::const_reference const_reference`
- `typedef rep_type::iterator iterator`
- `typedef rep_type::const_iterator const_iterator`
- `typedef rep_type::size_type size_type`
- `typedef Oscl_Pair< iterator, bool > pair_iterator_bool`
- `typedef Oscl_Pair< iterator, iterator > pair_iterator_iterator`
- `typedef Oscl_Pair< const_iterator, const_iterator > pair_citerator_citerator`

### Public Methods

- `Oscl_Map (const Compare &comp=Compare())`
- `Oscl_Map (const self &x)`
- `self & operator= (const self &x)`
- `key_compare key_comp () const`
- `value_compare value_comp () const`
- `iterator begin ()`
- `const_iterator begin () const`
- `iterator end ()`
- `const_iterator end () const`
- `bool empty () const`
- `size_type size () const`
- `size_type max_size () const`
- `T & operator[ ] (const key_type &k)`
- `pair_iterator_bool insert (const value_type &x)`
- `iterator insert (iterator position, const value_type &x)`
- `void insert (const value_type *first, const value_type *last)`
- `void erase (iterator position)`
- `size_type erase (const key_type &x)`
- `void erase (iterator first, iterator last)`
- `void clear ()`
- `iterator find (const key_type &x)`
- `const_iterator find (const key_type &x) const`
- `size_type count (const key_type &x) const`
- `iterator lower_bound (const key_type &x)`
- `const_iterator lower_bound (const key_type &x) const`
- `iterator upper_bound (const key_type &x)`

- `const_iterator upper_bound (const key_type &x) const`
- `pair_iterator iterator equal_range (const key_type &x)`
- `pair_citerator citerator equal_range (const key_type &x) const`

### 6.55.1 Detailed Description

**template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> class Oscl\_Map< Key, T, Alloc, Compare >**

Oscl\_Map Class. A subset of STL::Map methods. Oscl\_Map is a sorted associative container that associates objects of type Key with objects of type T. It is also a unique associative container, meaning that no two elements have the same key. Oscl\_Map uses the key to speed lookup, insertion, and deletion of elements. It is often superior to all other containers when you need to lookup an element by key value. Due to the underlying tree structure, inserts and erases can be performed in logarithmic time, where a vector would take linear time in some cases.

## 6.55.2 Member Typedef Documentation

- 6.55.2.1 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> typedef rep\_type::const\_iterator Oscl\_Map< Key, T, Alloc, Compare >::const\_iterator
- 6.55.2.2 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> typedef rep\_type::const\_reference Oscl\_Map< Key, T, Alloc, Compare >::const\_reference
- 6.55.2.3 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> typedef rep\_type::iterator Oscl\_Map< Key, T, Alloc, Compare >::iterator
- 6.55.2.4 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> typedef Compare Oscl\_Map< Key, T, Alloc, Compare >::key\_compare
- 6.55.2.5 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> typedef Key Oscl\_Map< Key, T, Alloc, Compare >::key\_type
- 6.55.2.6 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> typedef Oscl\_Pair<const\_iterator, const\_iterator> Oscl\_Map< Key, T, Alloc, Compare >::pair\_citerator\_citerator
- 6.55.2.7 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> typedef Oscl\_Pair<iterator, bool> Oscl\_Map< Key, T, Alloc, Compare >::pair\_iterator\_bool
- 6.55.2.8 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> typedef Oscl\_Pair<iterator, iterator> Oscl\_Map< Key, T, Alloc, Compare >::pair\_iterator\_iterator
- 6.55.2.9 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> typedef rep\_type::pointer Oscl\_Map< Key, T, Alloc, Compare >::pointer
- 6.55.2.10 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> typedef rep\_type::reference Oscl\_Map< Key, T, Alloc, Compare >::reference
- 6.55.2.11 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> typedef Oscl\_Map<Key, T, Alloc, Compare> Oscl\_Map< Key, T, Alloc, Compare >::self
- 6.55.2.12 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> typedef rep\_type::size\_type Oscl\_Map< Key, T, Alloc, Compare >::size\_type
- 6.55.2.13 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> typedef Oscl\_Pair<const Key, T> Oscl\_Map< Key, T, Alloc, Compare >::value\_type

## 6.55.3 Constructor & Destructor Documentation

- 6.55.3.1 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> Oscl\_Map< Key, T, Alloc, Compare >::Oscl\_Map (const Compare & comp = Compare() ) [inline]

Creates an empty map using comp as the key compare object

**6.55.3.2 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> Oscl\_Map< Key, T, Alloc, Compare >::Oscl\_Map (const **self** & *x*) [inline]**

Oscl\_Map copy constructor

#### 6.55.4 Member Function Documentation

**6.55.4.1 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> const\_iterator Oscl\_Map< Key, T, Alloc, Compare >::begin () const [inline]**

Returns a const iterator pointing to the beginning of the map

**6.55.4.2 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> iterator Oscl\_Map< Key, T, Alloc, Compare >::begin () [inline]**

Returns an iterator pointing to the beginning of the map

**6.55.4.3 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> void Oscl\_Map< Key, T, Alloc, Compare >::clear () [inline]**

Erases all elements

**6.55.4.4 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> size\_type Oscl\_Map< Key, T, Alloc, Compare >::count (const **key\_type** & *x*) const [inline]**

Returns the number of elements with key *x*. For map this will either be 0 or 1.

**6.55.4.5 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> bool Oscl\_Map< Key, T, Alloc, Compare >::empty () const [inline]**

Returns true if map size is 0

**6.55.4.6 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> const\_iterator Oscl\_Map< Key, T, Alloc, Compare >::end () const [inline]**

Returns a const iterator pointing to the end of the map.

**6.55.4.7 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> iterator Oscl\_Map< Key, T, Alloc, Compare >::end () [inline]**

Returns an iterator pointing to the end of the map.

**6.55.4.8 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> pair\_citerator\_citerator Oscl\_Map< Key, T, Alloc, Compare >::equal\_range (const **key\_type** & *x*) const [inline]**

Finds a range containing all elements whose key is *x*

**6.55.4.9 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>>  
 pair\_iterator\_iterator Oscl\_Map< Key, T, Alloc, Compare >::equal\_range (const  
 key\_type & x) [inline]**

Finds a range containing all elements whose key is x

**6.55.4.10 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> void  
 Oscl\_Map< Key, T, Alloc, Compare >::erase (iterator first, iterator last) [inline]**

Erases all elements in the range [first,last)

**6.55.4.11 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> size\_type  
 Oscl\_Map< Key, T, Alloc, Compare >::erase (const key\_type & x) [inline]**

Erases the element with key x

**6.55.4.12 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> void  
 Oscl\_Map< Key, T, Alloc, Compare >::erase (iterator position) [inline]**

Erases the element pointed to by position

**6.55.4.13 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>>  
 const\_iterator Oscl\_Map< Key, T, Alloc, Compare >::find (const key\_type & x) const  
 [inline]**

Finds an element whose key is x

**6.55.4.14 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> iterator  
 Oscl\_Map< Key, T, Alloc, Compare >::find (const key\_type & x) [inline]**

Finds an element whose key is x

**6.55.4.15 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> void  
 Oscl\_Map< Key, T, Alloc, Compare >::insert (const value\_type \*first, const value\_type  
 \*last) [inline]**

Inserts the range [first,last) into the map

**6.55.4.16 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> iterator  
 Oscl\_Map< Key, T, Alloc, Compare >::insert (iterator position, const value\_type & x)  
 [inline]**

Inserts x into the map using position as a hint as to where it should be inserted

**6.55.4.17 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>>  
`pair_iterator_bool` Oscl\_Map< Key, T, Alloc, Compare >::insert (const `value_type` & x) [inline]**

Inserts x into the map

**6.55.4.18 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>>  
`key_compare` Oscl\_Map< Key, T, Alloc, Compare >::key\_comp () const [inline]**

Returns the key compare object used by the map

**6.55.4.19 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>>  
`const_iterator` Oscl\_Map< Key, T, Alloc, Compare >::lower\_bound (const `key_type` & x) const [inline]**

Finds the first element whose key is not less than x

**6.55.4.20 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> iterator  
`Oscl_Map`< Key, T, Alloc, Compare >::lower\_bound (const `key_type` & x) [inline]**

Finds the first element whose key is not less than x

**6.55.4.21 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> size\_type  
`Oscl_Map`< Key, T, Alloc, Compare >::max\_size () const [inline]**

Returns the maximum possible size of the map

**6.55.4.22 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> self&  
`Oscl_Map`< Key, T, Alloc, Compare >::operator= (const `self` & x) [inline]**

Oscl\_Map assignment operator

**6.55.4.23 ]**

template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> T& Oscl\_Map< Key, T, Alloc, Compare >::operator[] (const `key_type` & k) [inline]

Returns a reference to the object that is associated with a particular key. If the map does not already contain such an object, operator[] inserts the default object `value_type()`.

**6.55.4.24 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> size\_type  
`Oscl_Map`< Key, T, Alloc, Compare >::size () const [inline]**

Returns the size of the map

**6.55.4.25 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>>  
const\_iterator Oscl\_Map< Key, T, Alloc, Compare >::upper\_bound (const key\_type &  
x) const [inline]**

Finds the first element whose key is not greater than x

**6.55.4.26 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> iterator  
Oscl\_Map< Key, T, Alloc, Compare >::upper\_bound (const key\_type & x) [inline]**

Finds the first element whose key is not greater than x

**6.55.4.27 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>>  
value\_compare Oscl\_Map< Key, T, Alloc, Compare >::value\_comp () const  
[inline]**

Returns the value compare object used by the map

The documentation for this class was generated from the following file:

- [oscl\\_map.h](#)

## 6.56 Oscl\_Map< Key, T, Alloc, Compare >::value\_compare Class Reference

```
#include <oscl_map.h>
```

### Public Methods

- bool [operator\(\)](#) (const [value\\_type](#) &x, const [value\\_type](#) &y) const

### Protected Methods

- [value\\_compare](#) (Compare c)

### Protected Attributes

- Compare [comp](#)

### Friends

- class [Oscl\\_Map< Key, T, Alloc, Compare >](#)

```
template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> class Oscl_Map< Key,  
T, Alloc, Compare >::value_compare
```

#### 6.56.1 Constructor & Destructor Documentation

```
6.56.1.1 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> Oscl\_Map< Key, T, Alloc, Compare >::value\_compare::value\_compare (Compare c) [inline, protected]
```

#### 6.56.2 Member Function Documentation

```
6.56.2.1 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> bool  
Oscl\_Map< Key, T, Alloc, Compare >::value\_compare::operator\(\) (const value\_type &  
x, const value\_type &y) const [inline]
```

#### 6.56.3 Friends And Related Function Documentation

```
6.56.3.1 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> friend class  
Oscl\_Map< Key, T, Alloc, Compare > [friend]
```

#### 6.56.4 Field Documentation

```
6.56.4.1 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> Compare  
Oscl\_Map< Key, T, Alloc, Compare >::value\_compare::comp [protected]
```

The documentation for this class was generated from the following file:

- [oscl\\_map.h](#)

## 6.57 Oscl\_MTLinked\_List< LLClass, Alloc, TheLock > Class Template Reference

```
#include <oscl_linked_list.h>
```

### Public Methods

- [Oscl\\_MTLinked\\_List \(\)](#)
- [~Oscl\\_MTLinked\\_List \(\)](#)
- int32 [dequeue\\_element \(LLClass &element\)](#)
- int32 [add\\_element \(LLClass &new\\_element\)](#)
- int32 [add\\_to\\_front \(LLClass &new\\_element\)](#)
- uint32 [get\\_element \(int32 index, LLClass &element\)](#)
- int32 [remove\\_element \(const LLClass &data\\_to\\_remove\)](#)
- int32 [get\\_index \(const LLClass &data\)](#)
- int32 [remove\\_element \(const int32 index\\_to\\_remove\)](#)
- int32 [move\\_to\\_end \(const LLClass &data\\_to\\_move\)](#)
- int32 [move\\_to\\_front \(const LLClass &data\\_to\\_move\)](#)

### Protected Attributes

- [Oscl\\_Linked\\_List< LLClass, Alloc > the\\_list](#)

#### 6.57.1 Detailed Description

**template<class LLClass, class Alloc, class TheLock> class Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >**

Oscl\_MTLinked\_List is a multi-threaded version of the LinkedList. It has mutex protection to allow access by different threads.

#### 6.57.2 Constructor & Destructor Documentation

**6.57.2.1 template<class LLClass, class Alloc, class TheLock> Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::Oscl\_MTLinked\_List () [inline]**

Constructor for Oscl\_MTLinked\_List

**6.57.2.2 template<class LLClass, class Alloc, class TheLock> Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::~Oscl\_MTLinked\_List () [inline]**

Destructor for Oscl\_MTLinked\_List

### 6.57.3 Member Function Documentation

#### 6.57.3.1 template<class LLClass, class Alloc, class TheLock> int32 Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::add\_element (LLClass & *new\_element*) [inline]

Adds new element to the Multi Threaded Linked list.if list is already there then it adds element at end of list otherwise it create the list and add the element as first element of list.

**Parameters:**

*new\_element* the element to be add in the list.

**Returns:**

32-bit integer on the success returns 1.

#### 6.57.3.2 template<class LLClass, class Alloc, class TheLock> int32 Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::add\_to\_front (LLClass & *new\_element*) [inline]

Adds new element at the start of the Multi Threaded Linked list. if list is already exist then it adds element at start of list otherwise it create the list and add the element as first element of list.

**Parameters:**

*new\_element* the element to be add in the list.

**Returns:**

32-bit integer on the success returns 1.

#### 6.57.3.3 template<class LLClass, class Alloc, class TheLock> int32 Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::dequeue\_element (LLClass & *element*) [inline]

#### 6.57.3.4 template<class LLClass, class Alloc, class TheLock> uint32 Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::get\_element (int32 *index*, LLClass & *element*) [inline]

Search and returns the element in the Multi Threaded Linked List for passed index.

**Parameters:**

*index, element* The index is the count for the node.

**Returns:**

32-bit integer on success returns 1 otherwise returns 0.

#### 6.57.3.5 template<class LLClass, class Alloc, class TheLock> int32 Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::get\_index (const LLClass & *data*) [inline]

Returns the index for requested element.

**Parameters:**

*data* the element for which index to be return.

**Returns:**

32-bit integer if data is found in the list it returns index otherwise it returns -1.

**6.57.3.6 template<class LLClass, class Alloc, class TheLock> int32 Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::move\_to\_end (const LLClass & *data\_to\_move*) [inline]**

Moves the element to end of the list

**Parameters:**

*data\_to\_move*

**Returns:**

On success returns 1 otherwise returns 0.

**6.57.3.7 template<class LLClass, class Alloc, class TheLock> int32 Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::move\_to\_front (const LLClass & *data\_to\_move*) [inline]**

Moves the element to front of the list

**Parameters:**

*data\_to\_move*

**Returns:**

On success returns 1 otherwise returns 0.

**6.57.3.8 template<class LLClass, class Alloc, class TheLock> int32 Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::remove\_element (const int32 *index\_to\_remove*) [inline]**

Removes the element for requested index.

**Parameters:**

*index\_to\_remove*

**Returns:**

on success return 1 otherwise return 0.

**6.57.3.9 template<class LLClass, class Alloc, class TheLock> int32 Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::remove\_element (const LLClass & *data\_to\_remove*) [inline]**

Removes the element from the list.

**Parameters:**

*data\_to\_remove*

**Returns:**

32-bit integer on if element fount in the list returns 1 otherwise returns 0.

---

**6.57.4 Field Documentation****6.57.4.1 template<class LLClass, class Alloc, class TheLock> [Oscl\\_Linked\\_List<LLClass, Alloc>](#) Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::the\_list [protected]**

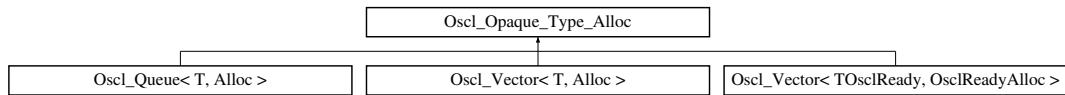
The documentation for this class was generated from the following file:

- [oscl\\_linked\\_list.h](#)

## 6.58 Oscl\_Opaque\_Type\_Alloc Class Reference

```
#include <oscl_opaque_type.h>
```

Inheritance diagram for Oscl\_Opaque\_Type\_Alloc::



### Public Methods

- virtual void **construct** (**OsclAny** \*p, const **OsclAny** \*init\_val)=0
- virtual void **destroy** (**OsclAny** \*p)=0
- virtual **OsclAny** \* **allocate** (const uint32 size)=0
- virtual void **deallocate** (**OsclAny** \*p)=0

#### 6.58.1 Detailed Description

This class combines opaque type operations with memory allocation operations.

#### 6.58.2 Member Function Documentation

**6.58.2.1 virtual **OsclAny**\* Oscl\_Opaque\_Type\_Alloc::allocate (const uint32 size) [pure virtual]**

Allocate "size" bytes

**6.58.2.2 virtual void Oscl\_Opaque\_Type\_Alloc::construct (**OsclAny** \* p, const **OsclAny** \* init\_val) [pure virtual]**

Construct element at p using element at init\_val as the initial value. Both pointers must be non-NULL.

**6.58.2.3 virtual void Oscl\_Opaque\_Type\_Alloc::deallocate (**OsclAny** \* p) [pure virtual]**

Deallocate memory previously allocated with "allocate"

**6.58.2.4 virtual void Oscl\_Opaque\_Type\_Alloc::destroy (**OsclAny** \* p) [pure virtual]**

Destroy element at p.

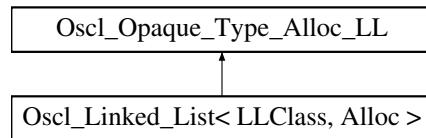
The documentation for this class was generated from the following file:

- [oscl\\_opaque\\_type.h](#)

## 6.59 Oscl\_Opaque\_Type\_Alloc\_LL Class Reference

```
#include <oscl_opaque_type.h>
```

Inheritance diagram for Oscl\_Opaque\_Type\_Alloc\_LL::



### Public Methods

- virtual void `construct (OsclAny *p, const OsclAny *init_val)=0`
- virtual void `destroy (OsclAny *p)=0`
- virtual `OsclAny * allocate (const uint32 size)=0`
- virtual void `deallocate (OsclAny *p)=0`
- virtual `OsclAny * get_next (const OsclAny *elem) const=0`
- virtual void `set_next (OsclAny *elem, const OsclAny *nextelem)=0`
- virtual void `get_data (OsclAny *elem, OsclAny *data_val)=0`
- virtual bool `compare_data (const OsclAny *elem, const OsclAny *data_val) const=0`

### 6.59.1 Detailed Description

This class combines opaque type operations with memory allocation operations and linked list support

### 6.59.2 Member Function Documentation

**6.59.2.1 virtual `OsclAny* Oscl_Opaque_Type_Alloc_LL::allocate (const uint32 size)` [pure virtual]**

Allocate "size" bytes

**6.59.2.2 virtual `bool Oscl_Opaque_Type_Alloc_LL::compare_data (const OsclAny * elem, const OsclAny * data_val) const` [pure virtual]**

Compare data.

**6.59.2.3 virtual `void Oscl_Opaque_Type_Alloc_LL::construct (OsclAny * p, const OsclAny * init_val)` [pure virtual]**

Construct element at p using element at init\_val as the initial value. Both pointers must be non-NULL.

**6.59.2.4 virtual `void Oscl_Opaque_Type_Alloc_LL::deallocate (OsclAny * p)` [pure virtual]**

Deallocate memory previously allocated with "allocate"

**6.59.2.5 virtual void Oscl\_Opaque\_Type\_Alloc\_LL::destroy (OsclAny \**p*) [pure virtual]**

Destroy element at p.

**6.59.2.6 virtual void Oscl\_Opaque\_Type\_Alloc\_LL::get\_data (OsclAny \**elem*, OsclAny \**data\_val*) [pure virtual]**

Get data

**6.59.2.7 virtual OsclAny\* Oscl\_Opaque\_Type\_Alloc\_LL::get\_next (const OsclAny \**elem*) const [pure virtual]**

Get next element in linked list.

**6.59.2.8 virtual void Oscl\_Opaque\_Type\_Alloc\_LL::set\_next (OsclAny \**elem*, const OsclAny \**nextelem*) [pure virtual]**

Set next element in linked list.

The documentation for this class was generated from the following file:

- [oscl\\_opaque\\_type.h](#)

## 6.60 Oscl\_Opaque\_Type\_Compare Class Reference

```
#include <oscl_opaque_type.h>
```

Inheritance diagram for Oscl\_Opaque\_Type\_Compare::



### Public Methods

- virtual void `swap (OsclAny *a, const OsclAny *b)=0`
- virtual int `compare_LT (OsclAny *a, OsclAny *b) const=0`
- virtual int `compare_EQ (const OsclAny *a, const OsclAny *b) const=0`

#### 6.60.1 Detailed Description

Opaque type operations with swap & comparisons.

#### 6.60.2 Member Function Documentation

##### 6.60.2.1 virtual int Oscl\_Opaque\_Type\_Compare::compare\_EQ (const OsclAny \* a, const OsclAny \* b) const [pure virtual]

Return a==b.

Implemented in `OsclPriorityQueue< Qelem, Alloc, Container, Compare >`, `OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclReadyCompare >`, and `OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >`.

##### 6.60.2.2 virtual int Oscl\_Opaque\_Type\_Compare::compare\_LT (OsclAny \* a, OsclAny \* b) const [pure virtual]

Return a<b.

Implemented in `OsclPriorityQueue< Qelem, Alloc, Container, Compare >`, `OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclReadyCompare >`, and `OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >`.

##### 6.60.2.3 virtual void Oscl\_Opaque\_Type\_Compare::swap (OsclAny \* a, const OsclAny \* b) [pure virtual]

Swap element at "a" with element at "b". Both pointers must be non-NULL.

Implemented in `OsclPriorityQueue< Qelem, Alloc, Container, Compare >`, `OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclReadyCompare >`, and `OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >`.

The documentation for this class was generated from the following file:

- [oscl\\_opaque\\_type.h](#)

## 6.61 Oscl\_Pair< T1, T2 > Struct Template Reference

```
#include <oscl_tree.h>
```

### Public Methods

- [Oscl\\_Pair \(\)](#)
- [Oscl\\_Pair \(const T1 &a, const T2 &b\)](#)

### Data Fields

- [T1 first](#)
- [T2 second](#)

```
template<class T1, class T2> struct Oscl_Pair< T1, T2 >
```

#### 6.61.1 Constructor & Destructor Documentation

**6.61.1.1 template<class T1, class T2> Oscl\_Pair< T1, T2 >::Oscl\_Pair () [inline]**

**6.61.1.2 template<class T1, class T2> Oscl\_Pair< T1, T2 >::Oscl\_Pair (const T1 & a, const T2 & b) [inline]**

#### 6.61.2 Field Documentation

**6.61.2.1 template<class T1, class T2> T1 Oscl\_Pair< T1, T2 >::first**

**6.61.2.2 template<class T1, class T2> T2 Oscl\_Pair< T1, T2 >::second**

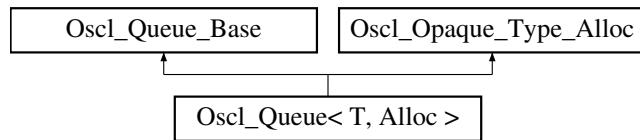
The documentation for this struct was generated from the following file:

- [oscl\\_tree.h](#)

## 6.62 Oscl\_Queue< T, Alloc > Class Template Reference

```
#include <oscl_queue.h>
```

Inheritance diagram for Oscl\_Queue< T, Alloc >::



### Public Types

- typedef T [value\\_type](#)
- typedef T \* [pointer](#)
- typedef T & [reference](#)
- typedef const T & [const\\_reference](#)
- typedef uint32 [size\\_type](#)

### Public Methods

- [Oscl\\_Queue \(\)](#)
- [Oscl\\_Queue \(uint32 n\)](#)
- virtual [~Oscl\\_Queue \(\)](#)
- void [push \(const T &x\)](#)
- [reference front \(\)](#)
- [const\\_reference front \(\) const](#)
- void [pop \(\)](#)
- [reference back \(\)](#)
- [const\\_reference back \(\) const](#)
- void [clear \(\)](#)

#### 6.62.1 Detailed Description

**template<class T, class Alloc> class Oscl\_Queue< T, Alloc >**

Oscl\_Queue Class. A subset of STL::Queue methods. Oscl\_Queue supports constant time insertion (at the end) and removal of elements at the front of the queue. It does not support insertion or removal of elements at the other ends or middle of the queue, nor random access to elements. \* No iteration capability is [currently] supplied. \* No assignment or copy capability is [currently] supplied. The number of elements in a queue can vary dynamically, and memory management is performed automatically.

## 6.62.2 Member Typedef Documentation

- 6.62.2.1 `template<class T, class Alloc> typedef const T& Oscl_Queue< T, Alloc >::const_reference`
- 6.62.2.2 `template<class T, class Alloc> typedef T* Oscl_Queue< T, Alloc >::pointer`
- 6.62.2.3 `template<class T, class Alloc> typedef T& Oscl_Queue< T, Alloc >::reference`
- 6.62.2.4 `template<class T, class Alloc> typedef uint32 Oscl_Queue< T, Alloc >::size_type`
- 6.62.2.5 `template<class T, class Alloc> typedef T Oscl_Queue< T, Alloc >::value_type`

## 6.62.3 Constructor & Destructor Documentation

- 6.62.3.1 `template<class T, class Alloc> Oscl_Queue< T, Alloc >::Oscl_Queue () [inline]`

Creates an empty queue.

- 6.62.3.2 `template<class T, class Alloc> Oscl_Queue< T, Alloc >::Oscl_Queue (uint32 n) [inline]`

Creates an empty queue with capacity n.

### Parameters:

*n* creates a queue with n elements. The main reason for specifying n is efficiency. If you know the capacity to which your queue must grow, then it is more efficient to allocate the queue all at once rather than rely on the automatic reallocation scheme.

- 6.62.3.3 `template<class T, class Alloc> virtual Oscl_Queue< T, Alloc >::~Oscl_Queue () [inline, virtual]`

The destructor.

## 6.62.4 Member Function Documentation

- 6.62.4.1 `template<class T, class Alloc> const_reference Oscl_Queue< T, Alloc >::back () const [inline]`

Returns the last element (const)

- 6.62.4.2 `template<class T, class Alloc> reference Oscl_Queue< T, Alloc >::back () [inline]`

Returns the last element: "back" (generally not too useful, but some debugging aids might want to find out what was just added)

- 6.62.4.3 `template<class T, class Alloc> void Oscl_Queue< T, Alloc >::clear () [inline]`

Removes all elements.

Reimplemented from [Oscl\\_Queue\\_Base](#).

**6.62.4.4 template<class T, class Alloc> const\_reference Oscl\_Queue< T, Alloc >::front () const [inline]**

Returns the first element (const)

**6.62.4.5 template<class T, class Alloc> reference Oscl\_Queue< T, Alloc >::front () [inline]**

Returns the first element.

Reimplemented from [Oscl\\_Queue\\_Base](#).

**6.62.4.6 template<class T, class Alloc> void Oscl\_Queue< T, Alloc >::pop () [inline]**

Removes the first element

Reimplemented from [Oscl\\_Queue\\_Base](#).

**6.62.4.7 template<class T, class Alloc> void Oscl\_Queue< T, Alloc >::push (const T & x) [inline]**

Inserts a new element at the end. Queue will be grown if necessary. If allocation fails, an OSCL\_LEAVE will occur

**Parameters:**

*x* new element

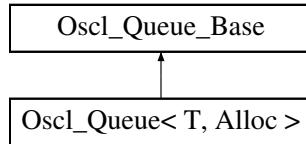
The documentation for this class was generated from the following file:

- [oscl\\_queue.h](#)

## 6.63 Oscl\_Queue\_Base Class Reference

```
#include <oscl_queue.h>
```

Inheritance diagram for Oscl\_Queue\_Base::



### Public Methods

- uint32 `size () const`
- uint32 `capacity () const`
- bool `empty () const`
- OSCL\_IMPORT\_REF void `reserve (uint32 n)`

### Protected Methods

- OSCL\_IMPORT\_REF void `construct (Oscl_Opaque_Type_Alloc *aType)`
- OSCL\_IMPORT\_REF void `construct (Oscl_Opaque_Type_Alloc *aType, uint32 n)`
- virtual `~Oscl_Queue_Base ()`
- OSCL\_IMPORT\_REF void `destroy ()`
- OSCL\_IMPORT\_REF void `push (const OsclAny *x)`
- OSCL\_IMPORT\_REF void `pop ()`
- OSCL\_IMPORT\_REF void `clear ()`

### Protected Attributes

- uint32 `numelems`
- uint32 `bufsize`
- `OsclAny * elems`
- uint32 `sizeof_T`
- uint32 `ifront`
- uint32 `irear`

#### 6.63.1 Detailed Description

`Oscl_Queue_Base` is a non-templatized base class for [Oscl\\_Queue](#). The purpose of this base class is to avoid large inline routines in the [Oscl\\_Queue](#) implementation. This class is not intended for direct instantiation except by [Oscl\\_Queue](#).

#### 6.63.2 Constructor & Destructor Documentation

**6.63.2.1 virtual Oscl\_Queue\_Base::~Oscl\_Queue\_Base () [inline, protected, virtual]**

The destructor.

### 6.63.3 Member Function Documentation

#### 6.63.3.1 **uint32 Oscl\_Queue\_Base::capacity () const [inline]**

Returns the allocated memory of the queue.

#### 6.63.3.2 **OSCL\_IMPORT\_REF void Oscl\_Queue\_Base::clear () [protected]**

Removes all elements.

Reimplemented in [Oscl\\_Queue< T, Alloc >](#).

#### 6.63.3.3 **OSCL\_IMPORT\_REF void Oscl\_Queue\_Base::construct (Oscl\_Opaque\_Type\_Alloc \* aType, uint32 n) [protected]**

#### 6.63.3.4 **OSCL\_IMPORT\_REF void Oscl\_Queue\_Base::construct (Oscl\_Opaque\_Type\_Alloc \* aType) [protected]**

#### 6.63.3.5 **OSCL\_IMPORT\_REF void Oscl\_Queue\_Base::destroy () [protected]**

Like an explicit destructor call.

#### 6.63.3.6 **bool Oscl\_Queue\_Base::empty () const [inline]**

True if there are no elements in the queue

#### 6.63.3.7 **OSCL\_IMPORT\_REF void Oscl\_Queue\_Base::pop () [protected]**

Removes the first element

Reimplemented in [Oscl\\_Queue< T, Alloc >](#).

#### 6.63.3.8 **OSCL\_IMPORT\_REF void Oscl\_Queue\_Base::push (const OsclAny \* x) [protected]**

Inserts a new element at the end. Queue will be grown if necessary. If allocation fails, an OSCL\_LEAVE will occur

**Parameters:**

*x* new element

#### 6.63.3.9 **OSCL\_IMPORT\_REF void Oscl\_Queue\_Base::reserve (uint32 n)**

Reallocates memory if necessary to a capacity of *n* elements. The main reason for reserve is efficiency. If you know the capacity to which your vector must grow, then it is more efficient to allocate the vector all at once rather than rely on the automatic reallocation scheme. This also helps control the invalidation of iterators.

**Parameters:**

*n* size of vector

**6.63.3.10 uint32 Oscl\_Queue\_Base::size () const [inline]**

Returns the size of the queue.

**6.63.4 Field Documentation****6.63.4.1 uint32 Oscl\_Queue\_Base::bufsize [protected]****6.63.4.2 OsclAny\* Oscl\_Queue\_Base::elems [protected]****6.63.4.3 uint32 Oscl\_Queue\_Base::ifront [protected]****6.63.4.4 uint32 Oscl\_Queue\_Base::irear [protected]****6.63.4.5 uint32 Oscl\_Queue\_Base::numelems [protected]****6.63.4.6 uint32 Oscl\_Queue\_Base::sizeof\_T [protected]**

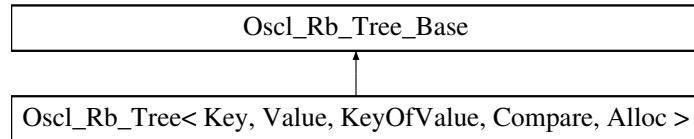
The documentation for this class was generated from the following file:

- [oscl\\_queue.h](#)

## 6.64 Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc > Class Template Reference

```
#include <oscl_tree.h>
```

Inheritance diagram for Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::



### Public Types

- `typedef Key key_type`
- `typedef Value value_type`
- `typedef value_type * pointer`
- `typedef const value_type * const_pointer`
- `typedef value_type & reference`
- `typedef const value_type & const_reference`
- `typedef Oscl_Rb_Tree_Node< Value >::link_type link_type`
- `typedef Oscl_Rb_Tree_Iterator< value_type > iterator`
- `typedef Oscl_Rb_Tree_Const_Iterator< value_type > const_iterator`
- `typedef uint32 size_type`
- `typedef int32 difference_type`

### Public Methods

- `Oscl_Rb_Tree (const Compare &comp=Compare())`
- `Oscl_Rb_Tree (const Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > &x)`
- `~Oscl_Rb_Tree ()`
- `Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > & operator= (const Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > &x)`
- `iterator begin ()`
- `const_iterator begin () const`
- `iterator end ()`
- `const_iterator end () const`
- `bool empty () const`
- `size_type size () const`
- `size_type max_size () const`
- `Oscl_Pair< iterator, bool > insert_unique (const value_type &v)`
- `iterator insert_unique (iterator position, const value_type &v)`
- `void insert_unique (const iterator first, const iterator last)`
- `void insert_unique (const value_type *first, const value_type *last)`
- `void erase (iterator position)`
- `size_type erase (const key_type &x)`
- `void erase (iterator first, iterator last)`
- `void erase (const key_type *first, const key_type *last)`



## 6.64 Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc > Class Template Reference

- void [clear \(\)](#)
- [iterator find \(const Key &k\)](#)
- [const\\_iterator find \(const Key &k\) const](#)
- [size\\_type count \(const Key &k\) const](#)
- [iterator lower\\_bound \(const Key &k\)](#)
- [const\\_iterator lower\\_bound \(const Key &k\) const](#)
- [iterator upper\\_bound \(const Key &k\)](#)
- [const\\_iterator upper\\_bound \(const Key &k\) const](#)
- [Oscl\\_Pair< iterator, iterator > equal\\_range \(const Key &k\)](#)
- [Oscl\\_Pair< const\\_iterator, const\\_iterator > equal\\_range \(const Key &k\) const](#)

template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> class Oscl\_Rb\_-Tree< Key, Value, KeyOfValue, Compare, Alloc >

### 6.64.1 Member Typedef Documentation

- 6.64.1.1 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef [Oscl\\_Rb\\_Tree\\_Const\\_Iterator<value\\_type>](#) Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::const\_iterator
- 6.64.1.2 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef const [value\\_type\\*](#) Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::const\_pointer
- 6.64.1.3 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef const [value\\_type&](#) Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::const\_reference
- 6.64.1.4 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef int32 Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::difference\_type
- 6.64.1.5 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef [Oscl\\_Rb\\_Tree\\_Iterator<value\\_type>](#) Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::iterator
- 6.64.1.6 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef Key Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::key\_type
- 6.64.1.7 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef [Oscl\\_Rb\\_Tree\\_Node<Value>::link\\_type](#) Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::link\_type
- 6.64.1.8 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef [value\\_type\\*](#) Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::pointer
- 6.64.1.9 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef [value\\_type&](#) Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::reference
- 6.64.1.10 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef uint32 Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::size\_type
- 6.64.1.11 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef Value Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::value\_type

### 6.64.2 Constructor & Destructor Documentation

- 6.64.2.1 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::Oscl\_Rb\_Tree (const Compare & *comp* = Compare() ) [inline]
- 6.64.2.2 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::Oscl\_Rb\_Tree (const Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc > & *x*) [inline]
- 6.64.2.3 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::~Oscl\_Rb\_Tree () [inline]



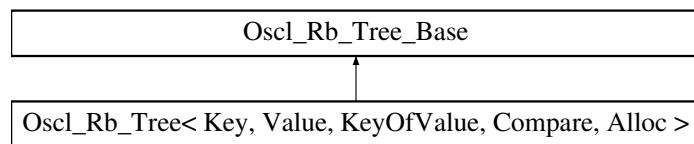
## 6.64 Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc > Class Template Reference

- [oscl\\_tree.h](#)

## 6.65 Oscl\_Rb\_Tree\_Base Class Reference

```
#include <oscl_tree.h>
```

Inheritance diagram for Oscl\_Rb\_Tree\_Base::



### Public Types

- `typedef Oscl_Rb_Tree_Node_Base::base_link_type base_link_type`

### Public Methods

- `OSCL_IMPORT_REF void rotate_left (base_link_type x, base_link_type &root)`
- `OSCL_IMPORT_REF void rotate_right (base_link_type x, base_link_type &root)`
- `OSCL_IMPORT_REF void rebalance (base_link_type x, base_link_type &root)`
- `OSCL_IMPORT_REF base_link_type rebalance_for_erase (base_link_type z, base_link_type &root, base_link_type &leftmost, base_link_type &rightmost)`

#### 6.65.1 Member Typedef Documentation

##### 6.65.1.1 `typedef Oscl_Rb_Tree_Node_Base::base_link_type Oscl_Rb_Tree_Base::base_link_type`

#### 6.65.2 Member Function Documentation

##### 6.65.2.1 `OSCL_IMPORT_REF void Oscl_Rb_Tree_Base::rebalance (base_link_type x, base_link_type &root)`

##### 6.65.2.2 `OSCL_IMPORT_REF base_link_type Oscl_Rb_Tree_Base::rebalance_for_erase (base_link_type z, base_link_type &root, base_link_type &leftmost, base_link_type &rightmost)`

##### 6.65.2.3 `OSCL_IMPORT_REF void Oscl_Rb_Tree_Base::rotate_left (base_link_type x, base_link_type &root)`

##### 6.65.2.4 `OSCL_IMPORT_REF void Oscl_Rb_Tree_Base::rotate_right (base_link_type x, base_link_type &root)`

The documentation for this class was generated from the following file:

- `oscl_tree.h`

## 6.66 Oscl\_Rb\_Tree\_Const\_Iterator< Value > Struct Template Reference

```
#include <oscl_tree.h>
```

### Public Types

- `typedef Value value_type`
- `typedef const value_type & reference`
- `typedef const value_type * pointer`
- `typedef Oscl_Rb_Tree_Const_Iterator< Value > const_iterator`
- `typedef Oscl_Rb_Tree_Const_Iterator< Value > self`
- `typedef Oscl_Rb_Tree_Node_Base * base_link_type`
- `typedef Oscl_Rb_Tree_Node< Value > * link_type`

### Public Methods

- `Oscl_Rb_Tree_Const_Iterator()`
- `Oscl_Rb_Tree_Const_Iterator(link_type x)`
- `Oscl_Rb_Tree_Const_Iterator(const const_iterator &it)`
- `reference operator * () const`
- `pointer operator → () const`
- `bool operator==(const self &x)`
- `bool operator!=(const self &x)`
- `self & operator++()`
- `self operator++(int)`
- `self & operator--()`
- `self operator--(int)`

### Data Fields

- `base_link_type node`

```
template<class Value> struct Oscl_Rb_Tree_Const_Iterator< Value >
```

### 6.66.1 Member Typedef Documentation

- 6.66.1.1 `template<class Value> typedef Oscl_Rb_Tree_Node_Base* Oscl_Rb_Tree_Const_Iterator< Value >::base_link_type`
- 6.66.1.2 `template<class Value> typedef Oscl_Rb_Tree_Const_Iterator<Value> Oscl_Rb_Tree_Const_Iterator< Value >::const_iterator`
- 6.66.1.3 `template<class Value> typedef Oscl_Rb_Tree_Node<Value>* Oscl_Rb_Tree_Const_Iterator< Value >::link_type`
- 6.66.1.4 `template<class Value> typedef const value_type* Oscl_Rb_Tree_Const_Iterator< Value >::pointer`
- 6.66.1.5 `template<class Value> typedef const value_type& Oscl_Rb_Tree_Const_Iterator< Value >::reference`
- 6.66.1.6 `template<class Value> typedef Oscl_Rb_Tree_Const_Iterator<Value> Oscl_Rb_Tree_Const_Iterator< Value >::self`
- 6.66.1.7 `template<class Value> typedef Value Oscl_Rb_Tree_Const_Iterator< Value >::value_type`

### 6.66.2 Constructor & Destructor Documentation

- 6.66.2.1 `template<class Value> Oscl_Rb_Tree_Const_Iterator< Value >::Oscl_Rb_Tree_Const_Iterator () [inline]`
- 6.66.2.2 `template<class Value> Oscl_Rb_Tree_Const_Iterator< Value >::Oscl_Rb_Tree_Const_Iterator (link_type x) [inline]`
- 6.66.2.3 `template<class Value> Oscl_Rb_Tree_Const_Iterator< Value >::Oscl_Rb_Tree_Const_Iterator (const const_iterator & it) [inline]`

### 6.66.3 Member Function Documentation

- 6.66.3.1 `template<class Value> reference Oscl_Rb_Tree_Const_Iterator< Value >::operator * () const [inline]`
- 6.66.3.2 `template<class Value> bool Oscl_Rb_Tree_Const_Iterator< Value >::operator!= (const self & x) [inline]`
- 6.66.3.3 `template<class Value> self Oscl_Rb_Tree_Const_Iterator< Value >::operator++ (int) [inline]`
- 6.66.3.4 `template<class Value> self& Oscl_Rb_Tree_Const_Iterator< Value >::operator++ () [inline]`
- 6.66.3.5 `template<class Value> self Oscl_Rb_Tree_Const_Iterator< Value >::operator- (int) [inline]`
- 6.66.3.6 `template<class Value> self& Oscl_Rb_Tree_Const_Iterator< Value >::operator- () [inline]`

- 
- [oscl\\_tree.h](#)

## 6.67 Oscl\_Rb\_Tree\_Iterator< Value > Struct Template Reference

```
#include <oscl_tree.h>
```

### Public Types

- `typedef Value value_type`
- `typedef value_type & reference`
- `typedef value_type * pointer`
- `typedef Oscl_Rb_Tree_Iterator< Value > iterator`
- `typedef Oscl_Rb_Tree_Iterator< Value > self`
- `typedef Oscl_Rb_Tree_Node_Base * base_link_type`
- `typedef Oscl_Rb_Tree_Node< Value > * link_type`

### Public Methods

- `Oscl_Rb_Tree_Iterator ()`
- `Oscl_Rb_Tree_Iterator (link_type x)`
- `Oscl_Rb_Tree_Iterator (const iterator &it)`
- `reference operator * () const`
- `pointer operator → () const`
- `bool operator== (const self &x)`
- `bool operator!= (const self &x)`
- `self & operator++ ()`
- `self operator++ (int)`
- `self & operator– ()`
- `self operator– (int)`

### Data Fields

- `base_link_type node`

template<class Value> struct Oscl\_Rb\_Tree\_Iterator< Value >

### 6.67.1 Member Typedef Documentation

- 6.67.1.1 template<class Value> typedef Oscl\_Rb\_Tree\_Node\_Base\* Oscl\_Rb\_Tree\_Iterator< Value >::base\_link\_type
- 6.67.1.2 template<class Value> typedef Oscl\_Rb\_Tree\_Iterator<Value> Oscl\_Rb\_Tree\_Iterator< Value >::iterator
- 6.67.1.3 template<class Value> typedef Oscl\_Rb\_Tree\_Node<Value>\* Oscl\_Rb\_Tree\_Iterator< Value >::link\_type
- 6.67.1.4 template<class Value> typedef value\_type\* Oscl\_Rb\_Tree\_Iterator< Value >::pointer
- 6.67.1.5 template<class Value> typedef value\_type& Oscl\_Rb\_Tree\_Iterator< Value >::reference
- 6.67.1.6 template<class Value> typedef Oscl\_Rb\_Tree\_Iterator<Value> Oscl\_Rb\_Tree\_Iterator< Value >::self
- 6.67.1.7 template<class Value> typedef Value Oscl\_Rb\_Tree\_Iterator< Value >::value\_type

### 6.67.2 Constructor & Destructor Documentation

- 6.67.2.1 template<class Value> Oscl\_Rb\_Tree\_Iterator< Value >::Oscl\_Rb\_Tree\_Iterator () [inline]
- 6.67.2.2 template<class Value> Oscl\_Rb\_Tree\_Iterator< Value >::Oscl\_Rb\_Tree\_Iterator (link\_type x) [inline]
- 6.67.2.3 template<class Value> Oscl\_Rb\_Tree\_Iterator< Value >::Oscl\_Rb\_Tree\_Iterator (const iterator & it) [inline]

### 6.67.3 Member Function Documentation

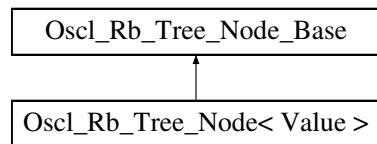
- 6.67.3.1 template<class Value> reference Oscl\_Rb\_Tree\_Iterator< Value >::operator \* () const [inline]
- 6.67.3.2 template<class Value> bool Oscl\_Rb\_Tree\_Iterator< Value >::operator!= (const self & x) [inline]
- 6.67.3.3 template<class Value> self Oscl\_Rb\_Tree\_Iterator< Value >::operator++ (int) [inline]
- 6.67.3.4 template<class Value> self& Oscl\_Rb\_Tree\_Iterator< Value >::operator++ () [inline]
- 6.67.3.5 template<class Value> self Oscl\_Rb\_Tree\_Iterator< Value >::operator- (int) [inline]
- 6.67.3.6 template<class Value> self& Oscl\_Rb\_Tree\_Iterator< Value >::operator- () [inline]
- 6.67.3.7 template<class Value> pointer Oscl\_Rb\_Tree\_Iterator< Value >::operator -> () const [inline]

- [oscl\\_tree.h](#)

## 6.68 Oscl\_Rb\_Tree\_Node< Value > Struct Template Reference

```
#include <oscl_tree.h>
```

Inheritance diagram for Oscl\_Rb\_Tree\_Node< Value >::



### Public Types

- [typedef Value value\\_type](#)
- [typedef Oscl\\_Rb\\_Tree\\_Node< Value > \\* link\\_type](#)

### Data Fields

- [value\\_type value](#)

```
template<class Value> struct Oscl_Rb_Tree_Node< Value >
```

#### 6.68.1 Member Typedef Documentation

**6.68.1.1 template<class Value> typedef Oscl\_Rb\_Tree\_Node<Value>\* Oscl\_Rb\_Tree\_Node< Value >::link\_type**

**6.68.1.2 template<class Value> typedef Value Oscl\_Rb\_Tree\_Node< Value >::value\_type**

#### 6.68.2 Field Documentation

**6.68.2.1 template<class Value> [value\\_type](#) Oscl\_Rb\_Tree\_Node< Value >::value**

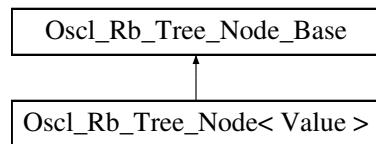
The documentation for this struct was generated from the following file:

- [oscl\\_tree.h](#)

## 6.69 Oscl\_Rb\_Tree\_Node\_Base Struct Reference

```
#include <oscl_tree.h>
```

Inheritance diagram for Oscl\_Rb\_Tree\_Node\_Base::



### Public Types

- `typedef Oscl_Rb_Tree_Node_Base * base_link_type`
- `typedef enum Oscl_Rb_Tree_Node_Base::RedBl color_type`
- `enum RedBl { red, black }`

### Static Public Methods

- `base_link_type minimum (base_link_type x)`
- `base_link_type maximum (base_link_type x)`

### Data Fields

- `color_type color`
- `base_link_type parent`
- `base_link_type left`
- `base_link_type right`

#### 6.69.1 Member Typedef Documentation

**6.69.1.1 `typedef Oscl_Rb_Tree_Node_Base* Oscl_Rb_Tree_Node_Base::base_link_type`**

**6.69.1.2 `typedef enum Oscl_Rb_Tree_Node_Base::RedBl Oscl_Rb_Tree_Node_Base::color_type`**

#### 6.69.2 Member Enumeration Documentation

**6.69.2.1 `enum Oscl_Rb_Tree_Node_Base::RedBl`**

Enumeration values:

`red`

`black`

### 6.69.3 Member Function Documentation

6.69.3.1 **base\_link\_type** Oscl\_Rb\_Tree\_Node\_Base::maximum (**base\_link\_type** *x*) [inline, static]

6.69.3.2 **base\_link\_type** Oscl\_Rb\_Tree\_Node\_Base::minimum (**base\_link\_type** *x*) [inline, static]

### 6.69.4 Field Documentation

6.69.4.1 **color\_type** Oscl\_Rb\_Tree\_Node\_Base::color

6.69.4.2 **base\_link\_type** Oscl\_Rb\_Tree\_Node\_Base::left

6.69.4.3 **base\_link\_type** Oscl\_Rb\_Tree\_Node\_Base::parent

6.69.4.4 **base\_link\_type** Oscl\_Rb\_Tree\_Node\_Base::right

The documentation for this struct was generated from the following file:

- [oscl\\_tree.h](#)

## 6.70 Oscl\_Select1st< V, U > Struct Template Reference

```
#include <oscl_map.h>
```

### Public Methods

- const U & [operator\(\)](#) (const V &x) const

```
template<class V, class U> struct Oscl_Select1st< V, U >
```

#### 6.70.1 Member Function Documentation

**6.70.1.1 template<class V, class U> const U& Oscl\_Select1st< V, U >::operator() (const V & x)  
const [inline]**

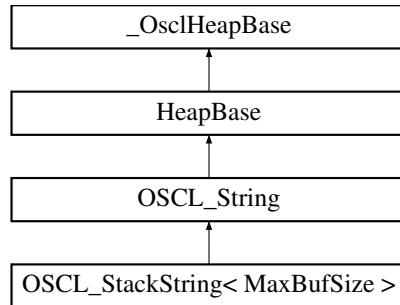
The documentation for this struct was generated from the following file:

- [oscl\\_map.h](#)

## 6.71 OSCL\_StackString< MaxBufSize > Class Template Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL\_StackString< MaxBufSize >::



### Public Types

- `typedef OSCL_String::chartype chartype`

### Public Methods

- `OSCL_StackString ()`
- `OSCL_StackString (const OSCL_StackString &src)`
- `OSCL_StackString (const OSCL_String &src)`
- `OSCL_StackString (const chartype *cstr)`
- `OSCL_StackString (const chartype *buf, uint32 length)`
- `~OSCL_StackString ()`
- `uint32 get_size () const`
- `uint32 get_maxsize () const`
- `const chartype * get_cstr () const`
- `chartype * get_str () const`
- `OSCL_StackString & operator= (const OSCL_StackString &src)`
- `OSCL_StackString & operator= (const OSCL_String &src)`
- `OSCL_StackString & operator= (const chartype *cstr)`
- `void set (const chartype *buf, uint32 length)`

### Friends

- class `OSCL_String`

#### 6.71.1 Detailed Description

`template<uint32 MaxBufSize> class OSCL_StackString< MaxBufSize >`

`OSCL_StackString` is a simple string class, compatible with regular character array strings.

The string array is fixed length, is allocated from the stack, and is modifiable. Operations that update the string will automatically truncate it to fit the fixed size storage. This is recommended for use for short strings (<255). Use `OSCL_HeapString` for very large strings to avoid stack overflow.

**Parameters:**

*C*: type of character.

*MaxBufSize*: maximum string length not including null terminator.

## 6.71.2 Member Typedef Documentation

### 6.71.2.1 template<uint32 MaxBufSize> typedef OSCL\_String::chartype OSCL\_StackString< MaxBufSize >::chartype

Reimplemented from [OSCL\\_String](#).

## 6.71.3 Friends And Related Function Documentation

### 6.71.3.1 template<uint32 MaxBufSize> friend class OSCL\_String [friend]

The documentation for this class was generated from the following file:

- [oscl\\_string\\_containers.h](#)

## 6.72 oscl\_stat\_buf Struct Reference

```
#include <oscl_file_dir_utils.h>
```

### Data Fields

- uint32 [mode](#)
- uint32 [perms](#)

#### 6.72.1 Field Documentation

##### 6.72.1.1 uint32 oscl\_stat\_buf::mode

##### 6.72.1.2 uint32 oscl\_stat\_buf::perms

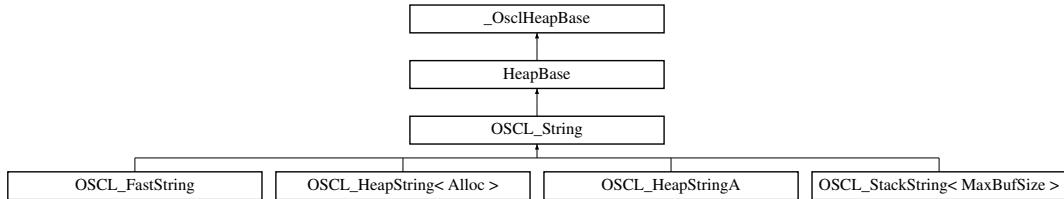
The documentation for this struct was generated from the following file:

- [oscl\\_file\\_dir\\_utils.h](#)

## 6.73 OSCL\_String Class Reference

```
#include <oscl_string.h>
```

Inheritance diagram for OSCL\_String::



### Public Types

- `typedef char chartype`

### Public Methods

- `virtual uint32 get_size () const=0`
- `virtual uint32 get_maxsize () const=0`
- `virtual const chartype * get_cstr () const=0`
- `virtual OSCL_IMPORT_REF bool is_writable () const`
- `virtual chartype * get_str () const=0`
- `OSCL_IMPORT_REF OSCL_String & operator= (const OSCL_String &src)`
- `OSCL_IMPORT_REF OSCL_String & operator= (const chartype *cstr)`
- `OSCL_IMPORT_REF OSCL_String & operator+= (const OSCL_String &src)`
- `OSCL_IMPORT_REF OSCL_String & operator+= (const chartype *cstr)`
- `OSCL_IMPORT_REF OSCL_String & operator+= (const chartype c)`
- `OSCL_IMPORT_REF bool operator== (const OSCL_String &src) const`
- `OSCL_IMPORT_REF bool operator!= (const OSCL_String &src) const`
- `OSCL_IMPORT_REF bool operator< (const OSCL_String &src) const`
- `OSCL_IMPORT_REF bool operator<= (const OSCL_String &src) const`
- `OSCL_IMPORT_REF bool operator> (const OSCL_String &src) const`
- `OSCL_IMPORT_REF bool operator>= (const OSCL_String &src) const`
- `OSCL_IMPORT_REF bool operator== (const chartype *cstr) const`
- `OSCL_IMPORT_REF chartype operator[] (uint32 index) const`
- `virtual OSCL_IMPORT_REF chartype read (uint32 index) const`
- `virtual OSCL_IMPORT_REF int8 hash () const`
- `virtual OSCL_IMPORT_REF void write (uint32 index, chartype c)`
- `virtual OSCL_IMPORT_REF void write (uint32 offset, uint32 length, const chartype *buf)`

### Protected Methods

- `OSCL_IMPORT_REF OSCL_String ()`
- `virtual OSCL_IMPORT_REF ~OSCL_String ()`
- `virtual void set_rep (const chartype *cstr)=0`
- `virtual void append_rep (const chartype *cstr)=0`
- `virtual void set_rep (const OSCL_String &src)=0`
- `virtual void append_rep (const OSCL_String &src)=0`
- `virtual void set_len (uint32 len)=0`

### 6.73.1 Detailed Description

A common base class for string classes with "char" character format

### 6.73.2 Member Typedef Documentation

#### 6.73.2.1 `typedef char OSCL_String::chartype`

Reimplemented in [OSCL\\_HeapString< Alloc >](#), [OSCL\\_HeapStringA](#), [OSCL\\_StackString< MaxBufSize >](#), [OSCL\\_FastString](#), and [OSCL\\_HeapString< OsclMemAllocator >](#).

### 6.73.3 Constructor & Destructor Documentation

#### 6.73.3.1 `OSCL_IMPORT_REF OSCL_String::OSCL_String()` [protected]

#### 6.73.3.2 `virtual OSCL_IMPORT_REF OSCL_String::~OSCL_String()` [protected, virtual]

### 6.73.4 Member Function Documentation

#### 6.73.4.1 `virtual void OSCL_String::append_rep(const OSCL_String & src)` [protected, pure virtual]

Append the input string to the current string. The string may be truncated to fit the available storage.

#### 6.73.4.2 `virtual void OSCL_String::append_rep(const chartype * cstr)` [protected, pure virtual]

Append the input null-terminated string to the current string. The string may be truncated to fit the available storage.

#### 6.73.4.3 `virtual const chartype* OSCL_String::get_cstr()` [pure virtual]

This function returns the C-style string for read access.

Implemented in [OSCL\\_HeapString< Alloc >](#), [OSCL\\_HeapStringA](#), [OSCL\\_StackString< MaxBufSize >](#), [OSCL\\_FastString](#), and [OSCL\\_HeapString< OsclMemAllocator >](#).

#### 6.73.4.4 `virtual uint32 OSCL_String::get_maxsize()` [pure virtual]

This function returns the maximum available storage size, not including null terminator. The maximum size may be larger than the current string size.

Implemented in [OSCL\\_HeapString< Alloc >](#), [OSCL\\_HeapStringA](#), [OSCL\\_StackString< MaxBufSize >](#), [OSCL\\_FastString](#), and [OSCL\\_HeapString< OsclMemAllocator >](#).

#### 6.73.4.5 `virtual uint32 OSCL_String::get_size()` [pure virtual]

This function returns the string size not including the null-terminator.

Implemented in [OSCL\\_HeapString< Alloc >](#), [OSCL\\_HeapStringA](#), [OSCL\\_StackString< MaxBufSize >](#), [OSCL\\_FastString](#), and [OSCL\\_HeapString< OsclMemAllocator >](#).

#### 6.73.4.6 virtual **chartype\*** OSCL\_String::get\_str () [pure virtual]

This function returns the C-style string for write access. If the string is not writable it returns NULL.

Implemented in [OSCL\\_HeapString< Alloc >](#), [OSCL\\_HeapStringA](#), [OSCL\\_StackString< MaxBufSize >](#), [OSCL\\_FastString](#), and [OSCL\\_HeapString< OsclMemAllocator >](#).

#### 6.73.4.7 virtual OSCL\_IMPORT\_REF int8 OSCL\_String::hash () [virtual]

This function performs a hash operation on the string. If the string is not writable, the function leaves.

#### 6.73.4.8 virtual OSCL\_IMPORT\_REF bool OSCL\_String::is\_writable () [virtual]

This function returns true if the string is writable.

#### 6.73.4.9 OSCL\_IMPORT\_REF bool OSCL\_String::operator!= (const OSCL\_String & src) const

#### 6.73.4.10 OSCL\_IMPORT\_REF OSCL\_String& OSCL\_String::operator+= (const **chartype** c)

Append operator. This operator appends the input character to this object. The string may be truncated to fit available storage.

#### 6.73.4.11 OSCL\_IMPORT\_REF OSCL\_String& OSCL\_String::operator+= (const **chartype** \* cstr)

Append operator. This operator appends the input string to this object. The string may be truncated to fit available storage.

**am:** null-terminated string

#### 6.73.4.12 OSCL\_IMPORT\_REF OSCL\_String& OSCL\_String::operator+= (const OSCL\_String & src)

Append operator. This operator appends the input string to this object. The string may be truncated to fit available storage.

#### 6.73.4.13 OSCL\_IMPORT\_REF bool OSCL\_String::operator< (const OSCL\_String & src) const

#### 6.73.4.14 OSCL\_IMPORT\_REF bool OSCL\_String::operator<= (const OSCL\_String & src) const

#### 6.73.4.15 OSCL\_IMPORT\_REF OSCL\_String& OSCL\_String::operator= (const **chartype** \* cstr)

Assignment operator

**am:** null-terminated string

Reimplemented in [OSCL\\_HeapString< Alloc >](#), [OSCL\\_HeapStringA](#), [OSCL\\_StackString< MaxBufSize >](#), [OSCL\\_FastString](#), and [OSCL\\_HeapString< OsclMemAllocator >](#).

#### 6.73.4.16 **OSCL\_IMPORT\_REF OSCL\_String& OSCL\_String::operator= (const OSCL\_String & src)**

Assignment operator

Reimplemented in [OSCL\\_HeapString< Alloc >](#), [OSCL\\_HeapStringA](#), [OSCL\\_StackString< MaxBufSize >](#), and [OSCL\\_HeapString< OsclMemAllocator >](#).

#### 6.73.4.17 **OSCL\_IMPORT\_REF bool OSCL\_String::operator== (const chartype \* cstr) const**

Comparison operator

**am:** null-terminated string

#### 6.73.4.18 **OSCL\_IMPORT\_REF bool OSCL\_String::operator== (const OSCL\_String & src) const**

Comparison operators

#### 6.73.4.19 **OSCL\_IMPORT\_REF bool OSCL\_String::operator> (const OSCL\_String & src) const**

#### 6.73.4.20 **OSCL\_IMPORT\_REF bool OSCL\_String::operator>= (const OSCL\_String & src) const**

#### 6.73.4.21 ]

**OSCL\_IMPORT\_REF chartype OSCL\_String::operator[ ] (uint32 index) const**

This is subscript notation to access a character at the given position. If the index is outside the current size range then the function leaves.

#### 6.73.4.22 **virtual OSCL\_IMPORT\_REF chartype OSCL\_String::read (uint32 index) const [virtual]**

This function returns the character at the given position. If the index is outside the current size range then the function leaves.

#### 6.73.4.23 **virtual void OSCL\_String::set\_len (uint32 len) [protected, pure virtual]**

Update the length of the string. This function will only be called when the string is writable.

**6.73.4.24 virtual void OSCL\_String::set\_rep (const OSCL\_String & src) [protected, pure virtual]**

Set string representation to input string.

**6.73.4.25 virtual void OSCL\_String::set\_rep (const chartype \* cstr) [protected, pure virtual]**

Set string representation to input null-terminated string.

**6.73.4.26 virtual OSCL\_IMPORT\_REF void OSCL\_String::write (uint32 offset, uint32 length, const chartype \* buf) [virtual]**

This function replaces characters at the specified offset within the current string. If the string is not writable, the function leaves. The characters may be truncated to fit the current storage.

**Parameters:**

*offset*: the offset into the existing string buffer

*length*: number of characters to copy.

*ptr*: character buffer, not necessarily null-terminated.

**6.73.4.27 virtual OSCL\_IMPORT\_REF void OSCL\_String::write (uint32 index, chartype c) [virtual]**

This function stores a character at the specified position. If the string is not writable, the function leaves. If the index is outside the current size range then the function leaves.

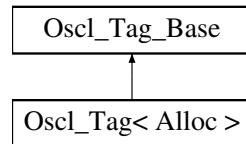
The documentation for this class was generated from the following file:

- [oscl\\_string.h](#)

## 6.74 Oscl\_Tag< Alloc > Struct Template Reference

```
#include <oscl_tagtree.h>
```

Inheritance diagram for Oscl\_Tag< Alloc >::



### Public Methods

- [Oscl\\_Tag](#) (const Oscl\_Tag< Alloc > &t)
- [Oscl\\_Tag](#) (const [tag\\_base\\_type](#) &t)
- [~Oscl\\_Tag](#) ()
- bool [operator<](#) (const Oscl\_Tag< Alloc > &x) const

### Data Fields

- [Oscl\\_TAlloc< tag\\_base\\_unit, Alloc > tagAllocator](#)
- [tag\\_base\\_type tag](#)

```
template<class Alloc> struct Oscl_Tag< Alloc >
```

#### 6.74.1 Constructor & Destructor Documentation

**6.74.1.1 template<class Alloc> Oscl\_Tag< Alloc >::Oscl\_Tag (const Oscl\_Tag< Alloc > & t)**  
`[inline]`

**6.74.1.2 template<class Alloc> Oscl\_Tag< Alloc >::Oscl\_Tag (const tag\_base\_type & t)**  
`[inline]`

**6.74.1.3 template<class Alloc> Oscl\_Tag< Alloc >::~Oscl\_Tag ()** `[inline]`

#### 6.74.2 Member Function Documentation

**6.74.2.1 template<class Alloc> bool Oscl\_Tag< Alloc >::operator< (const Oscl\_Tag< Alloc > & x) const** `[inline]`

#### 6.74.3 Field Documentation

**6.74.3.1 template<class Alloc> tag\_base\_type Oscl\_Tag< Alloc >::tag**

**6.74.3.2 template<class Alloc> Oscl\_TAlloc<tag\_base\_unit, Alloc> Oscl\_Tag< Alloc >::tagAllocator**

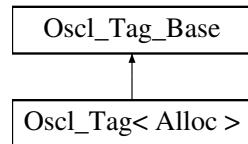
The documentation for this struct was generated from the following file:

- [oscl\\_tagtree.h](#)

## 6.75 Oscl\_Tag\_Base Struct Reference

```
#include <oscl_tagtree.h>
```

Inheritance diagram for Oscl\_Tag\_Base::



### Public Types

- `typedef char tag_base_unit`
- `typedef tag_base_unit * tag_base_type`
- `typedef uint32 size_type`

### Public Methods

- `bool operator() (const tag_base_type &x, const tag_base_type &y) const`
- `size_type tag_len (const tag_base_type &t) const`
- `tag_base_type tag_copy (tag_base_type &dest, const tag_base_type &src) const`
- `int32 tag_cmp (const tag_base_type &x, const tag_base_type &y) const`
- `OSCL_IMPORT_REF tag_base_type tag_ancestor (tag_base_type &dest, const tag_base_type &src) const`
- `OSCL_IMPORT_REF size_type tag_depth (const tag_base_type &t) const`

### 6.75.1 Member Typedef Documentation

- 6.75.1.1 `typedef uint32 Oscl_Tag_Base::size_type`
- 6.75.1.2 `typedef tag_base_unit* Oscl_Tag_Base::tag_base_type`
- 6.75.1.3 `typedef char Oscl_Tag_Base::tag_base_unit`

### 6.75.2 Member Function Documentation

- 6.75.2.1 `bool Oscl_Tag_Base::operator() (const tag_base_type & x, const tag_base_type & y) const [inline]`
- 6.75.2.2 `OSCL_IMPORT_REF tag_base_type Oscl_Tag_Base::tag_ancestor (tag_base_type & dest, const tag_base_type & src) const`
- 6.75.2.3 `int32 Oscl_Tag_Base::tag_cmp (const tag_base_type & x, const tag_base_type & y) const [inline]`
- 6.75.2.4 `tag_base_type Oscl_Tag_Base::tag_copy (tag_base_type & dest, const tag_base_type & src) const [inline]`
- 6.75.2.5 `OSCL_IMPORT_REF size_type Oscl_Tag_Base::tag_depth (const tag_base_type & t) const`
- 6.75.2.6 `size_type Oscl_Tag_Base::tag_len (const tag_base_type & t) const [inline]`

The documentation for this struct was generated from the following file:

- [oscl\\_tagtree.h](#)

## 6.76 Oscl\_TagTree< T, Alloc > Class Template Reference

```
#include <oscl_tagtree.h>
```

### Public Types

- typedef [Oscl\\_Tag< Alloc > tag\\_type](#)
- typedef tag\_type::tag\_base\_type [tag\\_base\\_type](#)
- typedef [Oscl\\_Vector< Node \\*, Alloc > children\\_type](#)
- typedef [Node node\\_type](#)
- typedef [node\\_type \\* node\\_ptr](#)
- typedef [Oscl\\_Map< const tag\\_base\\_type, node\\_ptr, Alloc, Oscl\\_Tag\\_Base > map\\_type](#)
- typedef map\_type::size\_type [size\\_type](#)
- typedef map\_type::value\_type [value\\_type](#)
- typedef [Oscl\\_Pair< iterator, bool > pair\\_iterator\\_bool](#)

### Public Methods

- [Oscl\\_TagTree \(size\\_type max\\_depth=0\)](#)
- [Oscl\\_TagTree \(const Oscl\\_TagTree< T, Alloc > &x\)](#)
- [Oscl\\_TagTree< T, Alloc > & operator=\(const Oscl\\_TagTree< T, Alloc > &x\)](#)
- [~Oscl\\_TagTree \(\)](#)
- [iterator begin \(\)](#)
- [const\\_iterator begin \(\) const](#)
- [iterator end \(\)](#)
- [const\\_iterator end \(\) const](#)
- [bool empty \(\) const](#)
- [size\\_type size \(\) const](#)
- [T & operator\[\] \(const tag\\_base\\_type &t\)](#)
- [pair\\_iterator\\_bool insert \(const tag\\_base\\_type &t, const T &x\)](#)
- [void erase \(iterator position\)](#)
- [size\\_type erase \(const tag\\_base\\_type &x\)](#)
- [void clear \(\)](#)
- [iterator find \(const tag\\_base\\_type &x\)](#)
- [size\\_type count \(const tag\\_base\\_type &x\) const](#)

### 6.76.1 Detailed Description

**template<class T, class Alloc> class Oscl\_TagTree< T, Alloc >**

Oscl\_TagTree Class.

## 6.76.2 Member Typedef Documentation

- 6.76.2.1 template<class T, class Alloc> typedef **Oscl\_Vector<Node\*, Alloc>** Oscl\_TagTree< T, Alloc >::children\_type
- 6.76.2.2 template<class T, class Alloc> typedef **Oscl\_Map<const tag\_base\_type, node\_ptr, Alloc, Oscl\_Tag\_Base>** Oscl\_TagTree< T, Alloc >::map\_type
- 6.76.2.3 template<class T, class Alloc> typedef **node\_type\*** Oscl\_TagTree< T, Alloc >::node\_ptr
- 6.76.2.4 template<class T, class Alloc> typedef **Node** Oscl\_TagTree< T, Alloc >::node\_type
- 6.76.2.5 template<class T, class Alloc> typedef **Oscl\_Pair<iterator, bool>** Oscl\_TagTree< T, Alloc >::pair\_iterator\_bool
- 6.76.2.6 template<class T, class Alloc> typedef map\_type::size\_type Oscl\_TagTree< T, Alloc >::size\_type
- 6.76.2.7 template<class T, class Alloc> typedef tag\_type::tag\_base\_type Oscl\_TagTree< T, Alloc >::tag\_base\_type
- 6.76.2.8 template<class T, class Alloc> typedef **Oscl\_Tag<Alloc>** Oscl\_TagTree< T, Alloc >::tag\_type
- 6.76.2.9 template<class T, class Alloc> typedef map\_type::value\_type Oscl\_TagTree< T, Alloc >::value\_type

## 6.76.3 Constructor & Destructor Documentation

- 6.76.3.1 template<class T, class Alloc> Oscl\_TagTree< T, Alloc >::Oscl\_TagTree (**size\_type max\_depth = 0**) [inline]

Creates a tag tree with only a root node with tag ""

- 6.76.3.2 template<class T, class Alloc> Oscl\_TagTree< T, Alloc >::Oscl\_TagTree (const Oscl\_TagTree< T, Alloc > & x) [inline]

Copy constructor

- 6.76.3.3 template<class T, class Alloc> Oscl\_TagTree< T, Alloc >::~Oscl\_TagTree () [inline]

Destructor

## 6.76.4 Member Function Documentation

- 6.76.4.1 template<class T, class Alloc> const\_iterator Oscl\_TagTree< T, Alloc >::begin () const [inline]

Returns an iterator pointing to the first node in the tree.

**6.76.4.2 template<class T, class Alloc> iterator Oscl\_TagTree< T, Alloc >::begin () [inline]**

Returns an iterator pointing to the first node in the tree.

**6.76.4.3 template<class T, class Alloc> void Oscl\_TagTree< T, Alloc >::clear () [inline]**

Erases the entire tag tree.

**6.76.4.4 template<class T, class Alloc> size\_type Oscl\_TagTree< T, Alloc >::count (const tag\_base\_type & x) const [inline]**

Returns the number of elements with key x. This can only be 0 or 1..

**6.76.4.5 template<class T, class Alloc> bool Oscl\_TagTree< T, Alloc >::empty () const [inline]**

Returns true if tree size is 0

**6.76.4.6 template<class T, class Alloc> const\_iterator Oscl\_TagTree< T, Alloc >::end () const [inline]**

Returns a const iterator pointing to the end of the tree.

**6.76.4.7 template<class T, class Alloc> iterator Oscl\_TagTree< T, Alloc >::end () [inline]**

Returns an iterator pointing to the end of the tree.

**6.76.4.8 template<class T, class Alloc> size\_type Oscl\_TagTree< T, Alloc >::erase (const tag\_base\_type & x) [inline]**

Erases the node with tag x. If the node has children, then the node will not be erased from the tree. It will be replaced with the default node value

**Parameters:**

*x* Tag of node to erase

**Returns:**

Returns the number of nodes erased. Since one-to-one mapping between nodes and tags, this will be either 0 or 1

**6.76.4.9 template<class T, class Alloc> void Oscl\_TagTree< T, Alloc >::erase (iterator position) [inline]**

Erases the element pointed to by the iterator. If the node has children, then the node will not be erased from the tree. It will be replaced with the default node value.

**Parameters:**

*position* Iterator pointing to the node to be erased

**6.76.4.10 template<class T, class Alloc> iterator Oscl\_TagTree< T, Alloc >::find (const tag\_base\_type & x) [inline]**

Finds an element whose key is x

**Returns:**

returns an iterator to the element with key x. If no element is found, returns [end\(\)](#)

**6.76.4.11 template<class T, class Alloc> pair\_iterator\_bool Oscl\_TagTree< T, Alloc >::insert (const tag\_base\_type & t, const T & x) [inline]**

Inserts x into the tree and associates it with tag t. If the tag already exists x will not be inserted, and an iterator pointing to the existing node with tag t is returned.

**Parameters:**

*t* tag to use

*x* element to insert

**Returns:**

Returns a pair of parameters, iterator and bool. The iterator points to the inserted node containing x. If the tag t already existed, then the iterator points to the node associated with tag t. The bool is true if x was inserted and false if it was not inserted due to an existing node with tag t.

**6.76.4.12 template<class T, class Alloc> Oscl\_TagTree<T, Alloc>& Oscl\_TagTree< T, Alloc >::operator= (const Oscl\_TagTree< T, Alloc > & x) [inline]**

Assignment operator

**6.76.4.13 ]**

template<class T, class Alloc> T& Oscl\_TagTree< T, Alloc >::operator[] (const tag\_base\_type & t) [inline]

Returns a reference to the object that is associated with a particular tag. If the map does not already contain such an object, operator[] inserts the default object T().

**6.76.4.14 template<class T, class Alloc> size\_type Oscl\_TagTree< T, Alloc >::size () const [inline]**

Returns the number of nodes stored in the tree

The documentation for this class was generated from the following file:

- [oscl\\_tagtree.h](#)

## 6.77 Oscl\_TagTree< T, Alloc >::const\_iterator Struct Reference

```
#include <oscl_tagtree.h>
```

### Public Types

- `typedef const node_type & reference`
- `typedef const node_type * pointer`
- `typedef map_type::const_iterator mapiter`
- `typedef const_iterator self`

### Public Methods

- `const_iterator()`
- `const_iterator(mapiter x)`
- `const_iterator(const const_iterator &it)`
- `reference operator*() const`
- `pointer operator->() const`
- `bool operator==(const self &x)`
- `bool operator!=(const self &x)`
- `self & operator++()`
- `self operator++(int)`
- `self & operator--()`
- `self operator--(int)`

### Data Fields

- `mapiter mapit`

template<class T, class Alloc> struct Oscl\_TagTree< T, Alloc >::const\_iterator

### 6.77.1 Member Typedef Documentation

- 6.77.1.1 template<class T, class Alloc> typedef map\_type::const\_iterator Oscl\_TagTree< T, Alloc >::const\_iterator::mapiter
- 6.77.1.2 template<class T, class Alloc> typedef const node\_type\* Oscl\_TagTree< T, Alloc >::const\_iterator::pointer
- 6.77.1.3 template<class T, class Alloc> typedef const node\_type& Oscl\_TagTree< T, Alloc >::const\_iterator::reference
- 6.77.1.4 template<class T, class Alloc> typedef const\_iterator Oscl\_TagTree< T, Alloc >::const\_iterator::self

### 6.77.2 Constructor & Destructor Documentation

- 6.77.2.1 template<class T, class Alloc> Oscl\_TagTree< T, Alloc >::const\_iterator::const\_iterator() [inline]
- 6.77.2.2 template<class T, class Alloc> Oscl\_TagTree< T, Alloc >::const\_iterator::const\_iterator(mapiter x) [inline]
- 6.77.2.3 template<class T, class Alloc> Oscl\_TagTree< T, Alloc >::const\_iterator::const\_iterator(const const\_iterator & it) [inline]

### 6.77.3 Member Function Documentation

- 6.77.3.1 template<class T, class Alloc> reference Oscl\_TagTree< T, Alloc >::const\_iterator::operator \*() const [inline]
- 6.77.3.2 template<class T, class Alloc> bool Oscl\_TagTree< T, Alloc >::const\_iterator::operator!= (const self & x) [inline]
- 6.77.3.3 template<class T, class Alloc> self Oscl\_TagTree< T, Alloc >::const\_iterator::operator++(int) [inline]
- 6.77.3.4 template<class T, class Alloc> self& Oscl\_TagTree< T, Alloc >::const\_iterator::operator++() [inline]
- 6.77.3.5 template<class T, class Alloc> self Oscl\_TagTree< T, Alloc >::const\_iterator::operator-(int) [inline]
- 6.77.3.6 template<class T, class Alloc> self& Oscl\_TagTree< T, Alloc >::const\_iterator::operator-() [inline]
- 6.77.3.7 template<class T, class Alloc> pointer Oscl\_TagTree< T, Alloc >::const\_iterator::operator -() const [inline]
- 6.77.3.8 template<class T, class Alloc> bool Oscl\_TagTree< T, Alloc >::const\_iterator::operator==(const self & x) [inline]

### 6.77.4 Field Documentation

- [oscl\\_tagtree.h](#)

## 6.78 Oscl\_TagTree< T, Alloc >::iterator Struct Reference

```
#include <oscl_tagtree.h>
```

### Public Types

- `typedef node_type & reference`
- `typedef node_type * pointer`
- `typedef map_type::iterator mapiter`
- `typedef iterator self`

### Public Methods

- `iterator ()`
- `iterator (mapiter x)`
- `iterator (const iterator &it)`
- `reference operator * () const`
- `pointer operator → () const`
- `bool operator== (const self &x)`
- `bool operator!= (const self &x)`
- `self & operator++ ()`
- `self operator++ (int)`
- `self & operator– ()`
- `self operator– (int)`

### Data Fields

- `mapiter mapit`

template<class T, class Alloc> struct Oscl\_TagTree< T, Alloc >::iterator

### 6.78.1 Member Typedef Documentation

- 6.78.1.1 template<class T, class Alloc> typedef map\_type::iterator Oscl\_TagTree< T, Alloc >::iterator::mapiter
- 6.78.1.2 template<class T, class Alloc> typedef node\_type\* Oscl\_TagTree< T, Alloc >::iterator::pointer
- 6.78.1.3 template<class T, class Alloc> typedef node\_type& Oscl\_TagTree< T, Alloc >::iterator::reference
- 6.78.1.4 template<class T, class Alloc> typedef iterator Oscl\_TagTree< T, Alloc >::iterator::self

### 6.78.2 Constructor & Destructor Documentation

- 6.78.2.1 template<class T, class Alloc> Oscl\_TagTree< T, Alloc >::iterator::iterator () [inline]
- 6.78.2.2 template<class T, class Alloc> Oscl\_TagTree< T, Alloc >::iterator::iterator (mapiter x) [inline]
- 6.78.2.3 template<class T, class Alloc> Oscl\_TagTree< T, Alloc >::iterator::iterator (const iterator & it) [inline]

### 6.78.3 Member Function Documentation

- 6.78.3.1 template<class T, class Alloc> reference Oscl\_TagTree< T, Alloc >::iterator::operator \* () const [inline]
- 6.78.3.2 template<class T, class Alloc> bool Oscl\_TagTree< T, Alloc >::iterator::operator!= (const self & x) [inline]
- 6.78.3.3 template<class T, class Alloc> self Oscl\_TagTree< T, Alloc >::iterator::operator++ (int) [inline]
- 6.78.3.4 template<class T, class Alloc> self& Oscl\_TagTree< T, Alloc >::iterator::operator++ () [inline]
- 6.78.3.5 template<class T, class Alloc> self Oscl\_TagTree< T, Alloc >::iterator::operator- (int) [inline]
- 6.78.3.6 template<class T, class Alloc> self& Oscl\_TagTree< T, Alloc >::iterator::operator- () [inline]
- 6.78.3.7 template<class T, class Alloc> pointer Oscl\_TagTree< T, Alloc >::iterator::operator → () const [inline]
- 6.78.3.8 template<class T, class Alloc> bool Oscl\_TagTree< T, Alloc >::iterator::operator== (const self & x) [inline]

### 6.78.4 Field Documentation

- 
- 6.78.4.1 template<class T, class Alloc> mapiter Oscl\_TagTree< T, Alloc >::iterator::mapit

- [oscl\\_tagtree.h](#)

## 6.79 Oscl\_TagTree< T, Alloc >::Node Struct Reference

```
#include <oscl_tagtree.h>
```

### Public Types

- `typedef Oscl_Vector< Node *, Alloc > children_type`

### Public Methods

- `Node()`
- `void sort_children()`
- `tag_type::size_type depth()`

### Data Fields

- `tag_type tag`
- `T value`
- `Node * parent`
- `children_type children`

template<class T, class Alloc> struct Oscl\_TagTree< T, Alloc >::Node

### 6.79.1 Member Typedef Documentation

6.79.1.1 template<class T, class Alloc> typedef Oscl\_Vector<Node\*, Alloc> Oscl\_TagTree< T, Alloc >::Node::children\_type

### 6.79.2 Constructor & Destructor Documentation

6.79.2.1 template<class T, class Alloc> Oscl\_TagTree< T, Alloc >::Node::Node () [inline]

### 6.79.3 Member Function Documentation

6.79.3.1 template<class T, class Alloc> tag\_type::size\_type Oscl\_TagTree< T, Alloc >::Node::depth () [inline]

6.79.3.2 template<class T, class Alloc> void Oscl\_TagTree< T, Alloc >::Node::sort\_children () [inline]

### 6.79.4 Field Documentation

6.79.4.1 template<class T, class Alloc> children\_type Oscl\_TagTree< T, Alloc >::Node::children

6.79.4.2 template<class T, class Alloc> Node\* Oscl\_TagTree< T, Alloc >::Node::parent

6.79.4.3 template<class T, class Alloc> tag\_type Oscl\_TagTree< T, Alloc >::Node::tag

6.79.4.4 template<class T, class Alloc> T Oscl\_TagTree< T, Alloc >::Node::value

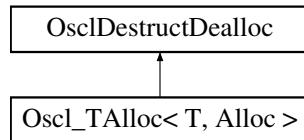
The documentation for this struct was generated from the following file:

- [oscl\\_tagtree.h](#)

## 6.80 Oscl\_TAlloc< T, Alloc > Class Template Reference

```
#include <oscl_defalloc.h>
```

Inheritance diagram for Oscl\_TAlloc< T, Alloc >::



### Public Types

- typedef T [value\\_type](#)
- typedef T \* [pointer](#)
- typedef const T \* [const\\_pointer](#)
- typedef uint32 [size\\_type](#)
- typedef T & [reference](#)
- typedef const T & [const\\_reference](#)

### Public Methods

- virtual [~Oscl\\_TAlloc \(\)](#)
- [pointer allocate\\_fl \(uint32 size, const char \\*file\\_name, const int line\\_num\)](#)
- [pointer allocate \(uint32 size\)](#)
- [pointer alloc\\_and\\_construct\\_fl \(const\\_reference val, const char \\*file\\_name, const int line\\_num\)](#)
- [pointer alloc\\_and\\_construct \(const\\_reference val\)](#)
- void [deallocate \(OsclAny \\*p\)](#)
- void [deallocate \(OsclAny \\*p, size\\_type n\)](#)
- void [destruct\\_and\\_dealloc \(OsclAny \\*p\)](#)
- [pointer address \(reference r\)](#)
- [const\\_pointer address \(const\\_reference r\) const](#)
- void [construct \(pointer p, const\\_reference val\)](#)
- void [destroy \(pointer p\)](#)

template<class T, class Alloc> class Oscl\_TAlloc< T, Alloc >

### 6.80.1 Member Typedef Documentation

6.80.1.1 template<class T, class Alloc> **typedef const T\* Oscl\_TAlloc< T, Alloc >::const\_pointer**

6.80.1.2 template<class T, class Alloc> **typedef const T& Oscl\_TAlloc< T, Alloc >::const\_reference**

6.80.1.3 template<class T, class Alloc> **typedef T\* Oscl\_TAlloc< T, Alloc >::pointer**

6.80.1.4 template<class T, class Alloc> **typedef T& Oscl\_TAlloc< T, Alloc >::reference**

6.80.1.5 template<class T, class Alloc> **typedef uint32 Oscl\_TAlloc< T, Alloc >::size\_type**

6.80.1.6 template<class T, class Alloc> **typedef T Oscl\_TAlloc< T, Alloc >::value\_type**

### 6.80.2 Constructor & Destructor Documentation

6.80.2.1 template<class T, class Alloc> **virtual Oscl\_TAlloc< T, Alloc >::~Oscl\_TAlloc ()**  
[inline, virtual]

### 6.80.3 Member Function Documentation

6.80.3.1 template<class T, class Alloc> **const\_pointer Oscl\_TAlloc< T, Alloc >::address**  
(**const\_reference r**) const [inline]

6.80.3.2 template<class T, class Alloc> **pointer Oscl\_TAlloc< T, Alloc >::address** (**reference r**)  
[inline]

6.80.3.3 template<class T, class Alloc> **pointer Oscl\_TAlloc< T, Alloc >::alloc\_and\_construct**  
(**const\_reference val**) [inline]

6.80.3.4 template<class T, class Alloc> **pointer Oscl\_TAlloc< T, Alloc >::alloc\_and\_construct\_fl**  
(**const\_reference val, const char \*file\_name, const int line\_num**) [inline]

6.80.3.5 template<class T, class Alloc> **pointer Oscl\_TAlloc< T, Alloc >::allocate** (uint32 **size**)  
[inline]

6.80.3.6 template<class T, class Alloc> **pointer Oscl\_TAlloc< T, Alloc >::allocate\_fl** (uint32 **size**,  
const char \***file\_name**, const int **line\_num**) [inline]

6.80.3.7 template<class T, class Alloc> **void Oscl\_TAlloc< T, Alloc >::construct** (**pointer p**,  
**const\_reference val**) [inline]

6.80.3.8 template<class T, class Alloc> **void Oscl\_TAlloc< T, Alloc >::deallocate** (**OsclAny \*p**,  
**size\_type n**) [inline]

6.80.3.9 template<class T, class Alloc> **void Oscl\_TAlloc< T, Alloc >::deallocate** (**OsclAny \*p**)  
[inline]

6.80.3.10 template<class T, class Alloc> **void Oscl\_TAlloc< T, Alloc >::destroy** (**pointer p**)  
[inline]

6.80.3.11 template<class T, class Alloc> **void Oscl\_TAlloc< T, Alloc >::destruct\_and\_dealloc**  
(**OsclAny \*p**) [inline, virtual]

The documentation for this class was generated from the following file:

- [oscl\\_defalloc.h](#)

## 6.81 Oscl\_TAlloc< T, Alloc >::rebind< U, V > Struct Template Reference

```
#include <oscl_defalloc.h>
```

### Public Types

- `typedef Oscl_TAlloc< U, V > other`

```
template<class T, class Alloc>template<class U, class V> struct Oscl_TAlloc< T, Alloc >::rebind< U, V >
```

#### 6.81.1 Member Typedef Documentation

```
6.81.1.1 template<class T, class Alloc> template<class U, class V> typedef Oscl_TAlloc<U, V>  
Oscl_TAlloc< T, Alloc >::rebind< U, V >::other
```

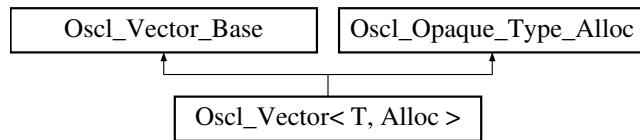
The documentation for this struct was generated from the following file:

- `oscl_defalloc.h`

## 6.82 Oscl\_Vector< T, Alloc > Class Template Reference

```
#include <oscl_vector.h>
```

Inheritance diagram for Oscl\_Vector< T, Alloc >::



### Public Types

- `typedef T value_type`
- `typedef T * pointer`
- `typedef T & reference`
- `typedef const T & const_reference`
- `typedef T * iterator`
- `typedef const T * const_iterator`

### Public Methods

- `Oscl_Vector()`
- `Oscl_Vector(uint32 n)`
- `Oscl_Vector(const Oscl_Vector< T, Alloc > &x)`
- `virtual ~Oscl_Vector()`
- `Oscl_Vector< T, Alloc > & operator=(const Oscl_Vector< T, Alloc > &x)`
- `void push_back(const T &x)`
- `void push_front(const T &x)`
- `iterator insert(iterator pos, const T &x)`
- `T & operator[](uint32 n)`
- `const T & operator[](uint32 n) const`
- `T & front()`
- `const T & front() const`
- `T & back()`
- `const T & back() const`
- `void pop_back()`
- `void clear()`
- `void destroy()`
- `iterator begin() const`
- `iterator end() const`
- `iterator erase(iterator pos)`
- `iterator erase(iterator first, iterator last)`

### 6.82.1 Detailed Description

**template<class T, class Alloc> class Oscl\_Vector< T, Alloc >**

Oscl\_Vector Class. A subset of STL::Vector methods. Oscl\_Vector supports random access to elements, constant time insertion and removal of elements at the end of the vector, and linear time insertion and removal of elements at the beginning or middle of the vector. The number of elements in a vector can vary dynamically, and memory management is performed automatically.

### 6.82.2 Member Typedef Documentation

**6.82.2.1 template<class T, class Alloc> typedef const T\* Oscl\_Vector< T, Alloc >::const\_iterator**

**6.82.2.2 template<class T, class Alloc> typedef const T& Oscl\_Vector< T, Alloc >::const\_reference**

**6.82.2.3 template<class T, class Alloc> typedef T\* Oscl\_Vector< T, Alloc >::iterator**

**6.82.2.4 template<class T, class Alloc> typedef T\* Oscl\_Vector< T, Alloc >::pointer**

**6.82.2.5 template<class T, class Alloc> typedef T& Oscl\_Vector< T, Alloc >::reference**

**6.82.2.6 template<class T, class Alloc> typedef T Oscl\_Vector< T, Alloc >::value\_type**

### 6.82.3 Constructor & Destructor Documentation

**6.82.3.1 template<class T, class Alloc> Oscl\_Vector< T, Alloc >::Oscl\_Vector () [inline]**

Creates an empty vector.

**6.82.3.2 template<class T, class Alloc> Oscl\_Vector< T, Alloc >::Oscl\_Vector (uint32 n) [inline]**

Creates an empty vector with capacity n.

**Parameters:**

*n* creates a vector with n elements. The main reason for specifying n is efficiency. If you know the capacity to which your vector must grow, then it is more efficient to allocate the vector all at once rather than rely on the automatic reallocation scheme. This also helps control the invalidation of iterators.

**6.82.3.3 template<class T, class Alloc> Oscl\_Vector< T, Alloc >::Oscl\_Vector (const Oscl\_Vector< T, Alloc > & x) [inline]**

Copy Constructor.

**Parameters:**

*x* vector class to copy.

**6.82.3.4 template<class T, class Alloc> virtual Oscl\_Vector< T, Alloc >::~Oscl\_Vector ()  
[inline, virtual]**

The destructor.

#### 6.82.4 Member Function Documentation

**6.82.4.1 template<class T, class Alloc> const T& Oscl\_Vector< T, Alloc >::back () const  
[inline]**

Returns the last element.

**6.82.4.2 template<class T, class Alloc> T& Oscl\_Vector< T, Alloc >::back () [inline]**

Returns the last element.

**6.82.4.3 template<class T, class Alloc> iterator Oscl\_Vector< T, Alloc >::begin () const  
[inline]**

Returns an iterator pointing to the beginning of the vector.

Reimplemented from [Oscl\\_Vector\\_Base](#).

**6.82.4.4 template<class T, class Alloc> void Oscl\_Vector< T, Alloc >::clear () [inline]**

Removes all elements.

**6.82.4.5 template<class T, class Alloc> void Oscl\_Vector< T, Alloc >::destroy () [inline]**

Destroy – this is like an explicit destructor call.

Reimplemented from [Oscl\\_Vector\\_Base](#).

**6.82.4.6 template<class T, class Alloc> iterator Oscl\_Vector< T, Alloc >::end () const  
[inline]**

Returns an iterator pointing to the end of the vector..

Reimplemented from [Oscl\\_Vector\\_Base](#).

**6.82.4.7 template<class T, class Alloc> iterator Oscl\_Vector< T, Alloc >::erase (iterator *first*,  
iterator *last*) [inline]**

Erases elements in range [*first*, *last*). Erasing an element invalidates all iterators pointing to elements following the deletion point.

##### Parameters:

*first* starting position

*last* ending position, this position is not erased

**6.82.4.8 template<class T, class Alloc> iterator Oscl\_Vector< T, Alloc >::erase (iterator pos) [inline]**

Erases the element pointed to by iterator pos. Erasing an element invalidates all iterators pointing to elements following the deletion point.

**Parameters:**

*pos* iterator at erase position

**6.82.4.9 template<class T, class Alloc> const T& Oscl\_Vector< T, Alloc >::front () const [inline]**

Returns the first element.

**6.82.4.10 template<class T, class Alloc> T& Oscl\_Vector< T, Alloc >::front () [inline]**

Returns the first element.

**6.82.4.11 template<class T, class Alloc> iterator Oscl\_Vector< T, Alloc >::insert (iterator pos, const T & x) [inline]**

Inserts a new element before the one at pos.

**Parameters:**

*pos* position at which to insert the new element.

*x* new element

**6.82.4.12 template<class T, class Alloc> Oscl\_Vector<T, Alloc>& Oscl\_Vector< T, Alloc >::operator= (const Oscl\_Vector< T, Alloc > & x) [inline]**

The assignment operator

**6.82.4.13 ]**

template<class T, class Alloc> const T& Oscl\_Vector< T, Alloc >::operator[] (uint32 n) const [inline]

Returns the n'th element.

**Parameters:**

*n* element position to return

**6.82.4.14 ]**

template<class T, class Alloc> T& Oscl\_Vector< T, Alloc >::operator[] (uint32 n) [inline]

Returns the n'th element.

**Parameters:**

*n* element position to return

**6.82.4.15 template<class T, class Alloc> void Oscl\_Vector< T, Alloc >::pop\_back () [inline]**

Removes the last element.

Reimplemented from [Oscl\\_Vector\\_Base](#).

**6.82.4.16 template<class T, class Alloc> void Oscl\_Vector< T, Alloc >::push\_back (const T & x) [inline]**

Inserts a new element at the end. Inserting an element invalidates all iterators if memory reallocation occurs as a result of the insertion.

**Parameters:**

*x* new element

**6.82.4.17 template<class T, class Alloc> void Oscl\_Vector< T, Alloc >::push\_front (const T & x) [inline]**

Inserts a new element at the front. Inserting an element invalidates all iterators if memory reallocation occurs as a result of the insertion.

**Parameters:**

*x* new element

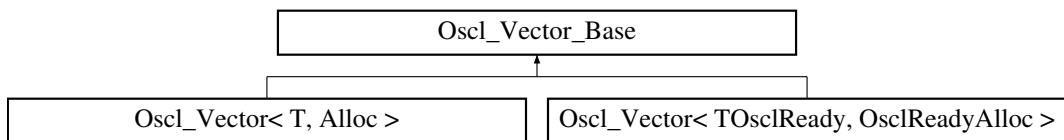
The documentation for this class was generated from the following file:

- [oscl\\_vector.h](#)

## 6.83 Oscl\_Vector\_Base Class Reference

```
#include <oscl_vector.h>
```

Inheritance diagram for Oscl\_Vector\_Base::



### Public Methods

- uint32 [size \(\) const](#)
- uint32 [capacity \(\) const](#)
- bool [empty \(\) const](#)
- OSCL\_IMPORT\_REF void [reserve \(uint32 n\)](#)

### Protected Methods

- OSCL\_IMPORT\_REF void [construct \(Oscl\\_Opaque\\_Type\\_Alloc \\*aType\)](#)
- OSCL\_IMPORT\_REF void [construct \(Oscl\\_Opaque\\_Type\\_Alloc \\*aType, uint32 n\)](#)
- OSCL\_IMPORT\_REF void [construct \(Oscl\\_Opaque\\_Type\\_Alloc \\*aType, const Oscl\\_Vector\\_Base &x\)](#)
- virtual [~Oscl\\_Vector\\_Base \(\)](#)
- OSCL\_IMPORT\_REF void [push\\_back \(const OsclAny \\*x\)](#)
- OSCL\_IMPORT\_REF void [pop\\_back \(\)](#)
- OSCL\_IMPORT\_REF void [push\\_front \(const OsclAny \\*x\)](#)
- OSCL\_IMPORT\_REF OsclAny \* [insert \(OsclAny \\*pos, const OsclAny \\*x\)](#)
- OSCL\_IMPORT\_REF OsclAny \* [erase \(OsclAny \\*pos\)](#)
- OSCL\_IMPORT\_REF OsclAny \* [erase \(OsclAny \\*first, OsclAny \\*last\)](#)
- OSCL\_IMPORT\_REF void [assign\\_vector \(const Oscl\\_Vector\\_Base &x\)](#)
- OSCL\_IMPORT\_REF void [destroy \(\)](#)

### Protected Attributes

- uint32 [numelems](#)
- uint32 [bufsize](#)
- [OsclAny \\* elems](#)
- uint32 [sizeof\\_T](#)

### Friends

- class [OsclPriorityQueueBase](#)

### 6.83.1 Detailed Description

Oscl\_Vector\_Base is a non-templatized base class for [Oscl\\_Vector](#). The purpose of this base class is to avoid large inline routines in the [Oscl\\_Vector](#) implementation. This class is not intended for direct instantiation except by [Oscl\\_Vector](#).

### 6.83.2 Constructor & Destructor Documentation

**6.83.2.1 virtual Oscl\_Vector\_Base::~Oscl\_Vector\_Base () [inline, protected, virtual]**

The destructor.

### 6.83.3 Member Function Documentation

**6.83.3.1 OSCL\_IMPORT\_REF void Oscl\_Vector\_Base::assign\_vector (const Oscl\_Vector\_Base & x) [protected]**

**6.83.3.2 uint32 Oscl\_Vector\_Base::capacity () const [inline]**

Returns the allocated memory of the vector in units of number of elements.

**6.83.3.3 OSCL\_IMPORT\_REF void Oscl\_Vector\_Base::construct (Oscl\_Opaque\_Type\_Alloc \* aType, const Oscl\_Vector\_Base & x) [protected]**

**6.83.3.4 OSCL\_IMPORT\_REF void Oscl\_Vector\_Base::construct (Oscl\_Opaque\_Type\_Alloc \* aType, uint32 n) [protected]**

**6.83.3.5 OSCL\_IMPORT\_REF void Oscl\_Vector\_Base::construct (Oscl\_Opaque\_Type\_Alloc \* aType) [protected]**

**6.83.3.6 OSCL\_IMPORT\_REF void Oscl\_Vector\_Base::destroy () [protected]**

Reimplemented in [Oscl\\_Vector< T, Alloc >](#), [Oscl\\_Vector< OsclComponentRegistryElement, OsclMemAllocator >](#), [Oscl\\_Vector< uint32, OsclMemAllocator >](#), [Oscl\\_Vector< OsclSocketServRequestQElem, OsclMemAllocator >](#), [Oscl\\_Vector< Node \\*, Alloc >](#), [Oscl\\_Vector< OsclFixedCacheParam, OsclMemAllocator >](#), [Oscl\\_Vector< OsclSocketRequest \\*, OsclMemAllocator >](#), [Oscl\\_Vector< entry\\_type \\*, Alloc >](#), [Oscl\\_Vector< OSCL\\_HeapString< OsclMemAllocator >, OsclMemAllocator >](#), [Oscl\\_Vector< OsclAsyncFileBuffer \\*, OsclMemAllocator >](#), [Oscl\\_Vector< TOsclFileOffset, OsclMemAllocator >](#), [Oscl\\_Vector< MemPoolBufferInfo \\*, OsclMemAllocator >](#), [Oscl\\_Vector< OsclSharedPtr< PVLoggerFilter >, alloc\\_type >](#), [Oscl\\_Vector< TOsclReady, OsclReadyAlloc >](#), [Oscl\\_Vector< OsclFileCacheBuffer, OsclMemAllocator >](#), [Oscl\\_Vector< OsclSharedPtr< PVLoggerAppender >, alloc\\_type >](#), [Oscl\\_Vector< OsclNetworkAddress, OsclMemAllocator >](#), and [Oscl\\_Vector< OsclAny \\*, OsclMemAllocator >](#).

**6.83.3.7 bool Oscl\_Vector\_Base::empty () const [inline]**

True if the vector's size is 0.

**6.83.3.8 OSCL\_IMPORT\_REF `OsclAny*` Oscl\_Vector\_Base::erase (`OsclAny *first, OsclAny *last`) [protected]**

Erases elements in range [first, last). Erasing an element invalidates all iterators pointing to elements following the deletion point.

**Parameters:**

*first* starting position

*last* ending position, this position is not erased

**6.83.3.9 OSCL\_IMPORT\_REF `OsclAny*` Oscl\_Vector\_Base::erase (`OsclAny *pos`) [protected]**

Erases the element pointed to by iterator pos. Erasing an element invalidates all iterators pointing to elements following the deletion point.

**Parameters:**

*pos* iterator at erase position

**6.83.3.10 OSCL\_IMPORT\_REF `OsclAny*` Oscl\_Vector\_Base::insert (`OsclAny *pos, const OsclAny *x`) [protected]**

Inserts a new element at a specific position.

**Parameters:**

*pos* iterator at insert position.

*x* pointer to new element

**6.83.3.11 OSCL\_IMPORT\_REF void Oscl\_Vector\_Base::pop\_back () [protected]**

Removes the last element.

Reimplemented in `Oscl_Vector< T, Alloc >`, `Oscl_Vector< OsclComponentRegistryElement, OsclMemAllocator >`, `Oscl_Vector< uint32, OsclMemAllocator >`, `Oscl_Vector< OsclSocketServRequestQELEM, OsclMemAllocator >`, `Oscl_Vector< Node *, Alloc >`, `Oscl_Vector< OsclFixedCacheParam, OsclMemAllocator >`, `Oscl_Vector< OsclSocketRequest *, OsclMemAllocator >`, `Oscl_Vector< entry_type *, Alloc >`, `Oscl_Vector< OSCL_HeapString< OsclMemAllocator >, OsclMemAllocator >`, `Oscl_Vector< OsclAsyncFileBuffer *, OsclMemAllocator >`, `Oscl_Vector< TOscIFileOffset, OsclMemAllocator >`, `Oscl_Vector< MemPoolBufferInfo *, OsclMemAllocator >`, `Oscl_Vector< OsclSharedPtr< PVLoggerFilter >, alloc_type >`, `Oscl_Vector< TOscIReady, OsclReadyAlloc >`, `Oscl_Vector< OsclFileCacheBuffer, OsclMemAllocator >`, `Oscl_Vector< OsclSharedPtr< PVLoggerAppender >, alloc_type >`, `Oscl_Vector< OsclNetworkAddress, OsclMemAllocator >`, and `Oscl_Vector< OsclAny *, OsclMemAllocator >`.

**6.83.3.12 OSCL\_IMPORT\_REF void Oscl\_Vector\_Base::push\_back (const `OsclAny *x`) [protected]**

Inserts a new element at the end. Inserting an element invalidates all iterators if memory reallocation occurs as a result of the insertion.

**Parameters:**

*x* pointer to the new element

**6.83.3.13 OSCL\_IMPORT\_REF void Oscl\_Vector\_Base::push\_front (const OsclAny \* *x*)  
[protected]**

Inserts a new element at the front. Inserting an element invalidates all iterators if memory reallocation occurs as a result of the insertion.

**Parameters:**

*x* pointer to new element

**6.83.3.14 OSCL\_IMPORT\_REF void Oscl\_Vector\_Base::reserve (uint32 *n*)**

Reallocates memory if necessary to a capacity of *n* elements. The main reason for reserve is efficiency. If you know the capacity to which your vector must grow, then it is more efficient to allocate the vector all at once rather than rely on the automatic reallocation scheme. This also helps control the invalidation of iterators.

**Parameters:**

*n* size of vector

**6.83.3.15 uint32 Oscl\_Vector\_Base::size () const [inline]**

Returns the size of the vector in units of number of elements.

## 6.83.4 Friends And Related Function Documentation

**6.83.4.1 friend class OsclPriorityQueueBase [friend]**

## 6.83.5 Field Documentation

**6.83.5.1 uint32 Oscl\_Vector\_Base::bufsize [protected]****6.83.5.2 OsclAny\* Oscl\_Vector\_Base::elems [protected]****6.83.5.3 uint32 Oscl\_Vector\_Base::numelems [protected]****6.83.5.4 uint32 Oscl\_Vector\_Base::sizeof\_T [protected]**

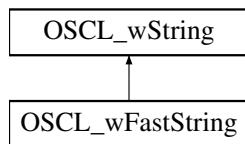
The documentation for this class was generated from the following file:

- [oscl\\_vector.h](#)

## 6.84 OSCL\_wFastString Class Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL\_wFastString::



### Public Types

- `typedef OSCL_wString::chartype chartype`

### Public Methods

- `OSCL_IMPORT_REF OSCL_wFastString()`
- `OSCL_IMPORT_REF OSCL_wFastString(const OSCL_wFastString &src)`
- `OSCL_IMPORT_REF OSCL_wFastString(const chartype *cstr)`
- `OSCL_IMPORT_REF OSCL_wFastString(chartype *buf, uint32 maxlen)`
- `OSCL_IMPORT_REF ~OSCL_wFastString()`
- `OSCL_IMPORT_REF uint32 get_size() const`
- `OSCL_IMPORT_REF uint32 get_maxsize() const`
- `OSCL_IMPORT_REF const chartype * get_cstr() const`
- `OSCL_IMPORT_REF chartype * get_str() const`
- `OSCL_IMPORT_REF OSCL_wFastString & operator=(const OSCL_wFastString &src)`
- `OSCL_IMPORT_REF OSCL_wFastString & operator=(const chartype *cstr)`
- `OSCL_IMPORT_REF void set(chartype *cstr, uint32 maxlen)`
- `OSCL_IMPORT_REF void set_length()`

### Friends

- class `OSCL_wString`

### 6.84.1 Detailed Description

OSCL\_wFastString is identical to `OSCL_FastString` except that it uses wide-character format. For descriptions, see `OSCL_FastString`.

### 6.84.2 Member Typedef Documentation

#### 6.84.2.1 `typedef OSCL_wString::chartype OSCL_wFastString::chartype`

Reimplemented from `OSCL_wString`.

### 6.84.3 Constructor & Destructor Documentation

- 6.84.3.1 **OSCL\_IMPORT\_REF OSCL\_wFastString::OSCL\_wFastString ()**
- 6.84.3.2 **OSCL\_IMPORT\_REF OSCL\_wFastString::OSCL\_wFastString (const OSCL\_wFastString & src)**
- 6.84.3.3 **OSCL\_IMPORT\_REF OSCL\_wFastString::OSCL\_wFastString (const chartype \* cstr)**
- 6.84.3.4 **OSCL\_IMPORT\_REF OSCL\_wFastString::OSCL\_wFastString (chartype \* buf, uint32 maxlen)**
- 6.84.3.5 **OSCL\_IMPORT\_REF OSCL\_wFastString::~OSCL\_wFastString ()**

### 6.84.4 Member Function Documentation

- 6.84.4.1 **OSCL\_IMPORT\_REF const chartype\* OSCL\_wFastString::get\_cstr () [virtual]**

Implements [OSCL\\_wString](#).

- 6.84.4.2 **OSCL\_IMPORT\_REF uint32 OSCL\_wFastString::get\_maxsize () [virtual]**

Implements [OSCL\\_wString](#).

- 6.84.4.3 **OSCL\_IMPORT\_REF uint32 OSCL\_wFastString::get\_size () [virtual]**

Implements [OSCL\\_wString](#).

- 6.84.4.4 **OSCL\_IMPORT\_REF chartype\* OSCL\_wFastString::get\_str () [virtual]**

Implements [OSCL\\_wString](#).

- 6.84.4.5 **OSCL\_IMPORT\_REF OSCL\_wFastString& OSCL\_wFastString::operator= (const chartype \* cstr)**

Reimplemented from [OSCL\\_wString](#).

- 6.84.4.6 **OSCL\_IMPORT\_REF OSCL\_wFastString& OSCL\_wFastString::operator= (const OSCL\_wFastString & src)**

- 6.84.4.7 **OSCL\_IMPORT\_REF void OSCL\_wFastString::set (chartype \* cstr, uint32 maxlen)**

- 6.84.4.8 **OSCL\_IMPORT\_REF void OSCL\_wFastString::set\_length ()**

### 6.84.5 Friends And Related Function Documentation

- 6.84.5.1 **friend class OSCL\_wString [friend]**

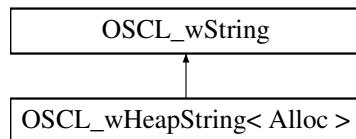
The documentation for this class was generated from the following file:

- [oscl\\_string\\_containers.h](#)

## 6.85 OSCL\_wHeapString< Alloc > Class Template Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL\_wHeapString< Alloc >::



### Public Types

- `typedef OSCL_wString::chartype chartype`

### Public Methods

- `OSCL_wHeapString()`
- `OSCL_wHeapString(const OSCL_wHeapString &src)`
- `OSCL_wHeapString(const OSCL_wString &src)`
- `OSCL_wHeapString(const chartype *cstr)`
- `OSCL_wHeapString(const chartype *buf, uint32 length)`
- `~OSCL_wHeapString()`
- `uint32 get_size() const`
- `uint32 get_maxsize() const`
- `const chartype * get_cstr() const`
- `chartype * get_str() const`
- `OSCL_wHeapString & operator=(const OSCL_wHeapString &src)`
- `OSCL_wHeapString & operator=(const OSCL_wString &src)`
- `OSCL_wHeapString & operator=(const chartype *cstr)`
- `void set(const chartype *buf, uint32 length)`

### Friends

- class `OSCL_wString`

#### 6.85.1 Detailed Description

```
template<class Alloc> class OSCL_wHeapString< Alloc >
```

`OSCL_wHeapString` is identical to `OSCL_HeapString` except that it uses wide-character format. For descriptions, see `OSCL_HeapString`.

#### 6.85.2 Member Typedef Documentation

##### 6.85.2.1 template<class Alloc> typedef OSCL\_wString::chartype OSCL\_wHeapString< Alloc >::chartype

Reimplemented from `OSCL_wString`.

### 6.85.3 Friends And Related Function Documentation

#### 6.85.3.1 template<class Alloc> friend class OSCL\_wString [friend]

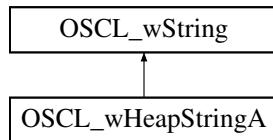
The documentation for this class was generated from the following file:

- [oscl\\_string\\_containers.h](#)

## 6.86 OSCL\_wHeapStringA Class Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL\_wHeapStringA::



### Public Types

- `typedef OSCL_wString::chartype chartype`

### Public Methods

- `OSCL_IMPORT_REF OSCL_wHeapStringA()`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(Oscl_DefAlloc *alloc, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(const OSCL_wHeapStringA &src)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(const OSCL_wHeapStringA &src, Oscl_DefAlloc *alloc, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(const OSCL_wString &src, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(const chartype *cstr, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(const chartype *buf, uint32 length, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF ~OSCL_wHeapStringA()`
- `OSCL_IMPORT_REF uint32 get_size() const`
- `OSCL_IMPORT_REF uint32 get_maxsize() const`
- `OSCL_IMPORT_REF const chartype * get_cstr() const`
- `OSCL_IMPORT_REF chartype * get_str() const`
- `OSCL_IMPORT_REF OSCL_wHeapStringA & operator=(const OSCL_wHeapStringA &src)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA & operator=(const OSCL_wString &src)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA & operator=(const chartype *cstr)`
- `OSCL_IMPORT_REF void set(const chartype *buf, uint32 length)`

### Friends

- class `OSCL_wString`

#### 6.86.1 Detailed Description

OSCL\_wHeapStringA is identical to `OSCL_HeapStringA` except that it uses wide-character format. For descriptions, see `OSCL_HeapStringA`.

## 6.86.2 Member Typedef Documentation

### 6.86.2.1 `typedef OSCL_wString::chartype OSCL_wHeapStringA::chartype`

Reimplemented from [OSCL\\_wString](#).

## 6.86.3 Constructor & Destructor Documentation

### 6.86.3.1 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA()`

### 6.86.3.2 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA (Oscl\_DefAlloc * alloc, OsclRefCounter *ref = NULL)`

### 6.86.3.3 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA (const OSCL_wHeapStringA &src)`

### 6.86.3.4 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA (const OSCL_wHeapStringA &src, Oscl\_DefAlloc *alloc, OsclRefCounter *ref = NULL)`

### 6.86.3.5 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA (const OSCL\_wString &src, Oscl\_DefAlloc *alloc = NULL, OsclRefCounter *ref = NULL)`

### 6.86.3.6 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA (const chartype * cstr, Oscl\_DefAlloc *alloc = NULL, OsclRefCounter *ref = NULL)`

### 6.86.3.7 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA (const chartype * buf, uint32 length, Oscl\_DefAlloc *alloc = NULL, OsclRefCounter *ref = NULL)`

### 6.86.3.8 `OSCL_IMPORT_REF OSCL_wHeapStringA::~OSCL_wHeapStringA()`

## 6.86.4 Member Function Documentation

### 6.86.4.1 `OSCL_IMPORT_REF const chartype* OSCL_wHeapStringA::get_cstr () [virtual]`

Implements [OSCL\\_wString](#).

### 6.86.4.2 `OSCL_IMPORT_REF uint32 OSCL_wHeapStringA::get_maxsize () [virtual]`

Implements [OSCL\\_wString](#).

### 6.86.4.3 `OSCL_IMPORT_REF uint32 OSCL_wHeapStringA::get_size () [virtual]`

Implements [OSCL\\_wString](#).

### 6.86.4.4 `OSCL_IMPORT_REF chartype* OSCL_wHeapStringA::get_str () [virtual]`

Implements [OSCL\\_wString](#).

**6.86.4.5 OSCL\_IMPORT\_REF OSCL\_wHeapStringA& OSCL\_wHeapStringA::operator= (const chartype \* *cstr*)**

Reimplemented from [OSCL\\_wString](#).

**6.86.4.6 OSCL\_IMPORT\_REF OSCL\_wHeapStringA& OSCL\_wHeapStringA::operator= (const OSCL\_wString & *src*)**

Reimplemented from [OSCL\\_wString](#).

**6.86.4.7 OSCL\_IMPORT\_REF OSCL\_wHeapStringA& OSCL\_wHeapStringA::operator= (const OSCL\_wHeapStringA & *src*)**

**6.86.4.8 OSCL\_IMPORT\_REF void OSCL\_wHeapStringA::set (const chartype \* *buf*, uint32 *length*)**

## 6.86.5 Friends And Related Function Documentation

**6.86.5.1 friend class OSCL\_wString [friend]**

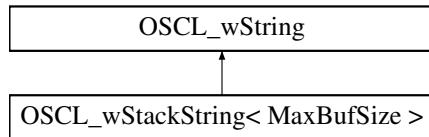
The documentation for this class was generated from the following file:

- [oscl\\_string\\_containers.h](#)

## 6.87 OSCL\_wStackString< MaxBufSize > Class Template Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL\_wStackString< MaxBufSize >::



### Public Types

- `typedef OSCL_wString::chartype chartype`

### Public Methods

- `OSCL_wStackString()`
- `OSCL_wStackString(const OSCL_wStackString &src)`
- `OSCL_wStackString(const OSCL_wString &src)`
- `OSCL_wStackString(const chartype *cstr)`
- `OSCL_wStackString(const chartype *buf, uint32 length)`
- `~OSCL_wStackString()`
- `uint32 get_size() const`
- `uint32 get_maxsize() const`
- `const chartype * get_cstr() const`
- `chartype * get_str() const`
- `OSCL_wStackString & operator=(const OSCL_wStackString &src)`
- `OSCL_wStackString & operator=(const OSCL_wString &src)`
- `OSCL_wStackString & operator=(const chartype *cstr)`
- `void set(const chartype *buf, uint32 length)`

### Friends

- class `OSCL_wString`

#### 6.87.1 Detailed Description

`template<uint32 MaxBufSize> class OSCL_wStackString< MaxBufSize >`

`OSCL_wStackString` is identical to `OSCL_StackString` except that it uses wide-character format. For descriptions, see `OSCL_StackString`.

## 6.87.2 Member Typedef Documentation

6.87.2.1 `template<uint32 MaxBufSize> typedef OSCL_wString::chartype OSCL_wStackString<MaxBufSize >::chartype`

Reimplemented from [OSCL\\_wString](#).

## 6.87.3 Friends And Related Function Documentation

6.87.3.1 `template<uint32 MaxBufSize> friend class OSCL_wString [friend]`

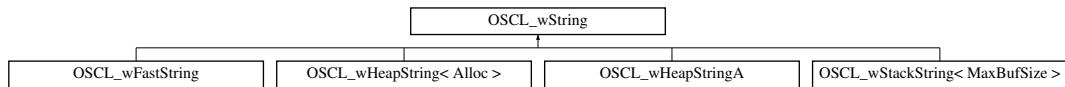
The documentation for this class was generated from the following file:

- [oscl\\_string\\_containers.h](#)

## 6.88 OSCL\_wString Class Reference

```
#include <oscl_string.h>
```

Inheritance diagram for OSCL\_wString::



### Public Types

- `typedef oscl_wchar chartype`

### Public Methods

- `virtual uint32 get_size () const=0`
- `virtual uint32 get_maxsize () const=0`
- `virtual const chartype * get_cstr () const=0`
- `virtual OSCL_IMPORT_REF bool is_writable () const`
- `virtual chartype * get_str () const=0`
- `OSCL_IMPORT_REF OSCL_wString & operator=(const OSCL_wString &src)`
- `OSCL_IMPORT_REF OSCL_wString & operator=(const chartype *cstr)`
- `OSCL_IMPORT_REF OSCL_wString & operator+=(const OSCL_wString &src)`
- `OSCL_IMPORT_REF OSCL_wString & operator+=(const chartype *cstr)`
- `OSCL_IMPORT_REF OSCL_wString & operator+=(const chartype c)`
- `OSCL_IMPORT_REF bool operator==(const OSCL_wString &src) const`
- `OSCL_IMPORT_REF bool operator!=(const OSCL_wString &src) const`
- `OSCL_IMPORT_REF bool operator< (const OSCL_wString &src) const`
- `OSCL_IMPORT_REF bool operator<= (const OSCL_wString &src) const`
- `OSCL_IMPORT_REF bool operator> (const OSCL_wString &src) const`
- `OSCL_IMPORT_REF bool operator>= (const OSCL_wString &src) const`
- `OSCL_IMPORT_REF bool operator==(const chartype *cstr) const`
- `OSCL_IMPORT_REF chartype operator[] (uint32 index) const`
- `virtual OSCL_IMPORT_REF chartype read (uint32 index) const`
- `virtual OSCL_IMPORT_REF int8 hash () const`
- `virtual OSCL_IMPORT_REF void write (uint32 index, chartype c)`
- `virtual OSCL_IMPORT_REF void write (uint32 offset, uint32 length, const chartype *buf)`

### Protected Methods

- `OSCL_IMPORT_REF OSCL_wString ()`
- `virtual OSCL_IMPORT_REF ~OSCL_wString ()`
- `virtual void set_rep (const chartype *cstr)=0`
- `virtual void append_rep (const chartype *cstr)=0`
- `virtual void set_rep (const OSCL_wString &src)=0`
- `virtual void append_rep (const OSCL_wString &src)=0`
- `virtual void set_len (uint32 len)=0`

### 6.88.1 Detailed Description

A common base class for string classes with wide character (oscl\_wchar) format. OSCL\_wString and [OSCL\\_String](#) are identical except for the character format. For descriptions, see [OSCL\\_String](#).

### 6.88.2 Member Typedef Documentation

#### 6.88.2.1 `typedef oscl_wchar OSCL_wString::chartype`

Reimplemented in [OSCL\\_wHeapString< Alloc >](#), [OSCL\\_wHeapStringA](#), [OSCL\\_wStackString< MaxBufSize >](#), and [OSCL\\_wFastString](#).

### 6.88.3 Constructor & Destructor Documentation

#### 6.88.3.1 `OSCL_IMPORT_REF OSCL_wString::OSCL_wString () [protected]`

#### 6.88.3.2 `virtual OSCL_IMPORT_REF OSCL_wString::~OSCL_wString () [protected, virtual]`

### 6.88.4 Member Function Documentation

#### 6.88.4.1 `virtual void OSCL_wString::append_rep (const OSCL_wString & src) [protected, pure virtual]`

#### 6.88.4.2 `virtual void OSCL_wString::append_rep (const chartype * cstr) [protected, pure virtual]`

#### 6.88.4.3 `virtual const chartype* OSCL_wString::get_cstr () [pure virtual]`

Implemented in [OSCL\\_wHeapString< Alloc >](#), [OSCL\\_wHeapStringA](#), [OSCL\\_wStackString< MaxBufSize >](#), and [OSCL\\_wFastString](#).

#### 6.88.4.4 `virtual uint32 OSCL_wString::get_maxsize () [pure virtual]`

Implemented in [OSCL\\_wHeapString< Alloc >](#), [OSCL\\_wHeapStringA](#), [OSCL\\_wStackString< MaxBufSize >](#), and [OSCL\\_wFastString](#).

#### 6.88.4.5 `virtual uint32 OSCL_wString::get_size () [pure virtual]`

Implemented in [OSCL\\_wHeapString< Alloc >](#), [OSCL\\_wHeapStringA](#), [OSCL\\_wStackString< MaxBufSize >](#), and [OSCL\\_wFastString](#).

#### 6.88.4.6 `virtual chartype* OSCL_wString::get_str () [pure virtual]`

Implemented in [OSCL\\_wHeapString< Alloc >](#), [OSCL\\_wHeapStringA](#), [OSCL\\_wStackString< MaxBufSize >](#), and [OSCL\\_wFastString](#).

- 6.88.4.7 **virtual OSCL\_IMPORT\_REF int8 OSCL\_wString::hash () [virtual]**
- 6.88.4.8 **virtual OSCL\_IMPORT\_REF bool OSCL\_wString::is\_writable () [virtual]**
- 6.88.4.9 **OSCL\_IMPORT\_REF bool OSCL\_wString::operator!= (const OSCL\_wString & src) const**
- 6.88.4.10 **OSCL\_IMPORT\_REF OSCL\_wString& OSCL\_wString::operator+= (const chartype c)**
- 6.88.4.11 **OSCL\_IMPORT\_REF OSCL\_wString& OSCL\_wString::operator+= (const chartype \* cstr)**
- 6.88.4.12 **OSCL\_IMPORT\_REF OSCL\_wString& OSCL\_wString::operator+= (const OSCL\_wString & src)**
- 6.88.4.13 **OSCL\_IMPORT\_REF bool OSCL\_wString::operator< (const OSCL\_wString & src) const**
- 6.88.4.14 **OSCL\_IMPORT\_REF bool OSCL\_wString::operator<= (const OSCL\_wString & src) const**
- 6.88.4.15 **OSCL\_IMPORT\_REF OSCL\_wString& OSCL\_wString::operator= (const chartype \* cstr)**

Reimplemented in [OSCL\\_wHeapString< Alloc >](#), [OSCL\\_wHeapStringA](#), [OSCL\\_wStackString< MaxBufSize >](#), and [OSCL\\_wFastString](#).

- 6.88.4.16 **OSCL\_IMPORT\_REF OSCL\_wString& OSCL\_wString::operator= (const OSCL\_wString & src)**

Reimplemented in [OSCL\\_wHeapString< Alloc >](#), [OSCL\\_wHeapStringA](#), and [OSCL\\_wStackString< MaxBufSize >](#).

- 6.88.4.17 **OSCL\_IMPORT\_REF bool OSCL\_wString::operator== (const chartype \* cstr) const**
- 6.88.4.18 **OSCL\_IMPORT\_REF bool OSCL\_wString::operator== (const OSCL\_wString & src) const**
- 6.88.4.19 **OSCL\_IMPORT\_REF bool OSCL\_wString::operator> (const OSCL\_wString & src) const**
- 6.88.4.20 **OSCL\_IMPORT\_REF bool OSCL\_wString::operator>= (const OSCL\_wString & src) const**
- 6.88.4.21 **]**

**OSCL\_IMPORT\_REF chartype OSCL\_wString::operator[ ] (uint32 index) const**

- 6.88.4.22 **virtual OSCL\_IMPORT\_REF chartype OSCL\_wString::read (uint32 *index*) const [virtual]**
- 6.88.4.23 **virtual void OSCL\_wString::set\_len (uint32 *len*) [protected, pure virtual]**
- 6.88.4.24 **virtual void OSCL\_wString::set\_rep (const OSCL\_wString & *src*) [protected, pure virtual]**
- 6.88.4.25 **virtual void OSCL\_wString::set\_rep (const chartype \* *cstr*) [protected, pure virtual]**
- 6.88.4.26 **virtual OSCL\_IMPORT\_REF void OSCL\_wString::write (uint32 *offset*, uint32 *length*, const chartype \* *buf*) [virtual]**
- 6.88.4.27 **virtual OSCL\_IMPORT\_REF void OSCL\_wString::write (uint32 *index*, chartype *c*) [virtual]**

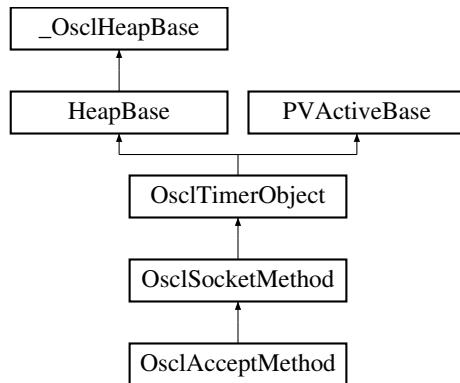
The documentation for this class was generated from the following file:

- [oscl\\_string.h](#)

## 6.89 OsclAcceptMethod Class Reference

```
#include <oscl_socket_accept.h>
```

Inheritance diagram for OsclAcceptMethod::



### Public Methods

- [~OsclAcceptMethod \(\)](#)
- [TPVSocketEvent Accept \(int32 aTimeout\)](#)
- [void DiscardAcceptedSocket \(\)](#)
- [OsclSocketI \\* GetAcceptedSocket \(\)](#)
- [OsclAcceptRequest \\* AcceptRequest \(\)](#)

### Static Public Methods

- [OsclAcceptMethod \\* NewL \(OsclIPSocketI &c\)](#)

#### 6.89.1 Constructor & Destructor Documentation

##### 6.89.1.1 OsclAcceptMethod::~OsclAcceptMethod ()

#### 6.89.2 Member Function Documentation

##### 6.89.2.1 TPVSocketEvent OsclAcceptMethod::Accept (int32 aTimeout)

##### 6.89.2.2 OsclAcceptRequest\* OsclAcceptMethod::AcceptRequest () [inline]

##### 6.89.2.3 void OsclAcceptMethod::DiscardAcceptedSocket ()

##### 6.89.2.4 OsclSocketI\* OsclAcceptMethod::GetAcceptedSocket ()

##### 6.89.2.5 OsclAcceptMethod\* OsclAcceptMethod::NewL (OsclIPSocketI &c) [static]

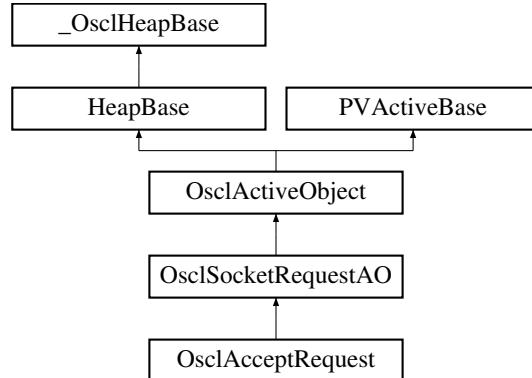
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_accept.h](#)

## 6.90 OsclAcceptRequest Class Reference

```
#include <oscl_socket_accept.h>
```

Inheritance diagram for OsclAcceptRequest::



### Public Methods

- [OsclAcceptRequest \(OsclSocketMethod &c\)](#)
- void [Accept \(OsclSocketI &aSocket\)](#)

#### 6.90.1 Constructor & Destructor Documentation

##### 6.90.1.1 OsclAcceptRequest::OsclAcceptRequest ([OsclSocketMethod & c](#)) [inline]

#### 6.90.2 Member Function Documentation

##### 6.90.2.1 void OsclAcceptRequest::Accept ([OsclSocketI & aSocket](#))

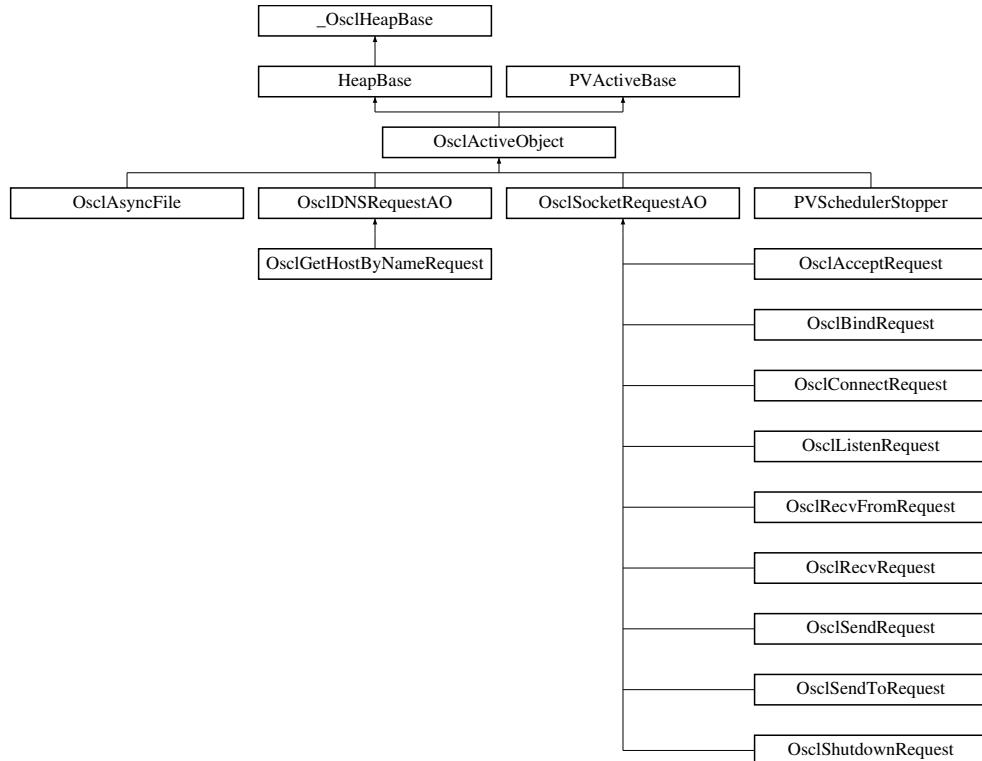
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_accept.h](#)

## 6.91 OsclActiveObject Class Reference

```
#include <oscl_scheduler_ao.h>
```

Inheritance diagram for OsclActiveObject::



### Public Types

- enum **OsclActivePriority** { **EPriorityIdle** = -100, **EPriorityLow** = -20, **EPriorityNominal** = 0, **EPriorityHigh** = 10, **EPriorityHighest** = 20 }

### Public Methods

- OSCL\_IMPORT\_REF **OsclActiveObject** (int32 aPriority, const char name[ ])
- virtual OSCL\_IMPORT\_REF ~**OsclActiveObject** ()
- OSCL\_IMPORT\_REF void **SetBusy** ()
- OSCL\_IMPORT\_REF bool **IsBusy** () const
- OSCL\_IMPORT\_REF void **PendForExec** ()
- OSCL\_IMPORT\_REF void **PendComplete** (int32 aStatus)
- OSCL\_IMPORT\_REF void **AddToScheduler** ()
- OSCL\_IMPORT\_REF void **RemoveFromScheduler** ()
- OSCL\_IMPORT\_REF void **RunIfNotReady** ()
- OSCL\_IMPORT\_REF void **Cancel** ()
- OSCL\_IMPORT\_REF int32 **Priority** () const
- OSCL\_IMPORT\_REF int32 **Status** () const
- OSCL\_IMPORT\_REF void **SetStatus** (int32)
- OSCL\_IMPORT\_REF **OsclAOStatus & StatusRef** ()

## Protected Methods

- virtual OSCL\_IMPORT\_REF void [DoCancel \(\)](#)
- virtual OSCL\_IMPORT\_REF int32 [RunError \(int32 aError\)](#)

### 6.91.1 Detailed Description

User base class for execution objects. OsclActiveObject defines an execution object without any timer. This AO can be used across threads, i.e. the request can be activated in one thread and completed in another.

### 6.91.2 Member Enumeration Documentation

#### 6.91.2.1 enum OsclActiveObject::OsclActivePriority

Scheduling priorities.

##### Enumeration values:

- EPriorityIdle** A low priority, useful for execution objects representing background processing.  
**EPriorityLow** A priority higher than EPriorityIdle but lower than EPriorityNominal.  
**EPriorityNominal** Most exec objects will have this priority.  
**EPriorityHigh** A priority higher than EPriorityNominal; useful for execution objects handling user input.  
**EPriorityHighest** A priority higher than EPriorityHighest.

### 6.91.3 Constructor & Destructor Documentation

#### 6.91.3.1 OSCL\_IMPORT\_REF OsclActiveObject::OsclActiveObject (int32 *aPriority*, const char *name*[ ])

Constructor.

##### Parameters:

- aPriority* (input param): scheduling priority  
*name* (input param): optional name for this AO.

#### 6.91.3.2 virtual OSCL\_IMPORT\_REF OsclActiveObject::~OsclActiveObject () [virtual]

Destructor.

### 6.91.4 Member Function Documentation

#### 6.91.4.1 OSCL\_IMPORT\_REF void OsclActiveObject::AddToScheduler ()

Add this exec object to the current thread's scheduler.

Reimplemented from [PVActiveBase](#).

**6.91.4.2 OSCL\_IMPORT\_REF void OsclActiveObject::Cancel ()**

Cancel any pending request. If the request is readied, this will call the DoCancel routine, wait for the request to cancel, then set the request idle. The AO will not run. If the request is not readied, it does nothing. Request must be canceled from the same thread in which it is scheduled.

Reimplemented from [PVActiveBase](#).

**6.91.4.3 virtual OSCL\_IMPORT\_REF void OsclActiveObject::DoCancel () [protected, virtual]**

Cancel request handler. This gets called by scheduler when the request is cancelled. The default routine will complete the request. If any additional action is needed, the derived class may override this. If the derived class does override DoCancel, it must complete the request.

Implements [PVActiveBase](#).

Reimplemented in [OsclDNSRequestAO](#), and [OsclSocketRequestAO](#).

**6.91.4.4 OSCL\_IMPORT\_REF bool OsclActiveObject::IsBusy ()**

Return true if this AO is pending, false otherwise.

**6.91.4.5 OSCL\_IMPORT\_REF void OsclActiveObject::PendComplete (int32 aStatus)**

Complete the active request for the AO. This API is thread-safe. If the request is not pending, this call will leave.

**Parameters:**

*aStatus*: request completion status.

**6.91.4.6 OSCL\_IMPORT\_REF void OsclActiveObject::PendForExec ()**

Set request active for this AO and set the status to pending. PendForExec is identical to SetActive, but it additionally sets the request status to OSCL\_REQUEST\_PENDING.

**6.91.4.7 OSCL\_IMPORT\_REF int32 OsclActiveObject::Priority ()**

Return scheduling priority of this exec object.

**6.91.4.8 OSCL\_IMPORT\_REF void OsclActiveObject::RemoveFromScheduler ()**

Remove this AO from its scheduler. Will leave if the calling thread context does not match the scheduling thread. Cancels any readied request before removing.

Reimplemented from [PVActiveBase](#).

**6.91.4.9 virtual OSCL\_IMPORT\_REF int32 OsclActiveObject::RunError (int32 *aError*)  
[protected, virtual]**

Run Error handler. This gets called by scheduler when the Run routine leaves. The default implementation simply returns the leave code. If the derived class wants to handle errors from Run, it may override this. The RunError should return OsclErrNone if it handles the error, otherwise it should return the input error code.

**Parameters:**

*aError*: the leave code generated by the Run.

Implements [PVActiveBase](#).

**6.91.4.10 OSCL\_IMPORT\_REF void OsclActiveObject::RunIfNotReady ()**

Complete this AO's request immediately. If the AO is already active, this will do nothing. Will leave if the AO is not added to any scheduler, or if the calling thread context does not match the scheduling thread.

**6.91.4.11 OSCL\_IMPORT\_REF void OsclActiveObject::SetBusy ()**

Set object ready for this AO, additionally sets the request status to OSCL\_REQUEST\_PENDING. Will leave if the request is already readied, or the execution object is not added to any scheduler, or the calling thread context does not match the scheduler thread.

**6.91.4.12 OSCL\_IMPORT\_REF void OsclActiveObject::SetStatus (int32)****6.91.4.13 OSCL\_IMPORT\_REF int32 OsclActiveObject::Status ()**

Request status access

**6.91.4.14 OSCL\_IMPORT\_REF OsclAOStatus& OsclActiveObject::StatusRef ()**

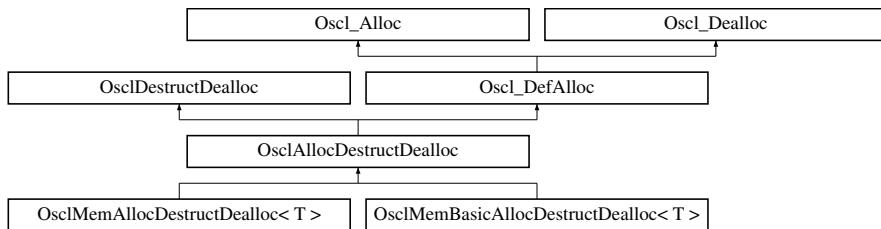
The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_ao.h](#)

## 6.92 OsclAllocDestructDealloc Class Reference

```
#include <oscl_defalloc.h>
```

Inheritance diagram for OsclAllocDestructDealloc::



### Public Methods

- virtual ~OsclAllocDestructDealloc ()

#### 6.92.1 Constructor & Destructor Documentation

**6.92.1.1 virtual OsclAllocDestructDealloc::~OsclAllocDestructDealloc () [inline, virtual]**

The documentation for this class was generated from the following file:

- [oscl\\_defalloc.h](#)

## 6.93 OsclAOStatus Class Reference

```
#include <oscl_aostatus.h>
```

### Public Methods

- OSCL\_INLINE OsclAOStatus ()
- OSCL\_INLINE OsclAOStatus (int32 aStatus)
- OSCL\_INLINE int32 operator= (int32 aStatus)
- OSCL\_INLINE int32 operator== (int32 aStatus) const
- OSCL\_INLINE int32 operator!= (int32 aStatus) const
- OSCL\_INLINE int32 operator>= (int32 aStatus) const
- OSCL\_INLINE int32 operator<= (int32 aStatus) const
- OSCL\_INLINE int32 operator> (int32 aStatus) const
- OSCL\_INLINE int32 operator< (int32 aStatus) const
- OSCL\_INLINE int32 Value () const

#### 6.93.1 Constructor & Destructor Documentation

6.93.1.1 OSCL\_INLINE OsclAOStatus::OsclAOStatus ()

6.93.1.2 OSCL\_INLINE OsclAOStatus::OsclAOStatus (int32 *aStatus*)

#### 6.93.2 Member Function Documentation

6.93.2.1 OSCL\_INLINE int32 OsclAOStatus::operator!= (int32 *aStatus*) const

6.93.2.2 OSCL\_INLINE int32 OsclAOStatus::operator< (int32 *aStatus*) const

6.93.2.3 OSCL\_INLINE int32 OsclAOStatus::operator<= (int32 *aStatus*) const

6.93.2.4 OSCL\_INLINE int32 OsclAOStatus::operator= (int32 *aStatus*)

6.93.2.5 OSCL\_INLINE int32 OsclAOStatus::operator== (int32 *aStatus*) const

6.93.2.6 OSCL\_INLINE int32 OsclAOStatus::operator> (int32 *aStatus*) const

6.93.2.7 OSCL\_INLINE int32 OsclAOStatus::operator>= (int32 *aStatus*) const

6.93.2.8 OSCL\_INLINE int32 OsclAOStatus::Value ()

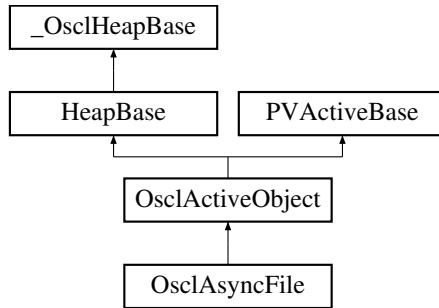
The documentation for this class was generated from the following file:

- [oscl\\_aostatus.h](#)

## 6.94 OsclAsyncFile Class Reference

```
#include <oscl_file_async_read.h>
```

Inheritance diagram for OsclAsyncFile::



### Public Methods

- [`~OsclAsyncFile \(\)`](#)
- [`int32 Open \(const oscl\_wchar \*filename, uint32 mode, const OsclNativeFileParams &params, Oscl\_FileServer &fileserv\)`](#)
- [`int32 Open \(const char \*filename, uint32 mode, const OsclNativeFileParams &params, Oscl\_FileServer &fileserv\)`](#)
- [`int32 Seek \(TOsclFileOffset offset, Oscl\_File::seek\_type origin\)`](#)
- [`TOsclFileOffset Tell \(\)`](#)
- [`uint32 Read \(OsclAny \*aBuffer1, uint32 aDataSize, uint32 aNumElements\)`](#)
- [`int32 EndOfFile \(\)`](#)
- [`TOsclFileOffset Size \(\)`](#)
- [`int32 Close \(\)`](#)
- [`uint32 Write \(const OsclAny \*aBuffer1, uint32 aDataSize, uint32 aNumElements\)`](#)
- [`uint32 Flush \(\)`](#)

### Static Public Methods

- [`OsclAsyncFile \* NewL \(OsclNativeFile &aAsyncFile, int32 aCacheSize, PVLogger \*\)`](#)
- [`void Delete \(OsclAsyncFile \*\)`](#)

### Data Fields

- [`uint32 iNumOfRun`](#)
- [`uint32 iNumOfRunErr`](#)

#### 6.94.1 Detailed Description

OsclAsyncFile

## 6.94.2 Constructor & Destructor Documentation

### 6.94.2.1 OsclAsyncFile::~OsclAsyncFile ()

Destructor.

## 6.94.3 Member Function Documentation

### 6.94.3.1 int32 OsclAsyncFile::Close ()

### 6.94.3.2 void OsclAsyncFile::Delete (OsclAsyncFile \*) [static]

### 6.94.3.3 int32 OsclAsyncFile::EndOfFile ()

### 6.94.3.4 uint32 OsclAsyncFile::Flush () [inline]

### 6.94.3.5 OsclAsyncFile\* OsclAsyncFile::NewL (OsclNativeFile & aSyncFile, int32 aCacheSize, PVLogger \*) [static]

Two-phased constructor.

#### Parameters:

*aSyncFile*: open handle for async file read. Note: it is the caller's job to open/close this file handle.

*aSyncFile*: duplicate open handle for sync file read. Note: it is the caller's job to open this file handle, but this class will close the handle.

*aCacheSize*: size of one of the individual cache buffers. The total cached data size will be larger, since multiple buffers are used.

*aStartAsyncRead*: When true, async file read will start immediately. When false, read will not begin until StartAsyncRead is called.

- 6.94.3.6 int32 OsclAsyncFile::Open (const char \**filename*, uint32 *mode*, const OsclNativeFileParams & *params*, Oscl\_FileServer & *fileserv*)
- 6.94.3.7 int32 OsclAsyncFile::Open (const oscl\_wchar \**filename*, uint32 *mode*, const OsclNativeFileParams & *params*, Oscl\_FileServer & *fileserv*)
- 6.94.3.8 uint32 OsclAsyncFile::Read (OsclAny \**aBuffer1*, uint32 *aDataSize*, uint32 *aNumElements*)
- 6.94.3.9 int32 OsclAsyncFile::Seek (TOsclFileOffset *offset*, Oscl\_File::seek\_type *origin*)
- 6.94.3.10 TOsclFileOffset OsclAsyncFile::Size ()
- 6.94.3.11 TOsclFileOffset OsclAsyncFile::Tell ()
- 6.94.3.12 uint32 OsclAsyncFile::Write (const OsclAny \**aBuffer1*, uint32 *aDataSize*, uint32 *aNumElements*) [inline]

#### 6.94.4 Field Documentation

- 6.94.4.1 uint32 OsclAsyncFile::iNumOfRun

- 6.94.4.2 uint32 OsclAsyncFile::iNumOfRunErr

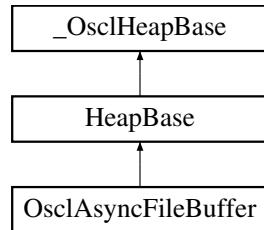
The documentation for this class was generated from the following file:

- [oscl\\_file\\_async\\_read.h](#)

## 6.95 OsclAsyncFileBuffer Class Reference

```
#include <oscl_file_async_read.h>
```

Inheritance diagram for OsclAsyncFileBuffer::



### Public Methods

- `~OsclAsyncFileBuffer ()`
- `void CleanInUse ()`
- `void SetInUse ()`
- `bool IsInUse ()`
- `bool IsValid ()`
- `TOsclFileOffset Offset ()`
- `void SetOffset (TOsclFileOffset aOffset)`
- `int32 Length ()`
- `bool HasThisOffset (TOsclFileOffset aOffset)`
- `int32 Id ()`
- `OsclBuf * Buffer ()`
- `void UpdateData ()`
- `void StartAsyncRead (bool aStartAsyncRead)`

### Static Public Methods

- `OsclAsyncFileBuffer * NewL (int32 aBufferSize, int32 aId)`

#### 6.95.1 Detailed Description

Buffer class used with async read. We keep an array of these, covering consecutive areas of the file. This allows for some seeking without requiring a full flush & refill each time.

## 6.95.2 Constructor & Destructor Documentation

6.95.2.1 `OsclAsyncFileBuffer::~OsclAsyncFileBuffer ()`

## 6.95.3 Member Function Documentation

6.95.3.1 `OsclBuf* OsclAsyncFileBuffer::Buffer ()`

6.95.3.2 `void OsclAsyncFileBuffer::CleanInUse () [inline]`

6.95.3.3 `bool OsclAsyncFileBuffer::HasThisOffset (TOsclFileOffset aOffset)`

6.95.3.4 `int32 OsclAsyncFileBuffer::Id () [inline]`

6.95.3.5 `bool OsclAsyncFileBuffer::IsInUse () [inline]`

6.95.3.6 `bool OsclAsyncFileBuffer::IsValid () [inline]`

6.95.3.7 `int32 OsclAsyncFileBuffer::Length () [inline]`

6.95.3.8 `OsclAsyncFileBuffer* OsclAsyncFileBuffer::NewL (int32 aBufferSize, int32 aId) [static]`

6.95.3.9 `TOsclFileOffset OsclAsyncFileBuffer::Offset () [inline]`

6.95.3.10 `void OsclAsyncFileBuffer::SetInUse () [inline]`

6.95.3.11 `void OsclAsyncFileBuffer::SetOffset (TOsclFileOffset aOffset) [inline]`

6.95.3.12 `void OsclAsyncFileBuffer::StartAsyncRead (bool aStartAsyncRead)`

6.95.3.13 `void OsclAsyncFileBuffer::UpdateData ()`

The documentation for this class was generated from the following file:

- `oscl_file_async_read.h`

## 6.96 OsclAuditCB Class Reference

```
#include <oscl_mem.h>
```

### Public Methods

- [OsclAuditCB \(\)](#)
- [OsclAuditCB \(const OsclMemStatsNode \\*myStatsNode, OsclMemAudit \\*ptr\)](#)

### Data Fields

- [const OsclMemStatsNode \\* pStatsNode](#)
- [OsclMemAudit \\* pAudit](#)

#### 6.96.1 Constructor & Destructor Documentation

**6.96.1.1 OsclAuditCB::OsclAuditCB () [inline]**

**6.96.1.2 OsclAuditCB::OsclAuditCB (const OsclMemStatsNode \* *myStatsNode*, OsclMemAudit \* *ptr*) [inline]**

#### 6.96.2 Field Documentation

**6.96.2.1 OsclMemAudit\* OsclAuditCB::pAudit**

**6.96.2.2 const OsclMemStatsNode\* OsclAuditCB::pStatsNode**

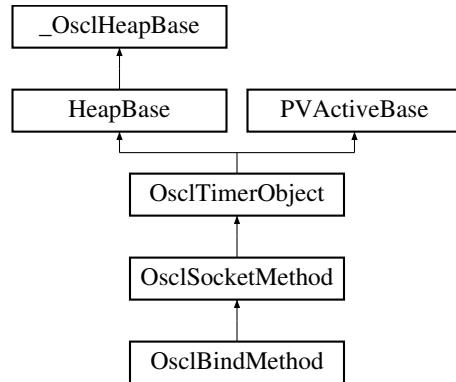
The documentation for this class was generated from the following file:

- [oscl\\_mem.h](#)

## 6.97 OsclBindMethod Class Reference

```
#include <oscl_socket_bind.h>
```

Inheritance diagram for OsclBindMethod::



### Public Methods

- [~OsclBindMethod \(\)](#)
- [TPVSocketEvent Bind \(OsclNetworkAddress &aAddress, int32 aTimeout\)](#)
- [OsclBindRequest \\* BindRequest \(\)](#)

### Static Public Methods

- [OsclBindMethod \\* NewL \(OsclIPSocketI &c\)](#)

#### 6.97.1 Constructor & Destructor Documentation

##### 6.97.1.1 OsclBindMethod::~OsclBindMethod ()

#### 6.97.2 Member Function Documentation

##### 6.97.2.1 TPVSocketEvent OsclBindMethod::Bind (OsclNetworkAddress & aAddress, int32 aTimeout)

##### 6.97.2.2 OsclBindRequest\* OsclBindMethod::BindRequest () [inline]

##### 6.97.2.3 OsclBindMethod\* OsclBindMethod::NewL (OsclIPSocketI &c) [static]

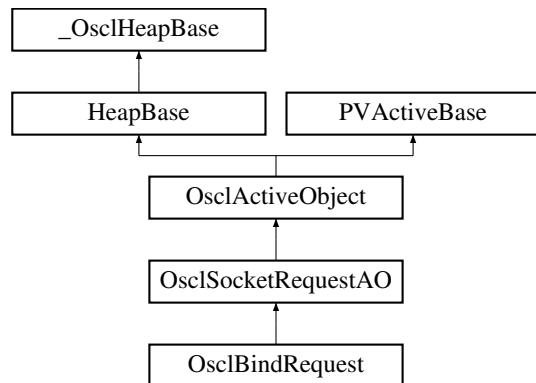
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_bind.h](#)

## 6.98 OsclBindRequest Class Reference

```
#include <oscl_socket_bind.h>
```

Inheritance diagram for OsclBindRequest::



### Public Methods

- [OsclBindRequest \(OsclSocketMethod &c\)](#)
- [void Bind \(OsclNetworkAddress &aAddress\)](#)

#### 6.98.1 Detailed Description

This is the AO that interacts with the socket server

#### 6.98.2 Constructor & Destructor Documentation

**6.98.2.1 OsclBindRequest::OsclBindRequest ([OsclSocketMethod & c](#)) [inline]**

#### 6.98.3 Member Function Documentation

**6.98.3.1 void OsclBindRequest::Bind ([OsclNetworkAddress & aAddress](#))**

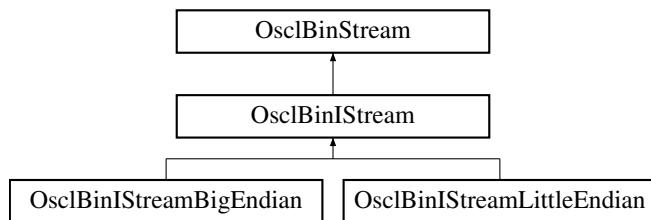
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_bind.h](#)

## 6.99 OsclBinIStream Class Reference

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinIStream::



### Public Methods

- [OsclBinIStream \(\)](#)
- [~OsclBinIStream \(\)](#)
- uint8 [Read\\_uint8 \(\)](#)

*This method reads an unsigned short from the stream.*

- OsclBinIStream & [get \(int8 \\*data, int32 size\)](#)

*This method reads 'length' number of bytes from the stream and places them in 'data'.*

#### 6.99.1 Constructor & Destructor Documentation

##### 6.99.1.1 OsclBinIStream::OsclBinIStream () [inline]

##### 6.99.1.2 OsclBinIStream::~OsclBinIStream () [inline]

#### 6.99.2 Member Function Documentation

##### 6.99.2.1 OsclBinIStream& OsclBinIStream::get (int8 \* data, int32 size)

This method reads 'length' number of bytes from the stream and places them in 'data'.

**Parameters:**

*data* is a pointer to the place to store the bytes read

*size* is the number of bytes to read

##### 6.99.2.2 uint8 OsclBinIStream::Read\_uint8 ()

This method reads an unsigned short from the stream.

**Returns:**

Unsigned short read from the stream.

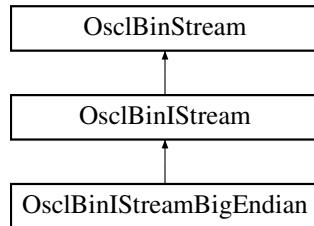
The documentation for this class was generated from the following file:

- [oscl\\_bin\\_stream.h](#)

## 6.100 OsclBinIStreamBigEndian Class Reference

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinIStreamBigEndian::



### Public Methods

- [OsclBinIStreamBigEndian \(\)](#)
- [void Read \(int8 &data\)](#)
- [void Read \(uint8 &data\)](#)
- [void Read \(int16 &data\)](#)
- [void Read \(uint16 &data\)](#)
- [void Read \(int32 &data\)](#)
- [void Read \(uint32 &data\)](#)
- [OsclBinIStreamBigEndian & operator>> \(int8 &data\)](#)

*This method reads a int8 from the stream and stores it in 'data'.*

- [OsclBinIStream & operator>> \(uint8 &data\)](#)

*This method reads a uint8 from the stream and stores it in 'data'.*

- [OsclBinIStreamBigEndian & operator>> \(int16 &data\)](#)

*This method reads a int16 from the stream and stores it in 'data'.*

- [OsclBinIStreamBigEndian & operator>> \(uint16 &data\)](#)

*This method reads a uint16 from the stream and stores it in 'data'.*

- [OsclBinIStreamBigEndian & operator>> \(int32 &data\)](#)

*This method reads a int32 from the stream and stores it in 'data'.*

- [OsclBinIStreamBigEndian & operator>> \(uint32 &data\)](#)

*This method reads a uint32 from the stream and stores it in 'data'.*

- [uint16 Read\\_uint16 \(\)](#)

*This method reads an unsigned short from the stream.*

- [uint32 Read\\_uint32 \(\)](#)

*This method reads an unsigned long from the stream.*

## 6.100.1 Constructor & Destructor Documentation

**6.100.1.1 OsclBinIStreamBigEndian::OsclBinIStreamBigEndian () [inline]**

## 6.100.2 Member Function Documentation

**6.100.2.1 OsclBinIStreamBigEndian& OsclBinIStreamBigEndian::operator>> (uint32 & data)**

This method reads a uint32 from the stream and stores it in 'data'.

**6.100.2.2 OsclBinIStreamBigEndian& OsclBinIStreamBigEndian::operator>> (int32 & data)**

This method reads a int32 from the stream and stores it in 'data'.

**6.100.2.3 OsclBinIStreamBigEndian& OsclBinIStreamBigEndian::operator>> (uint16 & data)**

This method reads a uint16 from the stream and stores it in 'data'.

**6.100.2.4 OsclBinIStreamBigEndian& OsclBinIStreamBigEndian::operator>> (int16 & data)**

This method reads a int16 from the stream and stores it in 'data'.

**6.100.2.5 OsclBinIStream& OsclBinIStreamBigEndian::operator>> (uint8 & data)**

This method reads a uint8 from the stream and stores it in 'data'.

**6.100.2.6 OsclBinIStreamBigEndian& OsclBinIStreamBigEndian::operator>> (int8 & data)**

This method reads a int8 from the stream and stores it in 'data'.

**6.100.2.7 void OsclBinIStreamBigEndian::Read (uint32 & data)**

**6.100.2.8 void OsclBinIStreamBigEndian::Read (int32 & data)**

**6.100.2.9 void OsclBinIStreamBigEndian::Read (uint16 & data)**

**6.100.2.10 void OsclBinIStreamBigEndian::Read (int16 & data)**

**6.100.2.11 void OsclBinIStreamBigEndian::Read (uint8 & data)**

**6.100.2.12 void OsclBinIStreamBigEndian::Read (int8 & data)**

**6.100.2.13 uint16 OsclBinIStreamBigEndian::Read\_uint16 ()**

This method reads an unsigned short from the stream.

### Returns:

Unsigned short read from the stream.

### 6.100.2.14 uint32 OsclBinIStreamBigEndian::Read\_uint32 ()

This method reads an unsigned long from the stream.

**Returns:**

unsigned long read from the stream.

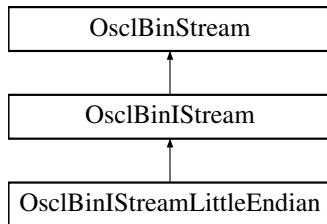
The documentation for this class was generated from the following file:

- [oscl\\_bin\\_stream.h](#)

## 6.101 OsclBinIStreamLittleEndian Class Reference

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinIStreamLittleEndian::



### Public Methods

- [OsclBinIStreamLittleEndian \(\)](#)  
[OsclBinIStreamLittleEndian & operator>> \(int8 &data\)](#)  
*This method reads a int8 from the stream and stores it in 'data'.*
- [OsclBinIStreamLittleEndian & operator>> \(uint8 &data\)](#)  
*This method reads a uint8 from the stream and stores it in 'data'.*
- [OsclBinIStreamLittleEndian & operator>> \(int16 &data\)](#)  
*This method reads a int16 from the stream and stores it in 'data'.*
- [OsclBinIStreamLittleEndian & operator>> \(uint16 &data\)](#)  
*This method reads a uint16 from the stream and stores it in 'data'.*
- [OsclBinIStreamLittleEndian & operator>> \(int32 &data\)](#)  
*This method reads a int32 from the stream and stores it in 'data'.*
- [OsclBinIStreamLittleEndian & operator>> \(uint32 &data\)](#)  
*This method reads a uint32 from the stream and stores it in 'data'.*

### Protected Methods

- uint16 [Read\\_uint16 \(\)](#)
- uint32 [Read\\_uint32 \(\)](#)

### 6.101.1 Constructor & Destructor Documentation

**6.101.1.1 OsclBinIStreamLittleEndian::OsclBinIStreamLittleEndian () [inline]**

### 6.101.2 Member Function Documentation

**6.101.2.1 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (uint32 & data)**

This method reads a uint32 from the stream and stores it in 'data'.

**6.101.2.2 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (int32 & data)**

This method reads a int32 from the stream and stores it in 'data'.

**6.101.2.3 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (uint16 & data)**

This method reads a uint16 from the stream and stores it in 'data'.

**6.101.2.4 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (int16 & data)**

This method reads a int16 from the stream and stores it in 'data'.

**6.101.2.5 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (uint8 & data)**

This method reads a uint8 from the stream and stores it in 'data'.

**6.101.2.6 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (int8 & data)**

This method reads a int8 from the stream and stores it in 'data'.

**6.101.2.7 uint16 OsclBinIStreamLittleEndian::Read\_uint16 () [protected]**

**6.101.2.8 uint32 OsclBinIStreamLittleEndian::Read\_uint32 () [protected]**

The documentation for this class was generated from the following file:

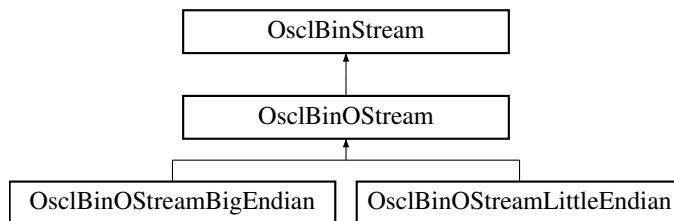
- [oscl\\_bin\\_stream.h](#)

## 6.102 OsclBinOStream Class Reference

Class OsclBinOStream implements the basic stream functions for an output stream.

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinOStream:::



### Public Methods

- [OsclBinOStream \(\)](#)
- virtual [~OsclBinOStream \(\)](#)
- [OsclBinOStream & write \(const int8 \\*data, int32 size\)](#)

*This method writes 'length' number of bytes stored in 'data' to the stream.*

### 6.102.1 Detailed Description

Class OsclBinOStream implements the basic stream functions for an output stream.

### 6.102.2 Constructor & Destructor Documentation

**6.102.2.1 [OsclBinOStream::OsclBinOStream \(\) \[inline\]](#)**

**6.102.2.2 [virtual OsclBinOStream::~OsclBinOStream \(\) \[inline, virtual\]](#)**

### 6.102.3 Member Function Documentation

**6.102.3.1 [OsclBinOStream& OsclBinOStream::write \(const int8 \\* data, int32 size\)](#)**

This method writes 'length' number of bytes stored in 'data' to the stream.

The documentation for this class was generated from the following file:

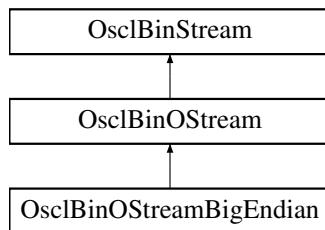
- [oscl\\_bin\\_stream.h](#)

## 6.103 OsclBinOStreamBigEndian Class Reference

Class OsclBinOStreamBigEndian implements a binary output stream using big endian byte ordering.

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinOStreamBigEndian::



### Public Methods

- [OsclBinOStreamBigEndian \(\)](#)
- OsclBinOStreamBigEndian & [operator<< \(const int8 &data\)](#)  
*This method writes a int8 from 'data' to the stream.*
- OsclBinOStreamBigEndian & [operator<< \(const uint8 &data\)](#)  
*This method writes a uint8 from 'data' to the stream.*
- OsclBinOStreamBigEndian & [operator<< \(const int16 &data\)](#)  
*This method writes a int16 from 'data' to the stream.*
- OsclBinOStreamBigEndian & [operator<< \(const uint16 &data\)](#)  
*This method writes a uint16 from 'data' to the stream.*
- OsclBinOStreamBigEndian & [operator<< \(const int32 &data\)](#)  
*This method writes a int32 from 'data' to the stream.*
- OsclBinOStreamBigEndian & [operator<< \(const uint32 &data\)](#)  
*This method writes a uint32 from 'data' to the stream.*

### Protected Methods

- void [WriteUnsignedShort \(const uint16 data\)](#)
- void [WriteUnsignedLong \(const uint32 data\)](#)

#### 6.103.1 Detailed Description

Class OsclBinOStreamBigEndian implements a binary output stream using big endian byte ordering.

## 6.103.2 Constructor & Destructor Documentation

**6.103.2.1 OsclBinOStreamBigEndian::OsclBinOStreamBigEndian () [inline]**

## 6.103.3 Member Function Documentation

**6.103.3.1 OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const uint32 & data)**

This method writes a uint32 from 'data' to the stream.

**6.103.3.2 OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const int32 & data)**

This method writes a int32 from 'data' to the stream.

**6.103.3.3 OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const uint16 & data)**

This method writes a uint16 from 'data' to the stream.

**6.103.3.4 OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const int16 & data)**

This method writes a int16 from 'data' to the stream.

**6.103.3.5 OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const uint8 & data)**

This method writes a uint8 from 'data' to the stream.

**6.103.3.6 OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const int8 & data)**

This method writes a int8 from 'data' to the stream.

**6.103.3.7 void OsclBinOStreamBigEndian::WriteUnsignedLong (const uint32 data) [protected]**

**6.103.3.8 void OsclBinOStreamBigEndian::WriteUnsignedShort (const uint16 data) [protected]**

The documentation for this class was generated from the following file:

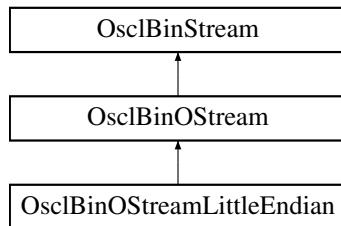
- [oscl\\_bin\\_stream.h](#)

## 6.104 OsclBinOStreamLittleEndian Class Reference

Class OsclBinOStreamLittleEndian implements a binary output stream using little endian byte ordering.

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinOStreamLittleEndian::



### Public Methods

- [OsclBinOStreamLittleEndian \(\)](#)
- [OsclBinOStreamLittleEndian & operator<< \(const int8 &data\)](#)  
*This method writes a int8 from 'data' to the stream.*
- [OsclBinOStreamLittleEndian & operator<< \(const uint8 &data\)](#)  
*This method writes a uint8 from 'data' to the stream.*
- [OsclBinOStreamLittleEndian & operator<< \(const int16 &data\)](#)  
*This method writes a int16 from 'data' to the stream.*
- [OsclBinOStreamLittleEndian & operator<< \(const uint16 &data\)](#)  
*This method writes a uint16 from 'data' to the stream.*
- [OsclBinOStreamLittleEndian & operator<< \(const int32 &data\)](#)  
*This method writes a int32 from 'data' to the stream.*
- [OsclBinOStreamLittleEndian & operator<< \(const uint32 &data\)](#)  
*This method writes a uint32 from 'data' to the stream.*

### Protected Methods

- [void WriteUnsignedShort \(const uint16 data\)](#)  
*This method writes 'data' (unsigned short) to the stream.*
- [void WriteUnsignedLong \(const uint32 data\)](#)  
*This method writes 'data' (unsigned long) to the stream.*

#### 6.104.1 Detailed Description

Class OsclBinOStreamLittleEndian implements a binary output stream using little endian byte ordering.

## 6.104.2 Constructor & Destructor Documentation

**6.104.2.1 OsclBinOStreamLittleEndian::OsclBinOStreamLittleEndian () [inline]**

## 6.104.3 Member Function Documentation

**6.104.3.1 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const uint32 & data)**

This method writes a uint32 from 'data' to the stream.

**6.104.3.2 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const int32 & data)**

This method writes a int32 from 'data' to the stream.

**6.104.3.3 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const uint16 & data)**

This method writes a uint16 from 'data' to the stream.

**6.104.3.4 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const int16 & data)**

This method writes a int16 from 'data' to the stream.

**6.104.3.5 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const uint8 & data)**

This method writes a uint8 from 'data' to the stream.

**6.104.3.6 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const int8 & data)**

This method writes a int8 from 'data' to the stream.

**6.104.3.7 void OsclBinOStreamLittleEndian::WriteUnsignedLong (const uint32 data) [protected]**

This method writes 'data' (unsigned long) to the stream.

**6.104.3.8 void OsclBinOStreamLittleEndian::WriteUnsignedShort (const uint16 data) [protected]**

This method writes 'data' (unsigned short) to the stream.

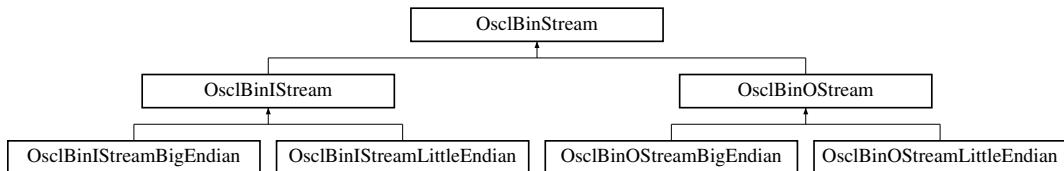
The documentation for this class was generated from the following file:

- [oscl\\_bin\\_stream.h](#)

## 6.105 OsclBinStream Class Reference

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinStream::



### Public Methods

- [OsclBinStream \(\)](#)
- [bool good \(\)](#)

*This method determines if the stream is ok.*

- [bool eof \(\)](#)

*This method determines if end of stream has been reached.*

- [bool fail \(\)](#)

*This method determines if an error has occurred in the stream.*

- [void Attach \(void \\*buffer, uint32 l\\_length\)](#)

*This method specifies the data buffer to attach to the stream.*

- [void Attach \(const uint32 numFragments, const OsclMemoryFragment \\*fragPtr\)](#)

*This method specifies the memory fragment array to use for input.*

- [uint32 tellg \(\)](#)

*This method returns the current stream position.*

- [void Seek \(uint32 absPosition\)](#)

*This method seeks to the specified stream position.*

- [uint32 PositionInBlock \(\)](#)

*This method returns the current stream position.*

- [void seekFromCurrentPosition \(int32 offset\)](#)

*This method seeks to the specified offset from the current location.*

### Protected Types

- enum [state\\_t](#) { [GOOD\\_STATE](#), [EOF\\_STATE](#), [FAIL\\_STATE](#) }

## Protected Methods

- bool [ReserveSpace](#) (uint32 size)
- bool [HaveRoomInCurrentBlock](#) (uint32 size)

## Protected Attributes

- [state\\_t state](#)
- uint8 \* [pBasePosition](#)
- uint8 \* [pPosition](#)
- uint32 [length](#)
- const [OsclMemoryFragment](#) \* [nextFragPtr](#)
- int [fragsLeft](#)
- const [OsclMemoryFragment](#) \* [firstFragPtr](#)
- int [numFrags](#)
- [OsclMemoryFragment](#) [specialFragBuffer](#)

### 6.105.1 Member Enumeration Documentation

#### 6.105.1.1 enum OsclBinStream::state\_t [protected]

Enumeration values:

- GOOD\_STATE**
- EOF\_STATE**
- FAIL\_STATE**

### 6.105.2 Constructor & Destructor Documentation

#### 6.105.2.1 OsclBinStream::OsclBinStream () [inline]

### 6.105.3 Member Function Documentation

#### 6.105.3.1 void OsclBinStream::Attach (const uint32 *numFragments*, const [OsclMemoryFragment](#) \**fragPtr*)

This method specifies the memory fragment array to use for input.

This array should remain static while the stream refers to it.

**Parameters:**

*numFragments* is the number of elements in the array

*fragPtr* is the pointer to the MemoryFragment array

#### 6.105.3.2 void OsclBinStream::Attach (void \**buffer*, uint32 *l\_length*)

This methods specifies the data buffer to attach to the stream.

**Parameters:**

*buffer* will provide the input

*length* of the buffer

**6.105.3.3 bool OsclBinStream::eof ()**

This method determines if end of stream has been reached.

**Returns:**

true if end of stream has been reached.

**6.105.3.4 bool OsclBinStream::fail ()**

This method determines if an error has occurred in the stream.

**Returns:**

true if an error occurred in the stream.

**6.105.3.5 bool OsclBinStream::good ()**

This method determines if the stream is ok.

**Returns:**

true if stream is ok.

**6.105.3.6 bool OsclBinStream::HaveRoomInCurrentBlock (uint32 *size*) [protected]****6.105.3.7 uint32 OsclBinStream::PositionInBlock ()**

This method returns the current stream position.

**Returns:**

stream position.

**6.105.3.8 bool OsclBinStream::ReserveSpace (uint32 *size*) [protected]****6.105.3.9 void OsclBinStream::Seek (uint32 *absPosition*)**

This method seeks to the specified stream position.

**Returns:**

Stream position.

**6.105.3.10 void OsclBinStream::seekFromCurrentPosition (int32 *offset*)**

This method seeks to the specified offset from the current location.

**Parameters:**

*offset* from current stream location

### 6.105.3.11 uint32 OsclBinStream::tellg ()

This method returns the current stream position.

This method is to be used if the input stream is a pointer to the MemoryFragment array

**Returns:**

Stream position.

## 6.105.4 Field Documentation

**6.105.4.1 const OsclMemoryFragment\* OsclBinStream::firstFragPtr [protected]**

**6.105.4.2 int OsclBinStream::fragsLeft [protected]**

**6.105.4.3 uint32 OsclBinStream::length [protected]**

**6.105.4.4 const OsclMemoryFragment\* OsclBinStream::nextFragPtr [protected]**

**6.105.4.5 int OsclBinStream::numFrags [protected]**

**6.105.4.6 uint8\* OsclBinStream::pBasePosition [protected]**

**6.105.4.7 uint8\* OsclBinStream::pPosition [protected]**

**6.105.4.8 OsclMemoryFragment OsclBinStream::specialFragBuffer [protected]**

**6.105.4.9 state\_t OsclBinStream::state [protected]**

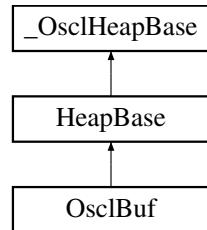
The documentation for this class was generated from the following file:

- [oscl\\_bin\\_stream.h](#)

## 6.106 OsclBuf Class Reference

```
#include <oscl_file_async_read.h>
```

Inheritance diagram for OsclBuf::



### Public Methods

- [OsclBuf](#) (int32 size)
- int32 [Length](#) ()
- [OsclPtr Des](#) ()
- [OsclPtrC DesC](#) ()

### Static Public Methods

- OsclBuf \* [NewL](#) (int32 size)
- void [Delete](#) (OsclBuf \*a)

### Data Fields

- uint8 \* [iBuffer](#)
- int32 [iMaxLength](#)
- int32 [iLength](#)

## 6.106.1 Constructor & Destructor Documentation

**6.106.1.1** `OsclBuf::OsclBuf (int32 size)` [inline]

## 6.106.2 Member Function Documentation

**6.106.2.1** `void OsclBuf::Delete (OsclBuf * a)` [inline, static]

**6.106.2.2** `OsclPtr OsclBuf::Des ()` [inline]

**6.106.2.3** `OsclPtrC OsclBuf::DesC ()` [inline]

**6.106.2.4** `int32 OsclBuf::Length ()` [inline]

**6.106.2.5** `OsclBuf* OsclBuf::NewL (int32 size)` [inline, static]

## 6.106.3 Field Documentation

**6.106.3.1** `uint8* OsclBuf::iBuffer`

**6.106.3.2** `int32 OsclBuf::iLength`

**6.106.3.3** `int32 OsclBuf::iMaxLength`

The documentation for this class was generated from the following file:

- [oscl\\_file\\_async\\_read.h](#)

## 6.107 OsclCompareLess< T > Class Template Reference

```
#include <oscl_priqueue.h>
```

### Public Methods

- int [compare](#) (T &a, T &b) const

```
template<class T> class OsclCompareLess< T >
```

#### 6.107.1 Member Function Documentation

**6.107.1.1 template<class T> int OsclCompareLess< T >::compare (T & a, T & b) const  
[inline]**

The documentation for this class was generated from the following file:

- [oscl\\_priqueue.h](#)

## 6.108 OsclComponentRegistry Class Reference

```
#include <oscl_registry_serv_impl.h>
```

### Public Methods

- [OsclComponentRegistry \(\)](#)
- [~OsclComponentRegistry \(\)](#)
- [int32 Register \(uint32 &aId, OSCL\\_String &, OsclComponentFactory\)](#)
- [int32 Unregister \(OSCL\\_String &\)](#)
- [int32 Unregister \(uint32\)](#)
- [OsclComponentFactory FindExact \(OSCL\\_String &\)](#)
- [void FindHierarchical \(OSCL\\_String &, Oscl\\_Vector< OsclRegistryAccessElement, OsclMemAllocator > &\)](#)
- [void OpenSession \(\)](#)
- [void CloseSession \(\)](#)

### Data Fields

- [OsclComponentRegistryData iData](#)
- [OsclMutex iMutex](#)
- [uint32 iComponentIdCounter](#)
- [uint32 iNumSessions](#)

#### 6.108.1 Detailed Description

Thread-safe singleton registry object.

## 6.108.2 Constructor & Destructor Documentation

**6.108.2.1** `OsclComponentRegistry::OsclComponentRegistry ()`

**6.108.2.2** `OsclComponentRegistry::~OsclComponentRegistry ()`

## 6.108.3 Member Function Documentation

**6.108.3.1** `void OsclComponentRegistry::CloseSession ()`

**6.108.3.2** `OsclComponentFactory OsclComponentRegistry::FindExact (OSCL_String &)`

**6.108.3.3** `void OsclComponentRegistry::FindHierarchical (OSCL_String &, Oscl_Vector< OsclRegistryAccessElement, OsclMemAllocator > &)`

**6.108.3.4** `void OsclComponentRegistry::OpenSession ()`

**6.108.3.5** `int32 OsclComponentRegistry::Register (uint32 & aId, OSCL_String &, OsclComponentFactory)`

**6.108.3.6** `int32 OsclComponentRegistry::Unregister (uint32)`

**6.108.3.7** `int32 OsclComponentRegistry::Unregister (OSCL_String &)`

## 6.108.4 Field Documentation

**6.108.4.1** `uint32 OsclComponentRegistry::iComponentIdCounter`

**6.108.4.2** `OsclComponentRegistryData OsclComponentRegistry::iData`

**6.108.4.3** `OsclMutex OsclComponentRegistry::iMutex`

**6.108.4.4** `uint32 OsclComponentRegistry::iNumSessions`

The documentation for this class was generated from the following file:

- `oscl_registry_serv_impl.h`

## 6.109 OsclComponentRegistryData Class Reference

```
#include <oscl_registry_serv_impl.h>
```

### Public Methods

- [OsclComponentRegistryElement \\* Find \(OSCL\\_String &, bool aExact\)](#)

### Data Fields

- [Oscl\\_Vector< OsclComponentRegistryElement, OsclMemAllocator > iVec](#)

#### 6.109.1 Detailed Description

Registry

#### 6.109.2 Member Function Documentation

##### 6.109.2.1 [OsclComponentRegistryElement\\* OsclComponentRegistryData::Find \(OSCL\\_String &, bool aExact\)](#)

#### 6.109.3 Field Documentation

##### 6.109.3.1 [Oscl\\_Vector<OsclComponentRegistryElement, OsclMemAllocator> OsclComponentRegistryData::iVec](#)

The documentation for this class was generated from the following file:

- [oscl\\_registry\\_serv\\_impl.h](#)

## 6.110 OsclComponentRegistryElement Class Reference

```
#include <oscl_registry_serv_impl.h>
```

### Public Methods

- [OsclComponentRegistryElement \(OSCL\\_String &, OsclComponentFactory\)](#)
- [OsclComponentRegistryElement \(const OsclComponentRegistryElement &\)](#)
- [OsclComponentRegistryElement & operator= \(const OsclComponentRegistryElement &src\)](#)
- [~OsclComponentRegistryElement \(\)](#)
- [bool Match \(OSCL\\_String &aStr, bool aExact\)](#)

### Data Fields

- [OSCL\\_String \\* iId](#)
- [OsclComponentFactory iFactory](#)
- [uint32 iComponentId](#)

#### 6.110.1 Detailed Description

Data for each registered component.

#### 6.110.2 Constructor & Destructor Documentation

**6.110.2.1 OsclComponentRegistryElement::OsclComponentRegistryElement (OSCL\_String &, OsclComponentFactory)**

**6.110.2.2 OsclComponentRegistryElement::OsclComponentRegistryElement (const OsclComponentRegistryElement &)**

**6.110.2.3 OsclComponentRegistryElement::~OsclComponentRegistryElement ()**

#### 6.110.3 Member Function Documentation

**6.110.3.1 bool OsclComponentRegistryElement::Match (OSCL\_String & aStr, bool aExact)**

**6.110.3.2 OsclComponentRegistryElement& OsclComponentRegistryElement::operator= (const OsclComponentRegistryElement & src)**

#### 6.110.4 Field Documentation

**6.110.4.1 uint32 OsclComponentRegistryElement::iComponentId**

**6.110.4.2 OsclComponentFactory OsclComponentRegistryElement::iFactory**

**6.110.4.3 OSCL\_String\* OsclComponentRegistryElement::iId**

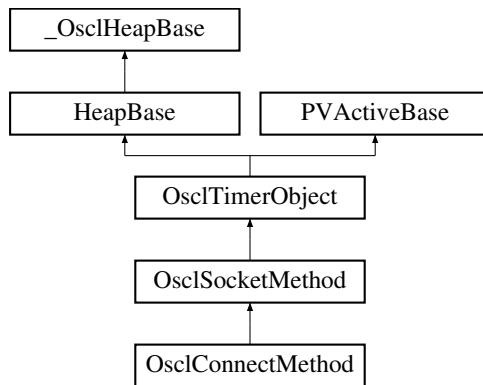
The documentation for this class was generated from the following file:

- [oscl\\_registry\\_serv\\_impl.h](#)

## 6.111 OsclConnectMethod Class Reference

```
#include <oscl_socket_connect.h>
```

Inheritance diagram for OsclConnectMethod::



### Public Methods

- [~OsclConnectMethod \(\)](#)
- [TPVSocketEvent Connect \(OsclNetworkAddress &aAddress, int32 aTimeout\)](#)
- [OsclConnectRequest \\* ConnectRequest \(\)](#)

### Static Public Methods

- [OsclConnectMethod \\* NewL \(OsclIPSocketI &c\)](#)

#### 6.111.1 Constructor & Destructor Documentation

##### 6.111.1.1 OsclConnectMethod::~OsclConnectMethod ()

#### 6.111.2 Member Function Documentation

##### 6.111.2.1 TPVSocketEvent OsclConnectMethod::Connect (OsclNetworkAddress & aAddress, int32 aTimeout)

##### 6.111.2.2 OsclConnectRequest\* OsclConnectMethod::ConnectRequest () [inline]

##### 6.111.2.3 OsclConnectMethod\* OsclConnectMethod::NewL (OsclIPSocketI & c) [static]

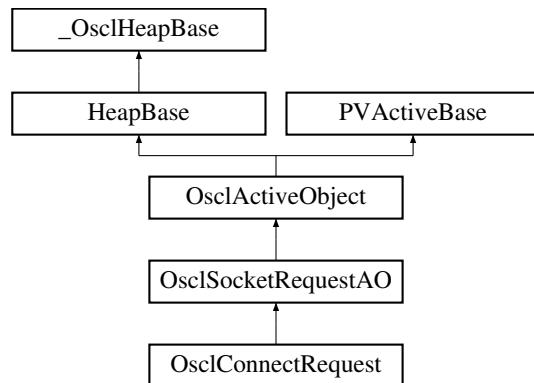
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_connect.h](#)

## 6.112 OsclConnectRequest Class Reference

```
#include <oscl_socket_connect.h>
```

Inheritance diagram for OsclConnectRequest::



### Public Methods

- [OsclConnectRequest \(OsclSocketMethod &c\)](#)
- [void Connect \(OsclNetworkAddress &aAddress\)](#)

#### 6.112.1 Detailed Description

This is the AO that interacts with the socket server

#### 6.112.2 Constructor & Destructor Documentation

**6.112.2.1 OsclConnectRequest::OsclConnectRequest ([OsclSocketMethod & c](#)) [inline]**

#### 6.112.3 Member Function Documentation

**6.112.3.1 void OsclConnectRequest::Connect ([OsclNetworkAddress & aAddress](#))**

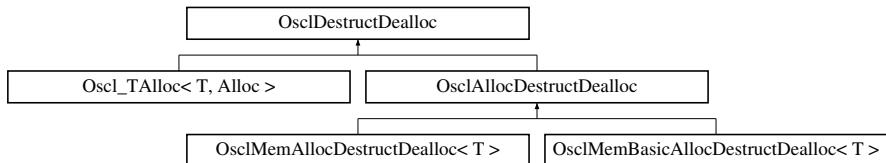
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_connect.h](#)

## 6.113 OsclDestructDealloc Class Reference

```
#include <oscl_defalloc.h>
```

Inheritance diagram for OsclDestructDealloc::



### Public Methods

- virtual void [destruct\\_and\\_dealloc \(OsclAny \\*ptr\)=0](#)

#### 6.113.1 Member Function Documentation

##### 6.113.1.1 virtual void OsclDestructDealloc::destruct\_and\_dealloc (OsclAny \*ptr) [pure virtual]

Implemented in [Oscl\\_TAlloc< T, Alloc >](#), [OsclMemAllocDestructDealloc< T >](#), [OsclMemBasicAllocDestructDealloc< T >](#), [Oscl\\_TAlloc< entry\\_type, Alloc >](#), [Oscl\\_TAlloc< node\\_type, TagTree\\_Allocator >](#), [Oscl\\_TAlloc< node\\_type, alloc\\_type >](#), [Oscl\\_TAlloc< MM\\_StatsNodeTagTreeType, OsclMemBasicAllocator >](#), [Oscl\\_TAlloc< char, alloc\\_type >](#), [Oscl\\_TAlloc< tag\\_base\\_unit, Alloc >](#), [Oscl\\_TAlloc< PVLogger, alloc\\_type >](#), and [Oscl\\_TAlloc< node\\_type, Alloc >](#).

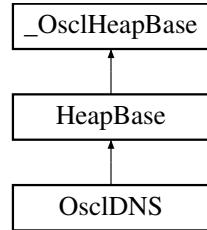
The documentation for this class was generated from the following file:

- [oscl\\_defalloc.h](#)

## 6.114 OsclDNS Class Reference

```
#include <oscl_dns.h>
```

Inheritance diagram for OsclDNS::



### Public Methods

- OSCL\_IMPORT\_REF ~OsclDNS ()
- OSCL\_IMPORT\_REF TPVDNSEvent GetHostByName (char \*name, OsclNetworkAddress &addr, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void CancelGetHostByName ()

### Static Public Methods

- OSCL\_IMPORT\_REF OsclDNS \* NewL (Oscl\_DefAlloc &alloc, OsclSocketServ &aServ, OsclDNSObserver &aObserver, uint32 aId)

### Friends

- class OsclDNSRequestAO

#### 6.114.1 Detailed Description

The DNS class

#### 6.114.2 Constructor & Destructor Documentation

##### 6.114.2.1 OSCL\_IMPORT\_REF OsclDNS::~OsclDNS ()

Destructor.

Note: the application must de-allocate the DNS object using the same allocator that was passed in the NewL object creation call.

#### 6.114.3 Member Function Documentation

##### 6.114.3.1 OSCL\_IMPORT\_REF void OsclDNS::CancelGetHostByName ()

Cancel GetHostByName

This method will cancel any pending GetHostByName operation on the current object, causing the GetHostByName to complete with error EPVDNSCancel. If there is no pending GetHostByName operation, this method will have no effect.

#### 6.114.3.2 OSCL\_IMPORT\_REF TPVDNSEvent OsclDNS::GetHostByName (char \* *name*, OsclNetworkAddress & *addr*, int32 *aTimeoutMsec* = -1)

GetHostByName. This is an asynchronous method.

**Parameters:**

*name*: Null-terminated string containing the host name.

*addr*: The output address. The ipAddr field will contain the network address of the host in dotted decimal notation.

*aTimeoutMsec*: A timeout for the request in milliseconds, or (-1) to indicate infinite wait. @returns: EPVDNSPending for success, EPVDNSFailure for failure.

#### 6.114.3.3 OSCL\_IMPORT\_REF OsclDNS\* OsclDNS::NewL (Oscl\_DefAlloc & *alloc*, OsclSocketServ & *aServ*, OsclDNSObserver & *aObserver*, uint32 *aId*) [static]

DNS object creation.

**Parameters:**

*alloc*: Memory allocator

*aServ*: Socket server.

*aObserver*: DNS Event observer

*aId*: Unique ID for this DNS object. This ID will be included in all callbacks associated with this DNS object.

### 6.114.4 Friends And Related Function Documentation

#### 6.114.4.1 friend class OsclDNSRequestAO [friend]

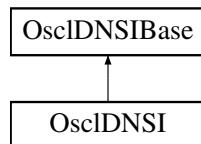
The documentation for this class was generated from the following file:

- [oscl\\_dns.h](#)

## 6.115 OsclDNSI Class Reference

```
#include <oscl_dns_imp_pv.h>
```

Inheritance diagram for OsclDNSI::



### Public Methods

- [~OsclDNSI \(\)](#)
- int32 [Open \(OsclSocketServI &aServer\)](#)
- int32 [Close \(\)](#)
- void [GetHostByName \(GetHostNameParam &, OsclDNSRequestAO &\)](#)
- void [GetHostByNameSuccess \(GetHostNameParam &\)](#)

### Static Public Methods

- OsclDNSI \* [NewL \(Oscl\\_DefAlloc &a\)](#)

### Friends

- class [OsclDNSRequest](#)
- class [DNSRequestParam](#)

### 6.115.1 Detailed Description

OsclDNSI, non-Symbian implementation

### 6.115.2 Constructor & Destructor Documentation

#### 6.115.2.1 OsclDNSI::~OsclDNSI ()

### 6.115.3 Member Function Documentation

#### 6.115.3.1 int32 OsclDNSI::Close () [virtual]

Implements [OsclDNSIBase](#).

#### 6.115.3.2 void OsclDNSI::GetHostByName (GetHostNameParam &, OsclDNSRequestAO &) [virtual]

Implements [OsclDNSIBase](#).

**6.115.3.3 void OsclDNSI::GetHostByNameSuccess ([GetHostNameParam](#) &)** [virtual]

Implements [OsclDNSIBase](#).

**6.115.3.4 OsclDNSI\* OsclDNSI::NewL ([Oscl\\_DefAlloc](#) & *a*)** [static]

**6.115.3.5 int32 OsclDNSI::Open ([OsclSocketServI](#) & *aServer*)** [virtual]

Implements [OsclDNSIBase](#).

## 6.115.4 Friends And Related Function Documentation

**6.115.4.1 friend class DNSRequestParam** [friend]

**6.115.4.2 friend class OsclDNSRequest** [friend]

Reimplemented from [OsclDNSIBase](#).

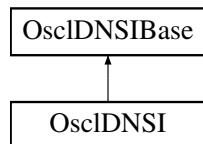
The documentation for this class was generated from the following file:

- [oscl\\_dns\\_imp\\_pv.h](#)

## 6.116 OsclDNSIBase Class Reference

```
#include <oscl_dns_imp_base.h>
```

Inheritance diagram for OsclDNSIBase::



### Public Methods

- virtual ~OsclDNSIBase ()
- virtual int32 Open (OsclSocketServI &aServer)=0
- virtual int32 Close ()=0
- virtual void GetHostByName (GetHostNameParam &, OsclDNSRequestAO &)=0
- virtual void GetHostByNameSuccess (GetHostNameParam &)=0
- void CancelFxn (TPVDNSFxn)

### Protected Methods

- OsclDNSIBase (Oscl\_DefAlloc &a)
- virtual bool IsReady (OsclDNSRequestAO &aObject)=0
- virtual void CancelGetHostName ()=0

### Protected Attributes

- Oscl\_DefAlloc & iAlloc
- OsclSocketServI \* iSocketServ

### Friends

- class OsclDNSRequest
- class OsclGetHostNameRequest

#### 6.116.1 Detailed Description

OsclDNSIBase is a common base class for all implementations.

## 6.116.2 Constructor & Destructor Documentation

**6.116.2.1** `virtual OsclDNSIBase::~OsclDNSIBase () [virtual]`

**6.116.2.2** `OsclDNSIBase::OsclDNSIBase (Oscl_DefAlloc & a) [protected]`

## 6.116.3 Member Function Documentation

**6.116.3.1** `void OsclDNSIBase::CancelFxn (TPVDNSFxn)`

**6.116.3.2** `virtual void OsclDNSIBase::CancelGetHostByName () [protected, pure virtual]`

**6.116.3.3** `virtual int32 OsclDNSIBase::Close () [pure virtual]`

Implemented in [OsclDNSI](#).

**6.116.3.4** `virtual void OsclDNSIBase::GetHostByName (GetHostNameParam &, OsclDNSRequestAO &) [pure virtual]`

Implemented in [OsclDNSI](#).

**6.116.3.5** `virtual void OsclDNSIBase::GetHostByNameSuccess (GetHostNameParam &) [pure virtual]`

Implemented in [OsclDNSI](#).

**6.116.3.6** `virtual bool OsclDNSIBase::IsReady (OsclDNSRequestAO & aObject) [protected, pure virtual]`

**6.116.3.7** `virtual int32 OsclDNSIBase::Open (OsclSocketServI & aServer) [pure virtual]`

Implemented in [OsclDNSI](#).

## 6.116.4 Friends And Related Function Documentation

**6.116.4.1** `friend class OsclDNSRequest [friend]`

Reimplemented in [OsclDNSI](#).

**6.116.4.2** `friend class OsclGetHostByNameRequest [friend]`

## 6.116.5 Field Documentation

**6.116.5.1** `Oscl_DefAlloc& OsclDNSIBase::iAlloc [protected]`

**6.116.5.2** `OsclSocketServI* OsclDNSIBase::iSocketServ [protected]`

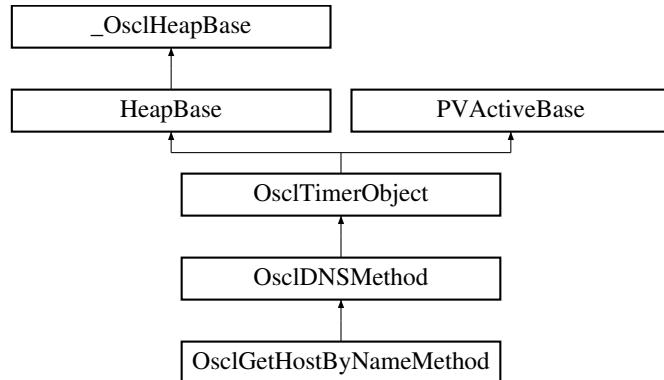
The documentation for this class was generated from the following file:

- 
- [oscl\\_dns\\_imp\\_base.h](#)

## 6.117 OsclDNSMethod Class Reference

```
#include <oscl_dns_method.h>
```

Inheritance diagram for OsclDNSMethod::



### Public Methods

- [OsclDNSMethod \(Oscl\\_DefAlloc &a, const char \\*name, TPVDNSFxn fxn\)](#)
- void [Abort \(\)](#)
- void [AbortAll \(\)](#)
- void [CancelMethod \(\)](#)
- void [Run \(\)](#)

### Data Fields

- [OsclDNSObserver \\* iDNSObserver](#)
- uint32 [iId](#)
- [Oscl\\_DefAlloc & iAlloc](#)
- [TPVDNSFxn iDNSFxn](#)
- [PVLogger \\* iLogger](#)

### Protected Methods

- void [ConstructL \(OsclDNSObserver \\*aObserver, OsclDNSRequestAO \\*aAO, uint32 aId\)](#)
- bool [StartMethod \(int32 aTimeoutMsec\)](#)
- void [MethodDone \(\)](#)

### Protected Attributes

- [OsclDNSRequestAO \\* iDNSRequestAO](#)

#### 6.117.1 Detailed Description

This is the base class for all socket methods. It provides the timeout on socket requests.

## 6.117.2 Constructor & Destructor Documentation

**6.117.2.1 OsclDNSMethod::OsclDNSMethod ([Oscl\\_DefAlloc](#) & *a*, const char \* *name*, [TPVDNSFxn](#) *fxn*) [inline]**

## 6.117.3 Member Function Documentation

**6.117.3.1 void OsclDNSMethod::Abort ()**

**6.117.3.2 void OsclDNSMethod::AbortAll ()**

**6.117.3.3 void OsclDNSMethod::CancelMethod ()**

**6.117.3.4 void OsclDNSMethod::ConstructL ([OsclDNSObserver](#) \* *aObserver*, [OsclDNSRequestAO](#) \* *aAO*, uint32 *aId*) [protected]**

**6.117.3.5 void OsclDNSMethod::MethodDone () [protected]**

**6.117.3.6 void OsclDNSMethod::Run () [virtual]**

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's `WaitForAnyRequest()` function completes.

Before calling this active object's `Run()` function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request

2. marked this active object's request as complete (i.e. the request is no longer outstanding)

`Run()` runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls `ExecError()` to handle the leave.

Note that once the active scheduler's `Start()` function has been called, all user code is run under one of the program's active object's `Run()` or `RunError()` functions.

Implements [PVActiveBase](#).

6.117.3.7 **bool OsclDNSMethod::StartMethod (int32 *aTimeoutMsec*)** [protected]

## 6.117.4 Field Documentation

6.117.4.1 **Oscl\_DefAlloc& OsclDNSMethod::iAlloc**

6.117.4.2 **TPVDNSFxn OsclDNSMethod::iDNSFxn**

6.117.4.3 **OsclDNSObserver\* OsclDNSMethod::iDNSObserver**

6.117.4.4 **OsclDNSRequestAO\* OsclDNSMethod::iDNSRequestAO** [protected]

6.117.4.5 **uint32 OsclDNSMethod::iId**

6.117.4.6 **PVLogger\* OsclDNSMethod::iLogger**

The documentation for this class was generated from the following file:

- [oscl\\_dns\\_method.h](#)

## 6.118 OsclDNSObserver Class Reference

```
#include <oscl_dns.h>
```

### Public Methods

- virtual OSCL\_IMPORT\_REF void [HandleDNSEvent](#) (int32 aId, [TPVDNSFxn](#) aFxn, [TPVDNSEvent](#) aEvent, int32 aError)=0
- virtual [~OsclDNSObserver](#) ()

### 6.118.1 Detailed Description

DNS event observer. The client implements this to get asynchronous command completion.

### 6.118.2 Constructor & Destructor Documentation

**6.118.2.1 virtual OsclDNSObserver::~OsclDNSObserver () [inline, virtual]**

### 6.118.3 Member Function Documentation

**6.118.3.1 virtual OSCL\_IMPORT\_REF void OsclDNSObserver::HandleDNSEvent (int32 *aId*,  
[TPVDNSFxn](#) *aFxn*, [TPVDNSEvent](#) *aEvent*, int32 *aError*) [pure virtual]**

DNS Event callback.

#### Parameters:

***aId*:** The ID that was supplied when the DNS object was created.

***aEvent*:** Function completion event. Will be EPVDNSSuccess, EPVDNSTimeout, or EPVDNSFailure.

***aError*:** When the event is EPVDNSFailure, this may contain a platform-specific error code, or zero if none is available.

The documentation for this class was generated from the following file:

- [oscl\\_dns.h](#)

## 6.119 OsclDNSRequest Class Reference

```
#include <oscl_dns_request.h>
```

### Public Methods

- [OsclDNSRequest \(\)](#)
- [~OsclDNSRequest \(\)](#)
- void [CancelRequest \(\)](#)
- void [Complete \(bool, int32 aStatus, int32 aSockErr\)](#)
- void [Activate \(DNSRequestParam \\*iParam, OsclDNSRequestAO &a\)](#)

### Data Fields

- [OsclDNSRequestAO \\* iDNSRequestAO](#)
- [DNSRequestParam \\* iDNSRequestParam](#)
- bool [iActive](#)

#### 6.119.1 Detailed Description

This class defines the interface to the dns implementation threads.

#### 6.119.2 Constructor & Destructor Documentation

**6.119.2.1 OsclDNSRequest::OsclDNSRequest () [inline]**

**6.119.2.2 OsclDNSRequest::~OsclDNSRequest () [inline]**

#### 6.119.3 Member Function Documentation

**6.119.3.1 void OsclDNSRequest::Activate (DNSRequestParam \* iParam, OsclDNSRequestAO & a)**

**6.119.3.2 void OsclDNSRequest::CancelRequest ()**

**6.119.3.3 void OsclDNSRequest::Complete (bool, int32 aStatus, int32 aSockErr)**

#### 6.119.4 Field Documentation

**6.119.4.1 bool OsclDNSRequest::iActive**

**6.119.4.2 OsclDNSRequestAO\* OsclDNSRequest::iDNSRequestAO**

**6.119.4.3 DNSRequestParam\* OsclDNSRequest::iDNSRequestParam**

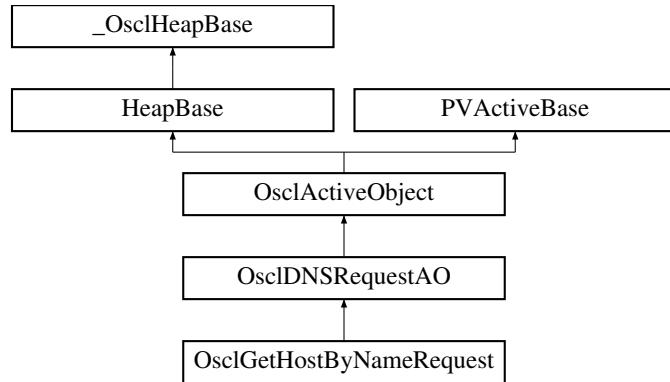
The documentation for this class was generated from the following file:

- [oscl\\_dns\\_request.h](#)

## 6.120 OsclDNSRequestAO Class Reference

```
#include <oscl_dns_method.h>
```

Inheritance diagram for OsclDNSRequestAO::



### Protected Methods

- [OsclDNSRequestAO](#) (const char \*name)
- void [ConstructL](#) ([OsclDNSI](#) \*aDNS, [OsclDNSMethod](#) \*aMethod)
- void [Abort](#) ()
- void [NewRequest](#) ()
- void [RequestDone](#) ()
- int [GetSocketError](#) ()
- [OsclSocketServI](#) \* [Serv](#) ()
- void [DoCancel](#) ()
- void [Run](#) ()
- virtual void [Success](#) ()

### Protected Attributes

- [OsclDNSI](#) \* [iDNSI](#)
- [OsclDNSMethod](#) \* [iDNSMethod](#)
- int32 [iSocketError](#)
- [PVLogger](#) \* [iLogger](#)

### Friends

- class [OsclDNSI](#)
- class [OsclDNSMethod](#)
- class [OsclDNSRequest](#)
- class [DNSRequestParam](#)

#### 6.120.1 Detailed Description

This is the base class for all requests to the socket server.

## 6.120.2 Constructor & Destructor Documentation

**6.120.2.1 OsclDNSRequestAO::OsclDNSRequestAO (const char \* *name*)** [inline, protected]

## 6.120.3 Member Function Documentation

**6.120.3.1 void OsclDNSRequestAO::Abort ()** [inline, protected]

**6.120.3.2 void OsclDNSRequestAO::ConstructL (OsclDNSI \* *aDNS*, OsclDNSMethod \* *aMethod*)** [inline, protected]

**6.120.3.3 void OsclDNSRequestAO::DoCancel ()** [protected, virtual]

Cancel request handler. This gets called by scheduler when the request is cancelled. The default routine will complete the request. If any additional action is needed, the derived class may override this. If the derived class does override DoCancel, it must complete the request.

Reimplemented from [OsclActiveObject](#).

**6.120.3.4 int OsclDNSRequestAO::GetSocketError ()** [protected]

**6.120.3.5 void OsclDNSRequestAO::NewRequest ()** [protected]

**6.120.3.6 void OsclDNSRequestAO::RequestDone ()** [protected]

**6.120.3.7 void OsclDNSRequestAO::Run ()** [protected, virtual]

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's WaitForAnyRequest() function completes.

Before calling this active object's [Run\(\)](#) function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request
2. marked this active object's request as complete (i.e. the request is no longer outstanding)

[Run\(\)](#) runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls ExecError() to handle the leave.

Note that once the active scheduler's Start() function has been called, all user code is run under one of the program's active object's [Run\(\)](#) or [RunError\(\)](#) functions.

Implements [PVActiveBase](#).

6.120.3.8 **OsclSocketServI\* OsclDNSRequestAO::Serv ()** [protected]

6.120.3.9 **virtual void OsclDNSRequestAO::Success ()** [inline, protected, virtual]

## 6.120.4 Friends And Related Function Documentation

6.120.4.1 **friend class DNSRequestParam** [friend]

6.120.4.2 **friend class OsclDNSI** [friend]

6.120.4.3 **friend class OsclDNSMethod** [friend]

6.120.4.4 **friend class OsclDNSRequest** [friend]

## 6.120.5 Field Documentation

6.120.5.1 **OsclDNSI\* OsclDNSRequestAO::iDNSI** [protected]

6.120.5.2 **OsclDNSMethod\* OsclDNSRequestAO::iDNSMethod** [protected]

6.120.5.3 **PVLogger\* OsclDNSRequestAO::iLogger** [protected]

6.120.5.4 **int32 OsclDNSRequestAO::iSocketError** [protected]

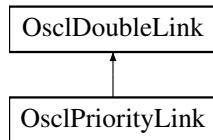
The documentation for this class was generated from the following file:

- [oscl\\_dns\\_method.h](#)

## 6.121 OsclDoubleLink Class Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OsclDoubleLink::



### Public Methods

- [OsclDoubleLink \(\)](#)
- void [Remove \(\)](#)
- void [InsertAfter \(OsclDoubleLink \\*aLink\)](#)
- void [InsertBefore \(OsclDoubleLink \\*aLink\)](#)

### Data Fields

- OsclDoubleLink \* [iNext](#)
- OsclDoubleLink \* [iPrev](#)

#### 6.121.1 Constructor & Destructor Documentation

##### 6.121.1.1 OsclDoubleLink::OsclDoubleLink () [inline]

#### 6.121.2 Member Function Documentation

##### 6.121.2.1 void OsclDoubleLink::InsertAfter (OsclDoubleLink \* *aLink*)

##### 6.121.2.2 void OsclDoubleLink::InsertBefore (OsclDoubleLink \* *aLink*)

##### 6.121.2.3 void OsclDoubleLink::Remove ()

#### 6.121.3 Field Documentation

##### 6.121.3.1 OsclDoubleLink\* OsclDoubleLink::iNext

##### 6.121.3.2 OsclDoubleLink\* OsclDoubleLink::iPrev

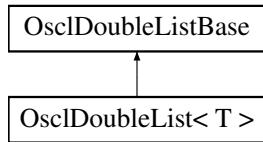
The documentation for this class was generated from the following file:

- [oscl\\_double\\_list.h](#)

## 6.122 OsclDoubleList< T > Class Template Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OsclDoubleList< T >::



### Public Methods

- OSCL\_INLINE OsclDoubleList()
- OSCL\_INLINE OsclDoubleList(int32 anOffset)
- OSCL\_INLINE void InsertHead(T &aRef)
- OSCL\_INLINE void InsertTail(T &aRef)
- OSCL\_INLINE bool IsHead(const T \*aPtr) const
- OSCL\_INLINE bool IsTail(const T \*aPtr) const
- OSCL\_INLINE T \* Head() const
- OSCL\_INLINE T \* Tail() const

```
template<class T> class OsclDoubleList< T >
```

#### 6.122.1 Constructor & Destructor Documentation

**6.122.1.1 template<class T> OSCL\_INLINE OsclDoubleList< T >::OsclDoubleList()**

**6.122.1.2 template<class T> OSCL\_INLINE OsclDoubleList< T >::OsclDoubleList(int32  
anOffset)**

#### 6.122.2 Member Function Documentation

**6.122.2.1 template<class T> OSCL\_INLINE T\* OsclDoubleList< T >::Head()**

**6.122.2.2 template<class T> OSCL\_INLINE void OsclDoubleList< T >::InsertHead(T & aRef)**

**6.122.2.3 template<class T> OSCL\_INLINE void OsclDoubleList< T >::InsertTail(T & aRef)**

**6.122.2.4 template<class T> OSCL\_INLINE bool OsclDoubleList< T >::IsHead(const T \* aPtr)  
const**

**6.122.2.5 template<class T> OSCL\_INLINE bool OsclDoubleList< T >::IsTail(const T \* aPtr)  
const**

**6.122.2.6 template<class T> OSCL\_INLINE T\* OsclDoubleList< T >::Tail()**

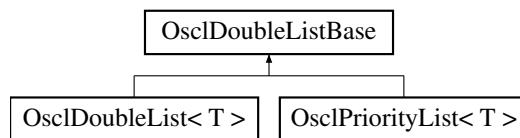
The documentation for this class was generated from the following file:

- [oscl\\_double\\_list.h](#)

## 6.123 OsclDoubleListBase Class Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OsclDoubleListBase::



### Public Methods

- bool [IsEmpty \(\) const](#)
- void [SetOffset \(int32 anOffset\)](#)
- void [Reset \(\)](#)
- [OsclDoubleLink \\* getHead \(\)](#)
- int32 [getOffset \(\)](#)

### Protected Methods

- [OsclDoubleListBase \(\)](#)
- [OsclDoubleListBase \(int32 anOffset\)](#)
- void [InsertHead \(OsclAny \\*aPtr\)](#)
- void [InsertTail \(OsclAny \\*aPtr\)](#)
- void [Insert \(OsclAny \\*aPtr\)](#)

### Protected Attributes

- [OsclDoubleLink iHead](#)
- int32 [iOffset](#)

### 6.123.1 Constructor & Destructor Documentation

6.123.1.1 **OsclDoubleListBase::OsclDoubleListBase ()** [protected]

6.123.1.2 **OsclDoubleListBase::OsclDoubleListBase (int32 *anOffset*)** [protected]

### 6.123.2 Member Function Documentation

6.123.2.1 **OsclDoubleLink\* OsclDoubleListBase::getHead ()** [inline]

6.123.2.2 **int32 OsclDoubleListBase::getOffset ()** [inline]

6.123.2.3 **void OsclDoubleListBase::Insert (OsclAny \* *aPtr*)** [protected]

6.123.2.4 **void OsclDoubleListBase::InsertHead (OsclAny \* *aPtr*)** [protected]

6.123.2.5 **void OsclDoubleListBase::InsertTail (OsclAny \* *aPtr*)** [protected]

6.123.2.6 **bool OsclDoubleListBase::IsEmpty ()**

6.123.2.7 **void OsclDoubleListBase::Reset ()**

6.123.2.8 **void OsclDoubleListBase::SetOffset (int32 *anOffset*)**

### 6.123.3 Field Documentation

6.123.3.1 **OsclDoubleLink OsclDoubleListBase::iHead** [protected]

6.123.3.2 **int32 OsclDoubleListBase::iOffset** [protected]

The documentation for this class was generated from the following file:

- [oscl\\_double\\_list.h](#)

## 6.124 OsclDoubleRunner< T > Class Template Reference

```
#include <oscl_double_list.h>
```

### Public Methods

- [OsclDoubleRunner \(OsclDoubleListBase &aQue\)](#)
- [void Set \(T &aLink\)](#)
- [operator T \\* \(\)](#)
- [T \\* operator++ \(int\)](#)
- [T \\* operator- \(int\)](#)
- [void SetToHead \(\)](#)
- [void SetToTail \(\)](#)

### Protected Attributes

- [int32 iOffset](#)
- [OsclDoubleLink \\* iHead](#)
- [OsclDoubleLink \\* iNext](#)

template<class T> class OsclDoubleRunner< T >

#### 6.124.1 Constructor & Destructor Documentation

6.124.1.1 template<class T> OsclDoubleRunner< T >::OsclDoubleRunner ([OsclDoubleListBase & aQue](#)) [inline]

#### 6.124.2 Member Function Documentation

6.124.2.1 template<class T> OsclDoubleRunner< T >::operator T \* () [inline]

6.124.2.2 template<class T> T\* OsclDoubleRunner< T >::operator++ (int) [inline]

6.124.2.3 template<class T> T\* OsclDoubleRunner< T >::operator- (int)

6.124.2.4 template<class T> void OsclDoubleRunner< T >::Set (T & aLink) [inline]

6.124.2.5 template<class T> void OsclDoubleRunner< T >::SetToHead () [inline]

6.124.2.6 template<class T> void OsclDoubleRunner< T >::SetToTail () [inline]

#### 6.124.3 Field Documentation

6.124.3.1 template<class T> [OsclDoubleLink\\* OsclDoubleRunner< T >::iHead](#) [protected]

6.124.3.2 template<class T> [OsclDoubleLink\\* OsclDoubleRunner< T >::iNext](#) [protected]

6.124.3.3 template<class T> int32 [OsclDoubleRunner< T >::iOffset](#) [protected]

The documentation for this class was generated from the following file:

- 
- [oscl\\_double\\_list.h](#)

## 6.125 OsclError Class Reference

```
#include <oscl_error.h>
```

### Static Public Methods

- OSCL\_IMPORT\_REF void [PushL \(\\_OsclHeapBase \\*aPtr\)](#)
- OSCL\_IMPORT\_REF void [PushL \(OsclAny \\*aPtr\)](#)
- OSCL\_IMPORT\_REF void [PushL \(OsclTrapItem anItem\)](#)
- OSCL\_IMPORT\_REF void [Pop \(\)](#)
- OSCL\_IMPORT\_REF void [Pop \(int32 aCount\)](#)
- OSCL\_IMPORT\_REF void [PopDealloc \(\)](#)
- OSCL\_IMPORT\_REF void [PopDealloc \(int32 aCount\)](#)
- OSCL\_IMPORT\_REF void [Leave \(int32 aReason\)](#)
- OSCL\_IMPORT\_REF void [LeaveIfNull \(OsclAny \\*a\)](#)
- OSCL\_IMPORT\_REF void [LeaveIfError \(int32 aReason\)](#)

### 6.125.1 Detailed Description

User Error class

### 6.125.2 Member Function Documentation

#### 6.125.2.1 OSCL\_IMPORT\_REF void OsclError::Leave (int32 *aReason*) [static]

Do a Leave error, with the given reason code. When a leave occurs, all items on the cleanup stack for the current trap level will be destroyed, and execution will jump to the trap handler.

#### 6.125.2.2 OSCL\_IMPORT\_REF void OsclError::LeaveIfError (int32 *aReason*) [static]

Evaluate the input parameter, and if it is an error code (non-zero), then do a Leave with the provided reason code.

#### 6.125.2.3 OSCL\_IMPORT\_REF void OsclError::LeaveIfNull (OsclAny \* *a*) [static]

Evaluate the input parameter, and if it is null, do a Leave with OsclErrNoMemory reason code.

#### 6.125.2.4 OSCL\_IMPORT\_REF void OsclError::Pop (int32 *aCount*) [static]

Pop the cleanup stack N times

#### 6.125.2.5 OSCL\_IMPORT\_REF void OsclError::Pop () [static]

Pop the cleanup stack

**6.125.2.6 OSCL\_IMPORT\_REF void OsclError::PopDealloc (int32 *aCount*) [static]**

PopDealloc N times

**6.125.2.7 OSCL\_IMPORT\_REF void OsclError::PopDealloc () [static]**

Destroy the item on the top of the cleanup stack and pop it

**6.125.2.8 OSCL\_IMPORT\_REF void OsclError::PushL ([OsclTrapItem](#) *anItem*) [static]**

Push an [OsclTrapItem](#) onto the cleanup stack

**6.125.2.9 OSCL\_IMPORT\_REF void OsclError::PushL ([OsclAny](#) \* *aPtr*) [static]**

Push an OsclAny item onto the cleanup stack.

**6.125.2.10 OSCL\_IMPORT\_REF void OsclError::PushL ([\\_OsclHeapBase](#) \* *aPtr*) [static]**

Push an [\\_OsclHeapBase](#) item onto the cleanup stack.

The documentation for this class was generated from the following file:

- [oscl\\_error.h](#)

## 6.126 OsclErrorAllocator Class Reference

This class provides static methods to invoke the user defined memory allocation routines.

```
#include <oscl_error_allocator.h>
```

### Public Methods

- **OsclErrorAllocator (Oscl\_DefAlloc \*allocator)**  
*constructor method*
- **void \* operator new (uint32 size, OsclAny \*aPtr)**  
*placement new operator that allocates memory using the user defined methods*
- **void operator delete (OsclAny \*aPtr, OsclAny \*aPtr2)**  
*delete operator that doesn't do anything, user has to deallocate manually*

### Static Public Methods

- **OsclAny \* allocate (uint32 aSize)**  
*static method to allocate a block of memory on heap*
- **OsclAny deallocate (OsclAny \*aPointer)**  
*static method to deallocate a block of memory on heap*

### 6.126.1 Detailed Description

This class provides static methods to invoke the user defined memory allocation routines.

This class must be instantiated before the static methods are called, else asserts will happen

### 6.126.2 Constructor & Destructor Documentation

#### 6.126.2.1 OsclErrorAllocator::OsclErrorAllocator (**Oscl\_DefAlloc \* allocator**) [inline]

constructor method

**Parameters:**

*allocator* - a pointer to the concrete object that provides the allocator/deallocator

### 6.126.3 Member Function Documentation

#### 6.126.3.1 OsclAny\* OsclErrorAllocator::allocate (uint32 *aSize*) [inline, static]

static method to allocate a block of memory on heap

**Parameters:**

*aSize* - number of bytes to allocate

**6.126.3.2 OsclAny OsclErrorAllocator::deallocate (OsclAny \* aPointer) [inline, static]**

static method to deallocate a block of memory on heap

**Parameters:**

*aPointer* - pointer to block of memory to be deallocated

**6.126.3.3 void OsclErrorAllocator::operator delete (OsclAny \* aPtr, OsclAny \* aPtr2) [inline]**

delete operator that doesn't do anything, user has to deallocate manually

**6.126.3.4 void\* OsclErrorAllocator::operator new (uint32 size, OsclAny \* aPtr) [inline]**

placement new operator that allocates memory using the user defined methods

The documentation for this class was generated from the following file:

- [oscl\\_error\\_allocator.h](#)

## 6.127 OsclErrorTrap Class Reference

```
#include <oscl_error.h>
```

### Static Public Methods

- OSCL\_IMPORT\_REF int32 [Init \(Oscl\\_DefAlloc \\*aAlloc=NULL\)](#)
- OSCL\_IMPORT\_REF int32 [Cleanup \(\)](#)
- OSCL\_IMPORT\_REF [OsclErrorTrapImp \\* GetErrorTrapImp \(\)](#)

#### 6.127.1 Member Function Documentation

##### 6.127.1.1 OSCL\_IMPORT\_REF int32 OsclErrorTrap::Cleanup () [static]

Cleanup and destroy error trap for the calling thread.

###### Returns:

0 for success, or an error

##### 6.127.1.2 OSCL\_IMPORT\_REF [OsclErrorTrapImp\\*](#) OsclErrorTrap::GetErrorTrapImp () [static]

Get the ErrorTrapImp for the current thread. Leaves on error.

##### 6.127.1.3 OSCL\_IMPORT\_REF int32 OsclErrorTrap::Init ([Oscl\\_DefAlloc \\* aAlloc = NULL](#)) [static]

Allocate and initialize error trap for the calling thread.

###### Parameters:

*aAlloc*: optional, allocator to use for the internal implementation.

###### Returns:

0 for success, or an error

The documentation for this class was generated from the following file:

- [oscl\\_error.h](#)

## 6.128 OsclErrorTrapImp Class Reference

```
#include <oscl_error_trapcleanup.h>
```

### Public Methods

- OSCL\_IMPORT\_REF void [UnTrap \(\)](#)

### Static Public Methods

- OSCL\_IMPORT\_REF OsclErrorTrapImp \* [Trap \(\)](#)
- OSCL\_IMPORT\_REF OsclErrorTrapImp \* [TrapNoTls \(OsclErrorTrapImp \\*\)](#)

### Data Fields

- [OsclJump \\* iJumpData](#)
- int32 [iLeave](#)
- [OsclTrapStack \\* iTrapStack](#)

### Friends

- class [OsclErrorTrap](#)
- class [OsclError](#)
- class [OsclExecScheduler](#)
- class [OsclExecSchedulerCommonBase](#)
- class [OsclJump](#)
- class [OsclJumpMark](#)
- class [OsclTrapStack](#)
- class [CPVInterfaceProxy](#)
- class [OsclScheduler](#)

### 6.128.1 Detailed Description

A per-thread cleanup stack with nested trap support.

### 6.128.2 Member Function Documentation

#### 6.128.2.1 OSCL\_IMPORT\_REF OsclErrorTrapImp\* OsclErrorTrapImp::Trap () [static]

PV trap cleanup. Public for use in macros only.

#### 6.128.2.2 OSCL\_IMPORT\_REF OsclErrorTrapImp\* OsclErrorTrapImp::TrapNoTls (OsclErrorTrapImp \*) [static]

#### 6.128.2.3 OSCL\_IMPORT\_REF void OsclErrorTrapImp::UnTrap ()

these are used in public macros, but aren't intended as public methods or members.

### 6.128.3 Friends And Related Function Documentation

6.128.3.1 **friend class CPVInterfaceProxy [friend]**

6.128.3.2 **friend class OsclError [friend]**

6.128.3.3 **friend class OsclErrorTrap [friend]**

6.128.3.4 **friend class OsclExecScheduler [friend]**

6.128.3.5 **friend class OsclExecSchedulerCommonBase [friend]**

6.128.3.6 **friend class OsclJump [friend]**

6.128.3.7 **friend class OsclJumpMark [friend]**

6.128.3.8 **friend class OsclScheduler [friend]**

6.128.3.9 **friend class OsclTrapStack [friend]**

### 6.128.4 Field Documentation

6.128.4.1 **OsclJump\* OsclErrorTrapImp::iJumpData**

6.128.4.2 **int32 OsclErrorTrapImp::iLeave**

6.128.4.3 **OsclTrapStack\* OsclErrorTrapImp::iTrapStack**

The documentation for this class was generated from the following file:

- [oscl\\_error\\_trapcleanup.h](#)

## 6.129 OsclException< LeaveCode > Class Template Reference

`oscl_exception.h` contains all the exception handling macros and classes This template class provides the base exception class that all exceptions derive from

```
#include <oscl_exception.h>
```

### Public Methods

- `OsclException ()`

### Static Public Methods

- `int getLeaveCode ()`

#### 6.129.1 Detailed Description

`template<int LeaveCode> class OsclException< LeaveCode >`

`oscl_exception.h` contains all the exception handling macros and classes This template class provides the base exception class that all exceptions derive from

All PacketVideo exception classes will be derived from the OsclException class. Each derived class will have a static function where the leave code can be obtained. This avoids the issue of having static members in a DLL. The function needs to be static so it can be called without an instance of the class

#### 6.129.2 Constructor & Destructor Documentation

**6.129.2.1 `template<int LeaveCode> OsclException< LeaveCode >::OsclException ()`**  
[inline]

#### 6.129.3 Member Function Documentation

**6.129.3.1 `template<int LeaveCode> int OsclException< LeaveCode >::getLeaveCode ()`**  
[inline, static]

The documentation for this class was generated from the following file:

- `oscl_exception.h`

## 6.130 OsclExclusiveArrayPtr< T > Class Template Reference

The OsclExclusiveArrayPtr class is a template class that defines an array pointer like object intended to be assigned an address obtained (directly or indirectly) by new. When the OsclExclusiveArrayPtr expires, its destructor uses delete to free the memory.

```
#include <oscl_exclusive_ptr.h>
```

### Public Methods

- **OsclExclusiveArrayPtr (T \*inPtr=0)**  
*Default constructor Initializes the pointer and takes ownership.*
- **OsclExclusiveArrayPtr (OsclExclusiveArrayPtr< T > &\_Y)**  
*Copy constructor.*
- **OsclExclusiveArrayPtr< T > & operator= (OsclExclusiveArrayPtr< T > &\_Y)**  
*Assignment operator from an another OsclExclusiveArrayPtr.*
- **virtual ~OsclExclusiveArrayPtr ()**  
*Destructor.*
- **T & operator\* () const**  
*The indirection operator (\*) accesses a value indirectly, through a pointer.*
- **T \* operator-> () const**  
*The indirection operator (->) accesses a value indirectly, through a pointer.*
- **T \* get () const**  
*get() method returns the pointer, currently owned by the class.*
- **T \* release ()**  
*release() method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.*
- **bool set (T \*ptr)**  
*set() method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.*

### Protected Attributes

- **T \* \_Ptr**

#### 6.130.1 Detailed Description

**template<class T> class OsclExclusiveArrayPtr< T >**

The OsclExclusiveArrayPtr class is a template class that defines an array pointer like object intended to be assigned an address obtained (directly or indirectly) by new. When the OsclExclusiveArrayPtr expires, its destructor uses delete to free the memory.

The purpose of this class is to provide a way to prevent accidental memory leaks in a class or a method, due to "not remembering to delete" variables allocated on the heap. Thus if you assign an address returned by new to an [OsclExclusivePtr](#) object, you don't have to remember to free the memory later, it will be freed automatically when the object goes out of scope. The [OsclExclusivePtr](#) is an example of a smart pointer, an object that acts like a pointer, but with additional features. The class is defined so that it acts like a regular pointer in most respects

### 6.130.2 Constructor & Destructor Documentation

**6.130.2.1 template<class T> OsclExclusiveArrayPtr< T >::OsclExclusiveArrayPtr (T \* *inPtr* = 0) [inline, explicit]**

Default constructor Initializes the pointer and takes ownership.

**6.130.2.2 template<class T> OsclExclusiveArrayPtr< T >::OsclExclusiveArrayPtr (OsclExclusiveArrayPtr< T > & *\_Y*) [inline]**

Copy constructor.

Initializes the pointer and takes ownership from another OsclExclusiveArrayPtr. Note that the other class does NOT own the pointer any longer, and hence it is NOT its responsibility to free it.

**6.130.2.3 template<class T> virtual OsclExclusiveArrayPtr< T >::~OsclExclusiveArrayPtr () [inline, virtual]**

Destructor.

The pointer is deleted in case this class still has ownership

### 6.130.3 Member Function Documentation

**6.130.3.1 template<class T> T\* OsclExclusiveArrayPtr< T >::get () const [inline]**

[get\(\)](#) method returns the pointer, currently owned by the class.

**6.130.3.2 template<class T> T& OsclExclusiveArrayPtr< T >::operator \* () const [inline]**

The indirection operator (\*) accesses a value indirectly, through a pointer.

This operator ensures that the OsclExclusiveArrayPtr can be used like the regular pointer that it was initialized with.

**6.130.3.3 template<class T> T\* OsclExclusiveArrayPtr< T >::operator -> () const [inline]**

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the OsclExclusiveArrayPtr can be used like the regular pointer that it was initialized with.

**6.130.3.4 template<class T> OsclExclusiveArrayPtr<T>& OsclExclusiveArrayPtr< T >::operator= (OsclExclusiveArrayPtr< T > & \_Y) [inline]**

Assignment operator from an another OsclExclusiveArrayPtr.

**Parameters:**

*\_Y* The value parameter should be another OsclExclusiveArrayPtr

**Returns:**

Returns a reference to this OsclExclusiveArrayPtr instance with pointer initialized.

**Precondition:**

The input class should be non-null and should point to a valid pointer.

This assignment operator initializes the class to the contents of the OsclExclusiveArrayPtr given as the input parameter. The ownership of the pointer is transferred.

**6.130.3.5 template<class T> T\* OsclExclusiveArrayPtr< T >::release () [inline]**

[release\(\)](#) method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.

**6.130.3.6 template<class T> bool OsclExclusiveArrayPtr< T >::set (T \*ptr) [inline]**

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

## 6.130.4 Field Documentation

**6.130.4.1 template<class T> T\* OsclExclusiveArrayPtr< T >::\_Ptr [protected]**

The documentation for this class was generated from the following file:

- [oscl\\_exclusive\\_ptr.h](#)

## 6.131 OsclExclusivePtr< T > Class Template Reference

The OsclExclusivePtr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or indirectly) by new. When the OsclExclusivePtr expires, its destructor uses delete to free the memory.

```
#include <oscl_exclusive_ptr.h>
```

### Public Methods

- **OsclExclusivePtr** (T \*inPtr=0)  
*Default constructor Initializes the pointer and takes ownership.*
- **OsclExclusivePtr** (OsclExclusivePtr< T > &\_Y)  
*Copy constructor.*
- OsclExclusivePtr< T > & **operator=** (OsclExclusivePtr< T > &\_Y)  
*Assignment operator from an another OsclExclusivePtr.*
- virtual ~**OsclExclusivePtr** ()  
*Destructor.*
- T & **operator \*** () const  
*The indirection operator (\*) accesses a value indirectly, through a pointer.*
- T \* **operator ->** () const  
*The indirection operator (->) accesses a value indirectly, through a pointer.*
- T \* **get** () const  
***get()** method returns the pointer, currently owned by the class.*
- T \* **release** ()  
***release()** method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.*
- bool **set** (T \*ptr)  
***set()** method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.*

### Protected Attributes

- T \* **\_Ptr**

#### 6.131.1 Detailed Description

**template<class T> class OsclExclusivePtr< T >**

The OsclExclusivePtr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or indirectly) by new. When the OsclExclusivePtr expires, its destructor uses delete to free the memory.

The purpose of this class is to provide a way to prevent accidental memory leaks in a class or a method, due to "not remembering to delete" variables allocated on the heap. Thus if you assign an address returned by new to an OsclExclusivePtr object, you don't have to remember to free the memory later, it will be freed automatically when the object goes out of scope. The OsclExclusivePtr is an example of a smart pointer, an object that acts like a pointer, but with additional features. The class is defined so that it acts like a regular pointer in most respects

### **6.131.2 Constructor & Destructor Documentation**

**6.131.2.1 template<class T> OsclExclusivePtr< T >::OsclExclusivePtr (T \* *inPtr* = 0) [inline, explicit]**

Default constructor Initializes the pointer and takes ownership.

**6.131.2.2 template<class T> OsclExclusivePtr< T >::OsclExclusivePtr (OsclExclusivePtr< T > & *\_Y*) [inline]**

Copy constructor.

Initializes the pointer and takes ownership from another OsclExclusivePtr. Note that the other class does NOT own the pointer any longer, and hence it is NOT its responsibility to free it.

**6.131.2.3 template<class T> virtual OsclExclusivePtr< T >::~OsclExclusivePtr () [inline, virtual]**

Destructor.

The pointer is deleted in case this class still has ownership

### **6.131.3 Member Function Documentation**

**6.131.3.1 template<class T> T\* OsclExclusivePtr< T >::get () const [inline]**

[get\(\)](#) method returns the pointer, currently owned by the class.

**6.131.3.2 template<class T> T& OsclExclusivePtr< T >::operator \* () const [inline]**

The indirection operator (\*) accesses a value indirectly, through a pointer.

This operator ensures that the OsclExclusivePtr can be used like the regular pointer that it was initialized with.

**6.131.3.3 template<class T> T\* OsclExclusivePtr< T >::operator -> () const [inline]**

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the OsclExclusivePtr can be used like the regular pointer that it was initialized with.

**6.131.3.4 template<class T> OsclExclusivePtr<T>& OsclExclusivePtr< T >::operator= (OsclExclusivePtr< T > & \_Y) [inline]**

Assignment operator from an another OsclExclusivePtr.

**Parameters:**

*\_Y* The value parameter should be another OsclExclusivePtr

**Returns:**

Returns a reference to this OsclExclusivePtr instance with pointer initialized.

**Precondition:**

The input class should be non-null and should point to a valid pointer.

This assignment operator initializes the class to the contents of the OsclExclusivePtr given as the input parameter. The ownership of the pointer is transferred.

**6.131.3.5 template<class T> T\* OsclExclusivePtr< T >::release () [inline]**

[release\(\)](#) method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.

**6.131.3.6 template<class T> bool OsclExclusivePtr< T >::set (T \*ptr) [inline]**

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

## 6.131.4 Field Documentation

**6.131.4.1 template<class T> T\* OsclExclusivePtr< T >::\_Ptr [protected]**

The documentation for this class was generated from the following file:

- [oscl\\_exclusive\\_ptr.h](#)

## 6.132 OsclExclusivePtrA< T, Alloc > Class Template Reference

The OsclExclusivePtrA class is a template class that defines any pointer like object intended to be assigned an address obtained (directly or indirectly) through Alloc. When the OsclExclusivePtrA expires, Alloc is used to free the memory.

```
#include <oscl_exclusive_ptr.h>
```

### Public Methods

- **OsclExclusivePtrA** (T \*inPtr=0)  
*Default constructor Initializes the pointer and takes ownership.*
- **OsclExclusivePtrA** (OsclExclusivePtrA< T, Alloc > &\_Y)  
*Copy constructor.*
- OsclExclusivePtrA< T, Alloc > & **operator=** (OsclExclusivePtrA< T, Alloc > &\_Y)  
*Assignment operator from an another OsclExclusiveArrayPtr.*
- virtual ~**OsclExclusivePtrA** ()  
*Destructor.*
- T & **operator \*** () const  
*The indirection operator (\*) accesses a value indirectly, through a pointer.*
- T \* **operator ->** () const  
*The indirection operator (->) accesses a value indirectly, through a pointer.*
- T \* **get** () const  
***get()** method returns the pointer, currently owned by the class.*
- T \* **release** ()  
***release()** method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.*
- bool **set** (T \*ptr)  
***set()** method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.*

### Protected Attributes

- T \* **\_Ptr**

#### 6.132.1 Detailed Description

**template<class T, class Alloc> class OsclExclusivePtrA< T, Alloc >**

The OsclExclusivePtrA class is a template class that defines any pointer like object intended to be assigned an address obtained (directly or indirectly) through Alloc. When the OsclExclusivePtrA expires, Alloc is used to free the memory.

The purpose of this class is to provide a way to prevent accidental memory leaks in a class or a method, due to "not remembering to delete" variables allocated on the heap. Thus if you assign an address returned by new to an [OsclExclusivePtr](#) object, you don't have to remember to free the memory later, it will be freed automatically when the object goes out of scope. The [OsclExclusivePtr](#) is an example of a smart pointer, an object that acts like a pointer, but with additional features. The class is defined so that it acts like a regular pointer in most respects

### 6.132.2 Constructor & Destructor Documentation

**6.132.2.1 template<class T, class Alloc> OsclExclusivePtrA< T, Alloc >::OsclExclusivePtrA (T \* *inPtr* = 0) [inline, explicit]**

Default constructor Initializes the pointer and takes ownership.

**6.132.2.2 template<class T, class Alloc> OsclExclusivePtrA< T, Alloc >::OsclExclusivePtrA (OsclExclusivePtrA< T, Alloc > & *\_Y*) [inline]**

Copy constructor.

Initializes the pointer and takes ownership from another [OsclExclusiveArrayPtr](#). Note that the other class does NOT own the pointer any longer, and hence it is NOT its responsibility to free it.

**6.132.2.3 template<class T, class Alloc> virtual OsclExclusivePtrA< T, Alloc >::~OsclExclusivePtrA () [inline, virtual]**

Destructor.

The pointer is deleted in case this class still has ownership

### 6.132.3 Member Function Documentation

**6.132.3.1 template<class T, class Alloc> T\* OsclExclusivePtrA< T, Alloc >::get () const [inline]**

[get\(\)](#) method returns the pointer, currently owned by the class.

**6.132.3.2 template<class T, class Alloc> T& OsclExclusivePtrA< T, Alloc >::operator \* () const [inline]**

The indirection operator (\*) accesses a value indirectly, through a pointer.

This operator ensures that the [OsclExclusiveArrayPtr](#) can be used like the regular pointer that it was initialized with.

**6.132.3.3 template<class T, class Alloc> T\* OsclExclusivePtrA< T, Alloc >::operator -> () const [inline]**

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the [OsclExclusiveArrayPtr](#) can be used like the regular pointer that it was initialized with.

**6.132.3.4 template<class T, class Alloc> OsclExclusivePtrA<T, Alloc>& OsclExclusivePtrA< T, Alloc >::operator= (OsclExclusivePtrA< T, Alloc > & \_Y) [inline]**

Assignment operator from an another [OsclExclusiveArrayPtr](#).

**Parameters:**

*\_Y* The value parameter should be another [OsclExclusiveArrayPtr](#)

**Returns:**

Returns a reference to this [OsclExclusiveArrayPtr](#) instance with pointer initialized.

**Precondition:**

The input class should be non-null and should point to a valid pointer.

This assignment operator initializes the class to the contents of the [OsclExclusiveArrayPtr](#) given as the input parameter. The ownership of the pointer is transferred.

**6.132.3.5 template<class T, class Alloc> T\* OsclExclusivePtrA< T, Alloc >::release () [inline]**

[release\(\)](#) method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.

**6.132.3.6 template<class T, class Alloc> bool OsclExclusivePtrA< T, Alloc >::set (T \*ptr) [inline]**

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

## 6.132.4 Field Documentation

**6.132.4.1 template<class T, class Alloc> T\* OsclExclusivePtrA< T, Alloc >::\_Ptr [protected]**

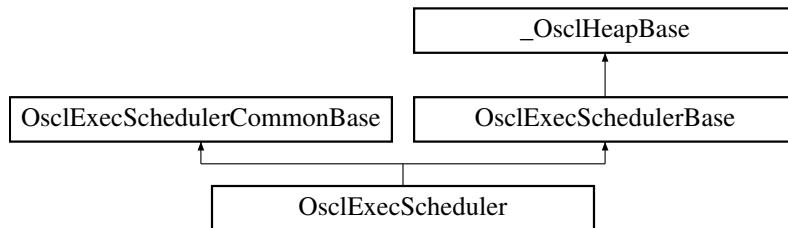
The documentation for this class was generated from the following file:

- [oscl\\_exclusive\\_ptr.h](#)

## 6.133 OsclExecScheduler Class Reference

```
#include <oscl_scheduler.h>
```

Inheritance diagram for OsclExecScheduler::



### Public Methods

- OSCL\_IMPORT\_REF void [RunSchedulerNonBlocking](#) (int32 aTargetCount, int32 &aReady, uint32 &aDelayMsec)
- OSCL\_IMPORT\_REF void [RegisterForCallback](#) ([OsclSchedulerObserver](#) \*aCallback, [OsclAny](#) \*aCallbackContext)

### Static Public Methods

- OSCL\_IMPORT\_REF [OsclExecScheduler](#) \* [Current](#) ()

### Friends

- class [OsclScheduler](#)

#### 6.133.1 Member Function Documentation

##### 6.133.1.1 OSCL\_IMPORT\_REF OsclExecScheduler\* OsclExecScheduler::Current () [static]

Get currently installed scheduler for calling thread, or NULL if no scheduler is installed.

##### 6.133.1.2 OSCL\_IMPORT\_REF void OsclExecScheduler::RegisterForCallback ([OsclSchedulerObserver](#) \* aCallback, [OsclAny](#) \* aCallbackContext)

Register for a notification when non-blocking scheduler needs to run again.

Note: On Symbian, non-blocking mode is not supported and this call will leave.

##### 6.133.1.3 OSCL\_IMPORT\_REF void OsclExecScheduler::RunSchedulerNonBlocking (int32 aTargetCount, int32 &aReady, uint32 &aDelayMsec)

Run PV scheduler in non-blocking mode. This call returns when the desired number of Run calls have been made, or when there are no more active objects that are ready to run.

**Parameters:**

*aTargetCount*: (input param) the maximum number of Run calls to make.

*aReady*: (output param) tells the number of active objects that are currently ready to run.

*aDelayMsec*: (output param) If no active objects are ready to run, but one or more active objects are waiting on timers, this parameter will tell the time interval from the current time until the first of the pending timer objects will be ready to run, in milliseconds.

Note: On Symbian, non-blocking mode is not supported and this call will leave.

## 6.133.2 Friends And Related Function Documentation

### 6.133.2.1 friend class OsclScheduler [friend]

Reimplemented from [OsclExecSchedulerCommonBase](#).

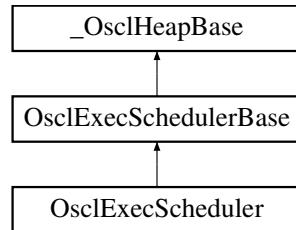
The documentation for this class was generated from the following file:

- [oscl\\_scheduler.h](#)

## 6.134 OsclExecSchedulerBase Class Reference

```
#include <oscl_scheduler_types.h>
```

Inheritance diagram for OsclExecSchedulerBase::



### Friends

- class [OsclExecScheduler](#)
- class [OsclCoeActiveScheduler](#)
- class [PVActiveBase](#)

#### 6.134.1 Detailed Description

OsclActiveSchedulerBase is the base for [OsclExecScheduler](#). The non-Symbian OsclActiveSchedulerBase class is functionally similar to a subset of Symbian CActiveScheduler class.

#### 6.134.2 Friends And Related Function Documentation

**6.134.2.1 friend class OsclCoeActiveScheduler [friend]**

**6.134.2.2 friend class OsclExecScheduler [friend]**

**6.134.2.3 friend class PVActiveBase [friend]**

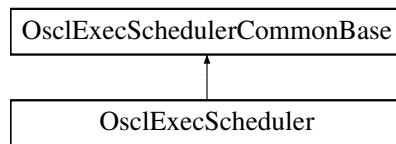
The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_types.h](#)

## 6.135 OsclExecSchedulerCommonBase Class Reference

```
#include <oscl_scheduler.h>
```

Inheritance diagram for OsclExecSchedulerCommonBase::



### Public Methods

- OSCL\_IMPORT\_REF void [StartScheduler](#) (OsclSemaphore \*sem=NULL)
- OSCL\_IMPORT\_REF void [StopScheduler](#) ()
- OSCL\_IMPORT\_REF void [SuspendScheduler](#) ()
- OSCL\_IMPORT\_REF void [ResumeScheduler](#) ()
- OSCL\_IMPORT\_REF void [StartNativeScheduler](#) ()

### Static Public Methods

- OSCL\_IMPORT\_REF [OsclNameString< PVSCHEDNAMELEN > \\*](#) [GetName](#) ()
- OSCL\_IMPORT\_REF uint32 [GetId](#) ()

### Protected Types

- enum [TOtherExecStats](#) { [EOtherExecStats\\_WaitTime](#), [EOtherExecStats\\_QueueTime](#), [EOtherExecStats\\_NativeOS](#), [EOtherExecStats\\_ReleaseTime](#), [EOtherExecStats\\_Last](#) }

### Protected Methods

- virtual ~[OsclExecSchedulerCommonBase](#) ()
- void [InstallScheduler](#) ()
- void [UninstallScheduler](#) ()
- void [Error](#) (int32 anError) const
- [OsclExecSchedulerCommonBase](#) ([Oscl\\_DefAlloc](#) \*)
- virtual void [ConstructL](#) (const char \*name, int)
- void [BeginScheduling](#) (bool blocking, bool native)
- void [EndScheduling](#) ()
- void [BlockingLoopL](#) ()
- bool [IsStarted](#) ()
- bool [IsInstalled](#) ()
- void [AddToExecTimerQ](#) ([PVActiveBase](#) \*active, uint32)
- void [PendComplete](#) ([PVActiveBase](#) \*, int32 aReason, [TPVThreadContext](#) aContext)
- void [RequestCanceled](#) ([PVActiveBase](#) \*)
- [PVActiveBase](#) \* [UpdateTimers](#) (uint32 &aDelay)
- [PVActiveBase](#) \* [UpdateTimersMsec](#) (uint32 &aDelay)
- [PVActiveBase](#) \* [WaitForReadyAO](#) ()

- void [CallRunExec \(PVActiveBase \\*\)](#)
- void [ConstructStatQ \(\)](#)
- void [BeginStats \(\)](#)
- void [EndStats \(\)](#)
- void [CleanupStatQ \(\)](#)
- PVActiveBase \* [FindPVBase \(PVActiveBase \\*active, OsclDoubleList< PVActiveBase > &\)](#)
- void [CleanupExecQ \(\)](#)
- void [InitExecQ \(int\)](#)
- void [ResetLogPerf \(\)](#)
- void [IncLogPerf \(uint32\)](#)

## Static Protected Methods

- OsclExecSchedulerCommonBase \* [GetScheduler \(\)](#)
- OsclExecSchedulerCommonBase \* [SetScheduler \(OsclExecSchedulerCommonBase \\*\)](#)
- void [ShowStats \(PVActiveStats \\*active\)](#)
- void [ShowSummaryStats \(PVActiveStats \\*active, PVLogger \\*, int64, int64 &, float &\)](#)

## Protected Attributes

- bool [iBlockingMode](#)
- bool [iNativeMode](#)
- PVSchedulerStopper \* [iStopper](#)
- OsclNoYieldMutex [iStopperCrit](#)
- PVThreadContext [iThreadContext](#)
- OsclNameString< PVSCHEDNAMELEN > [iName](#)
- bool [iDoStop](#)
- bool [iDoSuspend](#)
- bool [iSuspended](#)
- OsclSemaphore [iResumeSem](#)
- OsclErrorTrapImp \* [iErrorTrapImp](#)
- OsclReadyQ [iReadyQ](#)
- OsclTimerQ [iExecTimerQ](#)
- uint32 [iNumAOAdded](#)
- OsclDoubleList< PVActiveStats > [iPVStatQ](#)
- PVActiveStats \* [iOtherExecStats \[EOtherExecStats\\_Last\]](#)
- uint8 \* [iTTotalTicksTemp](#)
- int64 [iGrandTotalTicks](#)
- float [iTTotalPercent](#)
- uint32 [iTime](#)
- int32 [iDelta](#)
- PVActiveStats \* [iPVStats](#)
- PVLogger \* [iLogger](#)
- PVLogger \* [iDebugLogger](#)
- char \* [iLogPerfIndentStr](#)
- int32 [iLogPerfIndentStrLen](#)
- uint32 [iLogPerfTotal](#)
- Oscl\_DefAlloc \* [iAlloc](#)
- OsclMemAllocator [iDefAlloc](#)

## Static Protected Attributes

- const uint32 [iTTimeCompareThreshold](#)

## Friends

- class [OsclScheduler](#)
- class [PVThreadContext](#)
- class [OsclCoeActiveScheduler](#)
- class [OsclTimerCompare](#)
- class [OsclReadyQ](#)
- class [OsclError](#)
- class [PVActiveStats](#)
- class [OsclActiveObject](#)
- class [OsclTimerObject](#)
- class [PVActiveBase](#)
- class [PVSchedulerStopper](#)
- class [OsclExecScheduler](#)

## 6.135.1 Member Enumeration Documentation

### 6.135.1.1 enum OsclExecSchedulerCommonBase::TOtherExecStats [protected]

Enumeration values:

- EOtherExecStats\_WaitTime**
- EOtherExecStats\_QueueTime**
- EOtherExecStats\_NativeOS**
- EOtherExecStats\_ReleaseTime**
- EOtherExecStats\_Last**

## 6.135.2 Constructor & Destructor Documentation

- 6.135.2.1 **virtual OsclExecSchedulerCommonBase::~OsclExecSchedulerCommonBase ()**  
[protected, virtual]
- 6.135.2.2 **OsclExecSchedulerCommonBase::OsclExecSchedulerCommonBase (Oscl\_DefAlloc \*)**  
[protected]

## 6.135.3 Member Function Documentation

- 6.135.3.1 **void OsclExecSchedulerCommonBase::AddToExecTimerQ (PVActiveBase \* *active*, uint32)** [protected]
- 6.135.3.2 **void OsclExecSchedulerCommonBase::BeginScheduling (bool *blocking*, bool *native*)**  
[protected]
- 6.135.3.3 **void OsclExecSchedulerCommonBase::BeginStats ()** [protected]
- 6.135.3.4 **void OsclExecSchedulerCommonBase::BlockingLoopL ()** [protected]
- 6.135.3.5 **void OsclExecSchedulerCommonBase::CallRunExec (PVActiveBase \*)** [protected]
- 6.135.3.6 **void OsclExecSchedulerCommonBase::CleanupExecQ ()** [protected]
- 6.135.3.7 **void OsclExecSchedulerCommonBase::CleanupStatQ ()** [protected]
- 6.135.3.8 **virtual void OsclExecSchedulerCommonBase::ConstructL (const char \* *name*, int)**  
[protected, virtual]
- 6.135.3.9 **void OsclExecSchedulerCommonBase::ConstructStatQ ()** [protected]
- 6.135.3.10 **void OsclExecSchedulerCommonBase::EndScheduling ()** [protected]
- 6.135.3.11 **void OsclExecSchedulerCommonBase::EndStats ()** [protected]
- 6.135.3.12 **void OsclExecSchedulerCommonBase::Error (int32 *anError*) const** [protected]
- 6.135.3.13 **PVActiveBase\* OsclExecSchedulerCommonBase::FindPVBase (PVActiveBase \* *active*, OsclDoubleList< PVActiveBase > &)** [protected]
- 6.135.3.14 **OSCL\_IMPORT\_REF uint32 OsclExecSchedulerCommonBase::GetId ()** [static]

Get numeric ID of current thread.

- 6.135.3.15 **OSCL\_IMPORT\_REF OsclNameString< PVSCHEDNAMELEN >\* OsclExecSchedulerCommonBase::GetName ()** [static]

Get name of scheduler for current thread.

- 6.135.3.16** `OsclExecSchedulerCommonBase* OsclExecSchedulerCommonBase::GetScheduler ()`  
 [static, protected]
- 6.135.3.17** `void OsclExecSchedulerCommonBase::IncLogPerf (uint32) [protected]`
- 6.135.3.18** `void OsclExecSchedulerCommonBase::InitExecQ (int) [protected]`
- 6.135.3.19** `void OsclExecSchedulerCommonBase::InstallScheduler () [protected]`
- 6.135.3.20** `bool OsclExecSchedulerCommonBase::IsInstalled () [inline, protected]`
- 6.135.3.21** `bool OsclExecSchedulerCommonBase::IsStarted () [protected]`
- 6.135.3.22** `void OsclExecSchedulerCommonBase::PendComplete (PVActiveBase *, int32 aReason, TPVThreadContext aContext) [protected]`
- 6.135.3.23** `void OsclExecSchedulerCommonBase::RequestCanceled (PVActiveBase *) [protected]`
- 6.135.3.24** `void OsclExecSchedulerCommonBase::ResetLogPerf () [protected]`
- 6.135.3.25** `OSCL_IMPORT_REF void OsclExecSchedulerCommonBase::ResumeScheduler ()`

Resume scheduling immediately. This API only applies to a blocking loop scheduler.

- 6.135.3.26** `OsclExecSchedulerCommonBase* OsclExecSchedulerCommonBase::SetScheduler (OsclExecSchedulerCommonBase *) [static, protected]`
- 6.135.3.27** `void OsclExecSchedulerCommonBase::ShowStats (PVActiveStats * active) [static, protected]`
- 6.135.3.28** `void OsclExecSchedulerCommonBase::ShowSummaryStats (PVActiveStats * active, PVLogger *, int64, int64 &, float &) [static, protected]`
- 6.135.3.29** `OSCL_IMPORT_REF void OsclExecSchedulerCommonBase::StartNativeScheduler ()`

Start the OS native scheduling loop. This is an alternative to the PV scheduling loop. To stop the native scheduler, use the StopScheduler API.

- 6.135.3.30** `OSCL_IMPORT_REF void OsclExecSchedulerCommonBase::StartScheduler (OsclSemaphore * sem = NULL)`

Start scheduling. This call blocks until scheduler is stopped or an error occurs.

**Parameters:**

***sem***: optional startup semaphore. If provided, the scheduler will signal this semaphore when the startup has progressed to the point that it's safe to call StopScheduler or SuspendScheduler from another thread.

**6.135.3.31 OSCL\_IMPORT\_REF void OsclExecSchedulerCommonBase::StopScheduler ()**

Stop scheduling. This API may be called from the scheduling thread or some other thread.

**6.135.3.32 OSCL\_IMPORT\_REF void OsclExecSchedulerCommonBase::SuspendScheduler ()**

Suspend scheduling when the current Run is complete. This API only applies to a blocking loop scheduler.

**6.135.3.33 void OsclExecSchedulerCommonBase::UninstallScheduler () [protected]****6.135.3.34 PVActiveBase\* OsclExecSchedulerCommonBase::UpdateTimers (uint32 & aDelay) [protected]****6.135.3.35 PVActiveBase\* OsclExecSchedulerCommonBase::UpdateTimersMsec (uint32 & aDelay) [protected]****6.135.3.36 PVActiveBase\* OsclExecSchedulerCommonBase::WaitForReadyAO () [protected]****6.135.4 Friends And Related Function Documentation****6.135.4.1 friend class OsclActiveObject [friend]****6.135.4.2 friend class OsclCoeActiveScheduler [friend]****6.135.4.3 friend class OsclError [friend]****6.135.4.4 friend class OsclExecScheduler [friend]****6.135.4.5 friend class OsclReadyQ [friend]****6.135.4.6 friend class OsclScheduler [friend]**

Reimplemented in [OsclExecScheduler](#).



6.135.4.7 friend class OsclTimerCompare [friend]

6.135.4.8 friend class OsclTimerObject [friend]

6.135.4.9 friend class PVActiveBase [friend]

6.135.4.10 friend class PVActiveStats [friend]

6.135.4.11 friend class PVSchedulerStopper [friend]

6.135.4.12 friend class PVThreadContext [friend]

## 6.135.5 Field Documentation

6.135.5.1 **Oscl\_DefAlloc\*** OsclExecSchedulerCommonBase::iAlloc [protected]

6.135.5.2 bool OsclExecSchedulerCommonBase::iBlockingMode [protected]

6.135.5.3 **PVLogger\*** OsclExecSchedulerCommonBase::iDebugLogger [protected]

6.135.5.4 **OsclMemAllocator** OsclExecSchedulerCommonBase::iDefAlloc [protected]

6.135.5.5 int32 OsclExecSchedulerCommonBase::iDelta [protected]

6.135.5.6 bool OsclExecSchedulerCommonBase::iDoStop [protected]

6.135.5.7 bool OsclExecSchedulerCommonBase::iDoSuspend [protected]

6.135.5.8 **OsclErrorTrapImp\*** OsclExecSchedulerCommonBase::iErrorTrapImp  
[protected]

6.135.5.9 **OsclTimerQ** OsclExecSchedulerCommonBase::iExecTimerQ [protected]

6.135.5.10 **int64** OsclExecSchedulerCommonBase::iGrandTotalTicks [protected]

6.135.5.11 **PVLogger\*** OsclExecSchedulerCommonBase::iLogger [protected]

6.135.5.12 char\* OsclExecSchedulerCommonBase::iLogPerfIndentStr [protected]

6.135.5.13 int32 OsclExecSchedulerCommonBase::iLogPerfIndentStrLen [protected]

6.135.5.14 uint32 OsclExecSchedulerCommonBase::iLogPerfTotal [protected]

6.135.5.15 **OsclNameString<PVSCHEDEXNAMELEN>** OsclExecSchedulerCommonBase::iName  
[protected]

6.135.5.16 bool OsclExecSchedulerCommonBase::iNativeMode [protected]

6.135.5.17 uint32 OsclExecSchedulerCommonBase::iNumAOAdded [protected]

6.135.5.18 **PVActiveStats\*** OsclExecSchedulerCommonBase::iOtherExecStats[EOtherExecStats\_-  
Last] [protected]

6.135.5.19 **OsclDoubleList<PVActiveStats>** OsclExecSchedulerCommonBase::iPVStatQ

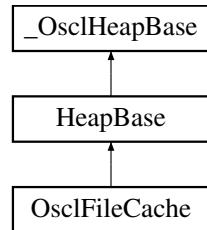
[protected]

- [oscl\\_scheduler.h](#)

## 6.136 OsclFileCache Class Reference

```
#include <oscl_file_cache.h>
```

Inheritance diagram for OsclFileCache::



### Public Methods

- [OsclFileCache \(Oscl\\_File &aContainer\)](#)
- [~OsclFileCache \(\)](#)
- int32 [Open \(uint32 mode, uint32 cache\\_size\)](#)
- void [Close \(\)](#)
- uint32 [Read \(void \\*outputBuffer, uint32 size, uint32 numelements\)](#)
- uint32 [Write \(const void \\*inputBuffer, uint32 size, uint32 numelements\)](#)
- [TOsclFileOffset FileSize \(\)](#)
- int32 [Seek \(TOsclFileOffset offset, Oscl\\_File::seek\\_type origin\)](#)
- [TOsclFileOffset Tell \(\)](#)
- int32 [Flush \(\)](#)
- int32 [EndOfFile \(\)](#)
- OSCL\_IMPORT\_REF [OsclFileCacheBuffer \\* AddFixedCache \(const Oscl\\_File::OsclFixedCacheParam &\)](#)

### Data Fields

- [OsclFileCacheBuffer \\_movableCache](#)
- [Oscl\\_Vector< OsclFileCacheBuffer, OsclMemAllocator > \\_fixedCaches](#)

### Friends

- class [OsclFileCacheBuffer](#)

### 6.136.1 Constructor & Destructor Documentation

6.136.1.1 **OsclFileCache::OsclFileCache ([Oscl\\_File](#) & *aContainer*)**

6.136.1.2 **OsclFileCache::~OsclFileCache ()**

### 6.136.2 Member Function Documentation

6.136.2.1 **OSCL\_IMPORT\_REF [OsclFileCacheBuffer](#)\* OsclFileCache::AddFixedCache (const [Oscl\\_File::OsclFixedCacheParam](#) &)**

6.136.2.2 **void OsclFileCache::Close ()**

6.136.2.3 **int32 OsclFileCache::EndOfFile () [inline]**

6.136.2.4 **[TOsclFileOffset](#) OsclFileCache::FileSize () [inline]**

6.136.2.5 **int32 OsclFileCache::Flush ()**

6.136.2.6 **int32 OsclFileCache::Open (uint32 *mode*, uint32 *cache\_size*)**

6.136.2.7 **uint32 OsclFileCache::Read (void \* *outputBuffer*, uint32 *size*, uint32 *numelements*)**

6.136.2.8 **int32 OsclFileCache::Seek ([TOsclFileOffset](#) *offset*, [Oscl\\_File::seek\\_type](#) *origin*)**

6.136.2.9 **[TOsclFileOffset](#) OsclFileCache::Tell () [inline]**

6.136.2.10 **uint32 OsclFileCache::Write (const void \* *inputBuffer*, uint32 *size*, uint32 *numelements*)**

### 6.136.3 Friends And Related Function Documentation

6.136.3.1 **friend class OsclFileCacheBuffer [friend]**

### 6.136.4 Field Documentation

6.136.4.1 **[Oscl\\_Vector](#)<[OsclFileCacheBuffer](#), [OsclMemAllocator](#)> OsclFileCache::\_fixedCaches**

6.136.4.2 **[OsclFileCacheBuffer](#) OsclFileCache::\_movableCache**

The documentation for this class was generated from the following file:

- [oscl\\_file\\_cache.h](#)

## 6.137 OsclFileCacheBuffer Class Reference

```
#include <oscl_file_cache.h>
```

### Public Methods

- [OsclFileCacheBuffer \(\)](#)
- int32 [SetPosition \(TOsclFileOffset pos\)](#)
- int32 [PrepRead \(\)](#)
- int32 [PrepWrite \(\)](#)
- int32 [WriteUpdatesToFile \(\)](#)
- int32 [FillFromFile \(uint32, uint32\)](#)
- bool [IsUpdated \(\)](#)
- bool [Contains \(TOsclFileOffset pos\)](#)
- bool [Preceeds \(TOsclFileOffset pos\)](#)

### Data Fields

- [OsclFileCache \\* iContainer](#)
- bool [isFixed](#)
- uint32 [capacity](#)
- uint32 [usableSize](#)
- uint8 \* [pBuffer](#)
- [TOsclFileOffset filePosition](#)
- uint32 [currentPos](#)
- uint32 [endPos](#)
- uint32 [updateStart](#)
- uint32 [updateEnd](#)

## 6.137.1 Constructor & Destructor Documentation

**6.137.1.1 OsclFileCacheBuffer::OsclFileCacheBuffer () [inline]**

## 6.137.2 Member Function Documentation

**6.137.2.1 bool OsclFileCacheBuffer::Contains (TOsclFileOffset *pos*) [inline]**

**6.137.2.2 int32 OsclFileCacheBuffer::FillFromFile (uint32, uint32)**

**6.137.2.3 bool OsclFileCacheBuffer::IsUpdated () [inline]**

**6.137.2.4 bool OsclFileCacheBuffer::Preceeds (TOsclFileOffset *pos*) [inline]**

**6.137.2.5 int32 OsclFileCacheBuffer::PrepRead ()**

**6.137.2.6 int32 OsclFileCacheBuffer::PrepWrite ()**

**6.137.2.7 int32 OsclFileCacheBuffer::SetPosition (TOsclFileOffset *pos*)**

**6.137.2.8 int32 OsclFileCacheBuffer::WriteUpdatesToFile ()**

## 6.137.3 Field Documentation

**6.137.3.1 uint32 OsclFileCacheBuffer::capacity**

**6.137.3.2 uint32 OsclFileCacheBuffer::currentPos**

**6.137.3.3 uint32 OsclFileCacheBuffer::endPos**

**6.137.3.4 TOsclFileOffset OsclFileCacheBuffer::filePosition**

**6.137.3.5 OsclFileCache\* OsclFileCacheBuffer::iContainer**

**6.137.3.6 bool OsclFileCacheBuffer::isFixed**

**6.137.3.7 uint8\* OsclFileCacheBuffer::pBuffer**

**6.137.3.8 uint32 OsclFileCacheBuffer::updateEnd**

**6.137.3.9 uint32 OsclFileCacheBuffer::updateStart**

**6.137.3.10 uint32 OsclFileCacheBuffer::usableSize**

The documentation for this class was generated from the following file:

- [oscl\\_file\\_cache.h](#)

## 6.138 OsclFileHandle Class Reference

```
#include <oscl_file_handle.h>
```

### Public Methods

- [OsclFileHandle \(TOsclFileHandle aHandle\)](#)
- [OsclFileHandle \(const OsclFileHandle &aHandle\)](#)
- [TOsclFileHandle Handle \(\) const](#)

### Friends

- class [Oscl\\_File](#)

#### 6.138.1 Detailed Description

OsclFileHandle is a container for a handle to a previously-opened file.

#### 6.138.2 Constructor & Destructor Documentation

**6.138.2.1 OsclFileHandle::OsclFileHandle (TOsclFileHandle *aHandle*) [inline]**

**6.138.2.2 OsclFileHandle::OsclFileHandle (const OsclFileHandle & *aHandle*) [inline]**

#### 6.138.3 Member Function Documentation

**6.138.3.1 TOsclFileHandle OsclFileHandle::Handle () const [inline]**

#### 6.138.4 Friends And Related Function Documentation

**6.138.4.1 friend class Oscl\_File [friend]**

The documentation for this class was generated from the following file:

- [oscl\\_file\\_handle.h](#)

## 6.139 OsclFileStats Class Reference

```
#include <oscl_file_stats.h>
```

### Public Methods

- [OsclFileStats \(Oscl\\_File \\*c\)](#)
- [void Start \(uint32 &aTicks\)](#)
- [void End \(TOsclFileOp aOp, uint32 aStart, uint32 aParam=0, TOsclFileOffset aParam2=0\)](#)
- [void Log \(TOsclFileOp, PVLogger \\*, uint32\)](#)
- [void LogAll \(PVLogger \\*, uint32\)](#)

### 6.139.1 Constructor & Destructor Documentation

#### 6.139.1.1 OsclFileStats::OsclFileStats ([Oscl\\_File](#) \* *c*)

### 6.139.2 Member Function Documentation

#### 6.139.2.1 void OsclFileStats::End ([TOsclFileOp](#) *aOp*, [uint32](#) *aStart*, [uint32](#) *aParam* = 0, [TOsclFileOffset](#) *aParam2* = 0)

#### 6.139.2.2 void OsclFileStats::Log ([TOsclFileOp](#), [PVLogger](#) \*, [uint32](#))

#### 6.139.2.3 void OsclFileStats::LogAll ([PVLogger](#) \*, [uint32](#))

#### 6.139.2.4 void OsclFileStats::Start ([uint32](#) & *aTicks*)

The documentation for this class was generated from the following file:

- [oscl\\_file\\_stats.h](#)

## 6.140 OsclFileStatsItem Class Reference

```
#include <oscl_file_stats.h>
```

### Data Fields

- uint32 [iOpCount](#)
- uint32 [iParam](#)
- [TOsclFileOffset](#) [iParam2](#)
- uint32 [iStartTick](#)
- uint32 [iTTotalTicks](#)

#### 6.140.1 Field Documentation

**6.140.1.1 uint32 OsclFileStatsItem::iOpCount**

**6.140.1.2 uint32 OsclFileStatsItem::iParam**

**6.140.1.3 TOsclFileOffset OsclFileStatsItem::iParam2**

**6.140.1.4 uint32 OsclFileStatsItem::iStartTick**

**6.140.1.5 uint32 OsclFileStatsItem::iTTotalTicks**

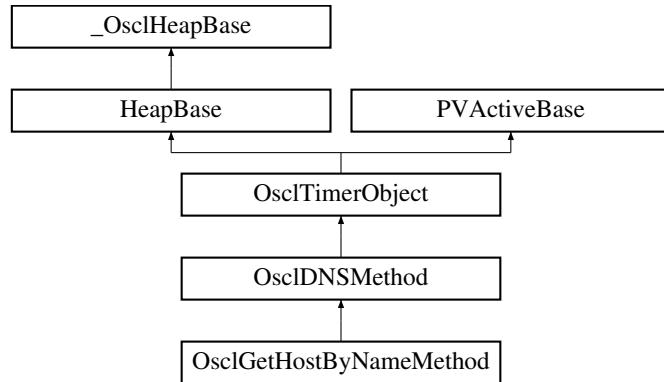
The documentation for this class was generated from the following file:

- [oscl\\_file\\_stats.h](#)

## 6.141 OsclGetHostByNameMethod Class Reference

```
#include <oscl_dns_gethostbyname.h>
```

Inheritance diagram for OsclGetHostByNameMethod::



### Public Methods

- [~OsclGetHostByNameMethod \(\)](#)
- [TPVDNSEvent GetHostByName \(char \\*name, OsclNetworkAddress \\*addr, int32 aTimeout\)](#)

### Static Public Methods

- [OsclGetHostByNameMethod \\* NewL \(Oscl\\_DefAlloc &a, OsclDNSI \\*aDNS, OsclDNSObserver \\*aObserver, uint32 aId\)](#)

#### 6.141.1 Constructor & Destructor Documentation

##### 6.141.1.1 OsclGetHostByNameMethod::~OsclGetHostByNameMethod ()

#### 6.141.2 Member Function Documentation

##### 6.141.2.1 [TPVDNSEvent OsclGetHostByNameMethod::GetHostByName \(char \\* name, OsclNetworkAddress \\* addr, int32 aTimeout\)](#)

##### 6.141.2.2 [OsclGetHostByNameMethod\\* OsclGetHostByNameMethod::NewL \(Oscl\\_DefAlloc &a, OsclDNSI \\*aDNS, OsclDNSObserver \\*aObserver, uint32 aId\) \[static\]](#)

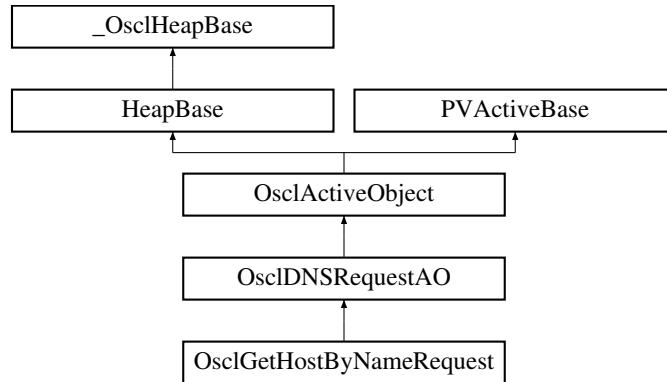
The documentation for this class was generated from the following file:

- [oscl\\_dns\\_gethostbyname.h](#)

## 6.142 OsclGetHostByNameRequest Class Reference

```
#include <oscl_dns_gethostbyname.h>
```

Inheritance diagram for OsclGetHostByNameRequest::



### Friends

- class [OsclGetHostByNameMethod](#)

#### 6.142.1 Friends And Related Function Documentation

##### 6.142.1.1 friend class OsclGetHostByNameMethod [friend]

The documentation for this class was generated from the following file:

- [oscl\\_dns\\_gethostbyname.h](#)

## 6.143 OsclInit Class Reference

```
#include <oscl_init.h>
```

### Static Public Methods

- OSCL\_IMPORT\_REF void [Init](#) (int32 &aError, const [OsclSelect](#) \*aSelect=NULL)
- OSCL\_IMPORT\_REF void [Cleanup](#) (int32 &aError, const [OsclSelect](#) \*aSelect=NULL)

#### 6.143.1 Detailed Description

Per-thread oscl initialization and cleanup.

#### 6.143.2 Member Function Documentation

##### 6.143.2.1 OSCL\_IMPORT\_REF void OsclInit::Cleanup (int32 & aError, const [OsclSelect](#) \* aSelect = NULL) [static]

This routine cleans up the Oscl modules in the calling thread.

###### Parameters:

*err*: (output) error code of any leave that occurs in initialization.

*config*: (input param) optional set of initialization parameters. If null, then full initialization with default parameters will be performed. For proper cleanup, the parameters should match the ones used during the Init call.

##### 6.143.2.2 OSCL\_IMPORT\_REF void OsclInit::Init (int32 & aError, const [OsclSelect](#) \* aSelect = NULL) [static]

This routine initializes the Oscl modules in the calling thread.

###### Parameters:

*err*: (output) error code of any leave that occurs in initialization.

*config*: (input param) optional set of initialization parameters. If null, then full initialization with default parameters will be performed.

The documentation for this class was generated from the following file:

- [oscl\\_init.h](#)

## 6.144 OsclInteger64Transport Struct Reference

```
#include <oscl_int64_utils.h>
```

### Data Fields

- uint32 [iHigh](#)
- uint32 [iLow](#)

#### 6.144.1 Detailed Description

OsclInteger64Transport Structure

Structure to only transport 64-bit integer values uint64 and int64 could be classes so needed for cases where having a class will not work.

#### 6.144.2 Field Documentation

##### 6.144.2.1 uint32 OsclInteger64Transport::iHigh

##### 6.144.2.2 uint32 OsclInteger64Transport::iLow

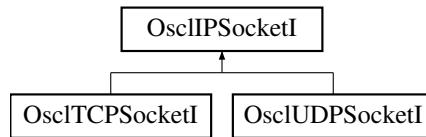
The documentation for this struct was generated from the following file:

- [oscl\\_int64\\_utils.h](#)

## 6.145 OsclIPSocketI Class Reference

```
#include <oscl_ip_socket.h>
```

Inheritance diagram for OsclIPSocketI::



### Public Methods

- int32 [Bind \(OsclNetworkAddress &aAddress\)](#)
- int32 [Join \(OsclNetworkAddress &aAddress\)](#)
- int32 [SetRecvBufferSize \(uint32 size\)](#)
- virtual int32 [Close \(\)=0](#)
- virtual uint8 \* [GetRecvData \(int32 \\*aLength\)=0](#)
- virtual uint8 \* [GetSendData \(int32 \\*aLength\)=0](#)
- virtual [~OsclIPSocketI \(\)](#)
- [OsclSocketServI \\* SocketServ \(\)](#)
- [Oscl\\_DefAlloc & Alloc \(\)](#)

### Protected Methods

- [OsclIPSocketI \(Oscl\\_DefAlloc &a\)](#)
- void [ConstructL \(OsclSocketObserver \\*aObs, OsclSocketI \\*aSock, OsclSocketServI \\*aServ, uint32 aId\)](#)

### Protected Attributes

- [Oscl\\_DefAlloc & iAlloc](#)
- [OsclNetworkAddress iAddress](#)
- uint32 [iId](#)
- [OsclSocketObserver \\* iObserver](#)
- [OsclSocketI \\* iSocket](#)
- [OsclSocketServI \\* iSocketServ](#)
- [PVLogger \\* iLogger](#)

### Friends

- class [OsclSocketRequestAO](#)
- class [OsclSocketMethod](#)

### 6.145.1 Constructor & Destructor Documentation

**6.145.1.1** virtual OsclIPSocketI::~OsclIPSocketI () [inline, virtual]

**6.145.1.2** OsclIPSocketI::OsclIPSocketI ([Oscl\\_DefAlloc](#) & *a*) [inline, protected]

### 6.145.2 Member Function Documentation

**6.145.2.1** [Oscl\\_DefAlloc](#)& OsclIPSocketI::Alloc () [inline]

**6.145.2.2** int32 OsclIPSocketI::Bind ([OsclNetworkAddress](#) & *aAddress*)

**6.145.2.3** virtual int32 OsclIPSocketI::Close () [pure virtual]

Implemented in [OsclTCPSocketI](#), and [OsclUDPSocketI](#).

**6.145.2.4** void OsclIPSocketI::ConstructL ([OsclSocketObserver](#) \* *aObs*, [OsclSocketI](#) \* *aSock*,  
[OsclSocketServI](#) \* *aServ*, uint32 *aId*) [protected]

**6.145.2.5** virtual uint8\* OsclIPSocketI::GetRecvData (int32 \* *aLength*) [pure virtual]

Implemented in [OsclTCPSocketI](#), and [OsclUDPSocketI](#).

**6.145.2.6** virtual uint8\* OsclIPSocketI::GetSendData (int32 \* *aLength*) [pure virtual]

Implemented in [OsclTCPSocketI](#), and [OsclUDPSocketI](#).

6.145.2.7 int32 OsclIPSocketI::Join ([OsclNetworkAddress](#) & *aAddress*)

6.145.2.8 int32 OsclIPSocketI::SetRecvBufferSize (uint32 *size*)

6.145.2.9 [OsclSocketServI](#)\* OsclIPSocketI::SocketServ () [inline]

### 6.145.3 Friends And Related Function Documentation

6.145.3.1 friend class OsclSocketMethod [friend]

6.145.3.2 friend class OsclSocketRequestAO [friend]

### 6.145.4 Field Documentation

6.145.4.1 [OsclNetworkAddress](#) OsclIPSocketI::iAddress [protected]

6.145.4.2 [Oscl\\_DefAlloc](#)& OsclIPSocketI::iAlloc [protected]

6.145.4.3 uint32 OsclIPSocketI::iId [protected]

6.145.4.4 [PVLogger](#)\* OsclIPSocketI::iLogger [protected]

6.145.4.5 [OsclSocketObserver](#)\* OsclIPSocketI::iObserver [protected]

6.145.4.6 [OsclSocketI](#)\* OsclIPSocketI::iSocket [protected]

6.145.4.7 [OsclSocketServI](#)\* OsclIPSocketI::iSocketServ [protected]

The documentation for this class was generated from the following file:

- [oscl\\_ip\\_socket.h](#)

## 6.146 OsclJump Class Reference

```
#include <oscl_error_imp_jumps.h>
```

### Public Methods

- void [Jump](#) (int a)
- jmp\_buf \* [Top](#) ()
- [~OsclJump](#) ()

### Static Public Methods

- OSCL\_IMPORT\_REF void [StaticJump](#) (int a)

### Friends

- class [OsclErrorTrapImp](#)

#### 6.146.1 Constructor & Destructor Documentation

##### 6.146.1.1 [OsclJump::~OsclJump](#) () [inline]

#### 6.146.2 Member Function Documentation

##### 6.146.2.1 [void OsclJump::Jump](#) (int *a*) [inline]

##### 6.146.2.2 [OSCL\\_IMPORT\\_REF void OsclJump::StaticJump](#) (int *a*) [static]

##### 6.146.2.3 [jmp\\_buf\\* OsclJump::Top](#) () [inline]

#### 6.146.3 Friends And Related Function Documentation

##### 6.146.3.1 [friend class OsclErrorTrapImp](#) [friend]

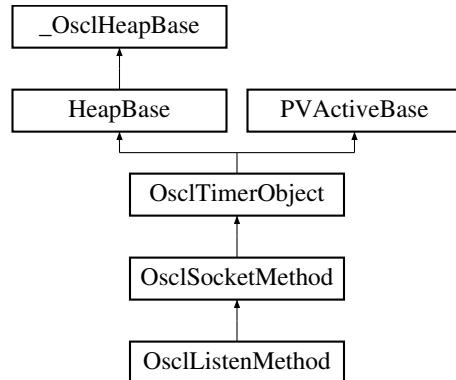
The documentation for this class was generated from the following file:

- [oscl\\_error\\_imp\\_jumps.h](#)

## 6.147 OsclListenMethod Class Reference

```
#include <oscl_socket_listen.h>
```

Inheritance diagram for OsclListenMethod::



### Public Methods

- [~OsclListenMethod \(\)](#)
- [TPVSocketEvent Listen \(uint32 qsize, int32 aTimeout\)](#)
- [OsclListenRequest \\* ListenRequest \(\)](#)

### Static Public Methods

- [OsclListenMethod \\* NewL \(OsclIPSocketI &c\)](#)

#### 6.147.1 Constructor & Destructor Documentation

##### 6.147.1.1 OsclListenMethod::~OsclListenMethod ()

#### 6.147.2 Member Function Documentation

##### 6.147.2.1 TPVSocketEvent OsclListenMethod::Listen (uint32 qsize, int32 aTimeout)

##### 6.147.2.2 OsclListenRequest\* OsclListenMethod::ListenRequest () [inline]

##### 6.147.2.3 OsclListenMethod\* OsclListenMethod::NewL (OsclIPSocketI &c) [static]

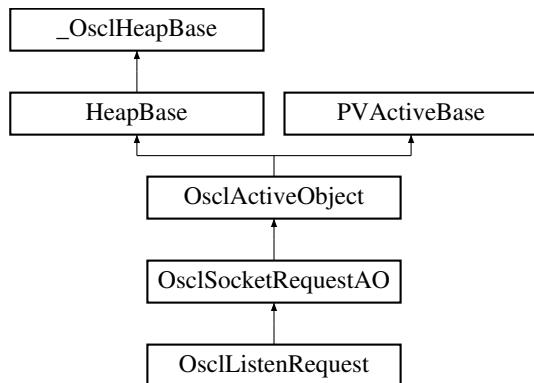
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_listen.h](#)

## 6.148 OsclListenRequest Class Reference

```
#include <oscl_socket_listen.h>
```

Inheritance diagram for OsclListenRequest::



### Public Methods

- [OsclListenRequest \(OsclSocketMethod &c\)](#)
- void [Listen \(uint32 qsize\)](#)

#### 6.148.1 Detailed Description

This is the AO that interacts with the socket server

#### 6.148.2 Constructor & Destructor Documentation

**6.148.2.1 OsclListenRequest::OsclListenRequest ([OsclSocketMethod & c](#)) [inline]**

#### 6.148.3 Member Function Documentation

**6.148.3.1 void OsclListenRequest::Listen (uint32 *qsize*)**

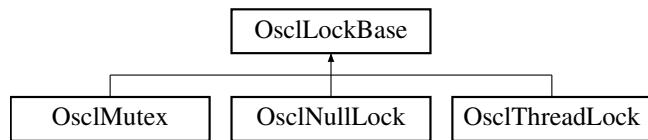
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_listen.h](#)

## 6.149 OsclLockBase Class Reference

```
#include <oscl_lock_base.h>
```

Inheritance diagram for OsclLockBase::



### Public Methods

- virtual void [Lock \(\)=0](#)
- virtual void [Unlock \(\)=0](#)
- virtual [~OsclLockBase \(\)](#)

#### 6.149.1 Constructor & Destructor Documentation

**6.149.1.1 virtual OsclLockBase::~OsclLockBase () [inline, virtual]**

#### 6.149.2 Member Function Documentation

**6.149.2.1 virtual void OsclLockBase::Lock () [pure virtual]**

Implemented in [OsclNullLock](#), [OsclMutex](#), and [OsclThreadLock](#).

**6.149.2.2 virtual void OsclLockBase::Unlock () [pure virtual]**

Implemented in [OsclNullLock](#), [OsclMutex](#), and [OsclThreadLock](#).

The documentation for this class was generated from the following file:

- [oscl\\_lock\\_base.h](#)

## 6.150 OsclMem Class Reference

```
#include <oscl_mem.h>
```

### Static Public Methods

- OSCL\_IMPORT\_REF void [Init \(\)](#)
- OSCL\_IMPORT\_REF void [Cleanup \(\)](#)

#### 6.150.1 Member Function Documentation

##### 6.150.1.1 OSCL\_IMPORT\_REF void OsclMem::Cleanup () [static]

Per-thread cleanup of Oscl Memory @exception: Leaves on error;

##### 6.150.1.2 OSCL\_IMPORT\_REF void OsclMem::Init () [static]

Per-thread initialization of Oscl Memory

#### Parameters:

*lock*: A lock class for use with multi-threaded applications. The lock is needed in use cases where memory may be allocated in one thread and freed in another. In this case, there must be a single lock object, and its pointer must be passed to the [OsclMem::Init](#) call in each thread. If no lock is provided, the memory manager will not be thread-safe. @exception: Leaves on error

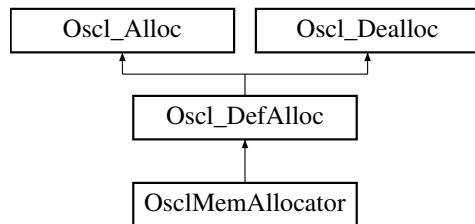
The documentation for this class was generated from the following file:

- [oscl\\_mem.h](#)

## 6.151 OsclMemAllocator Class Reference

```
#include <oscl_mem.h>
```

Inheritance diagram for OsclMemAllocator::



### Public Methods

- [OsclAny \\* allocate \(const uint32 n\)](#)
- [OsclAny \\* allocate\\_fl \(const uint32 n, const char \\*file\\_name, const int line\\_num\)](#)
- void [deallocate \(OsclAny \\*p\)](#)

#### 6.151.1 Detailed Description

A simple allocator class. Configurable as to whether this goes through the memory manager or not.

#### 6.151.2 Member Function Documentation

##### 6.151.2.1 [OsclAny\\* OsclMemAllocator::allocate \(const uint32 n\)](#) [inline, virtual]

This API throws an exception when malloc returns NULL. n must be greater than 0.

###### Returns:

pointer (or Leave with OsclErrNoMemory )

Implements [Oscl\\_DefAlloc](#).

##### 6.151.2.2 [OsclAny\\* OsclMemAllocator::allocate\\_fl \(const uint32 n, const char \\*file\\_name, const int line\\_num\)](#) [inline, virtual]

Reimplemented from [Oscl\\_DefAlloc](#).

##### 6.151.2.3 void [OsclMemAllocator::deallocate \(OsclAny \\*p\)](#) [inline, virtual]

Implements [Oscl\\_DefAlloc](#).

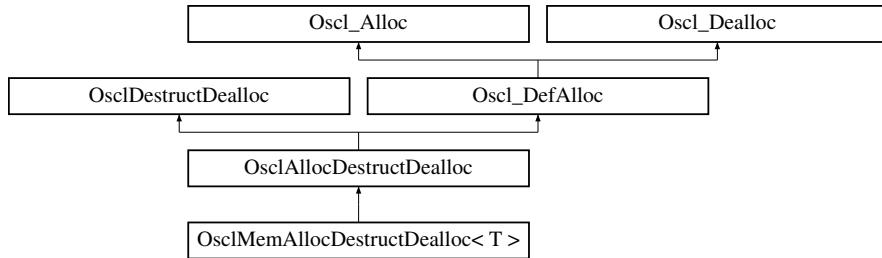
The documentation for this class was generated from the following file:

- [oscl\\_mem.h](#)

## 6.152 OsclMemAllocDestructDealloc< T > Class Template Reference

```
#include <oscl_mem.h>
```

Inheritance diagram for OsclMemAllocDestructDealloc< T >::



### Public Methods

- [OsclAny \\* allocate\\_f1](#) (const uint32 size, const char \*file\_name, const int line\_num)
- [OsclAny \\* allocate](#) (const uint32 size)
- void [deallocate](#) (OsclAny \*p)
- void [destruct\\_and\\_dealloc](#) (OsclAny \*p)

#### 6.152.1 Detailed Description

`template<class T> class OsclMemAllocDestructDealloc< T >`

An [OsclAllocDestructDealloc](#) class that uses [OsclMemAllocator](#).

#### 6.152.2 Member Function Documentation

**6.152.2.1 template<class T> [OsclAny\\*](#) OsclMemAllocDestructDealloc< T >::allocate (const uint32 size) [inline, virtual]**

Implements [Oscl\\_DefAlloc](#).

**6.152.2.2 template<class T> [OsclAny\\*](#) OsclMemAllocDestructDealloc< T >::allocate\_f1 (const uint32 size, const char \*file\_name, const int line\_num) [inline, virtual]**

Reimplemented from [Oscl\\_DefAlloc](#).

**6.152.2.3 template<class T> void OsclMemAllocDestructDealloc< T >::deallocate ([OsclAny](#) \* p) [inline, virtual]**

Implements [Oscl\\_DefAlloc](#).

**6.152.2.4 template<class T> void OsclMemAllocDestructDealloc< T >::destruct\_and\_dealloc  
(OsclAny \* p) [inline, virtual]**

Implements [OsclDestructDealloc](#).

The documentation for this class was generated from the following file:

- [oscl\\_mem.h](#)

## 6.153 OsclMemAudit Class Reference

```
#include <oscl_mem_audit.h>
```

### Public Methods

- [OsclMemAudit \(\)](#)
- [~OsclMemAudit \(\)](#)
- [void \\* MM\\_allocate \(const OsclMemStatsNode \\*statsNode, uint32 sizeIn, const char \\*pFileName, uint32 lineNumber, bool allocNodeTracking=false\)](#)
- [bool MM\\_deallocate \(void \\*pMemBlockIn\)](#)
- [MM\\_Stats\\_t \\* MM\\_GetStats \(const char \\*const tagIn\)](#)
- [uint32 MM\\_GetStatsInDepth \(const char \\*tagIn, MM\\_Stats\\_CB \\*array\\_ptr, uint32 max\\_nodes\)](#)
- [uint32 MM\\_GetTreeNodes \(const char \\*tagIn\)](#)
- [bool MM\\_AddTag \(const char \\*tagIn\)](#)
- [const OsclMemStatsNode \\* MM\\_GetTagName \(const char \\*tagIn\)](#)
- [const OsclMemStatsNode \\* MM\\_GetExistingTag \(const char \\*tagIn\)](#)
- [const OsclMemStatsNode \\* MM\\_GetRootNode \(\)](#)
- [uint32 MM\\_GetAllocNodeInfo \(MM\\_AllocQueryInfo \\*output\\_array, uint32 max\\_array\\_size, uint32 offset\)](#)
- [MM\\_AllocQueryInfo \\* MM\\_CreateAllocNodeInfo \(uint32 max\\_array\\_size\)](#)
- [void MM\\_ReleaseAllocNodeInfo \(MM\\_AllocQueryInfo \\*info\)](#)
- [bool MM\\_Validate \(const void \\*ptrIn\)](#)
- [uint32 MM\\_GetAllocNo \(void\)](#)
- [void MM\\_GetOverheadStats \(MM\\_AuditOverheadStats &stats\)](#)
- [uint32 MM\\_GetNumAllocNodes \(\)](#)
- [uint32 MM\\_GetMode \(void\)](#)
- [uint8 MM\\_GetPrefillPattern \(void\)](#)
- [uint32 MM\\_GetPostfillPattern \(void\)](#)
- [void MM\\_SetMode \(uint32 inMode\)](#)
- [void MM\\_SetPrefillPattern \(uint8 pattern\)](#)
- [void MM\\_SetPostfillPattern \(uint8 pattern\)](#)
- [void MM\\_SetTagLevel \(uint32 level\)](#)
- [bool MM\\_SetFailurePoint \(const char \\*tagIn, uint32 alloc\\_number\)](#)
- [void MM\\_UnsetFailurePoint \(const char \\*tagIn\)](#)
- [int32 MM\\_GetRefCount \(\)](#)
- [OsclLockBase \\* GetLock \(\)](#)

### Friends

- class [OsclMemGlobalAuditObject](#)

#### 6.153.1 Constructor & Destructor Documentation

##### 6.153.1.1 OsclMemAudit::OsclMemAudit () [inline]

Constructor, create the root node in statistics table

**6.153.1.2 OsclMemAudit::~OsclMemAudit () [inline]**

A destructor, remove all the nodes in allocation andstatistics table

**6.153.2 Member Function Documentation****6.153.2.1 OsclLockBase\* OsclMemAudit::GetLock () [inline]**

API to obtain mem lock ptr

**6.153.2.2 bool OsclMemAudit::MM\_AddTag (const char \* tagIn) [inline]**

API to add a node and zero out its counters; Note that this tag should be re-used

**Parameters:**

*tagIn* input tag

**Returns:**

true if operation succeeds;

**6.153.2.3 void\* OsclMemAudit::MM\_allocate (const OsclMemStatsNode \* statsNode, uint32 sizeIn, const char \* pFileName, uint32 lineNumber, bool allocNodeTracking = false) [inline]**

The following are APIs to \_\_nothrow\_ / const \_\_nothrow\_

**Returns:**

the memory pointer if operation succeeds.

**6.153.2.4 MM\_AllocQueryInfo\* OsclMemAudit::MM\_CreateAllocNodeInfo (uint32 max\_array\_size) [inline]****6.153.2.5 bool OsclMemAudit::MM\_deallocate (void \* pMemBlockIn) [inline]****Returns:**

true if operation succeeds;

**6.153.2.6 uint32 OsclMemAudit::MM\_GetAllocNo (void) [inline]**

API to get the current allocation number

**Returns:**

the current allocation number

**6.153.2.7** `uint32 OsclMemAudit::MM_GetAllocNodeInfo (MM_AllocQueryInfo * output_array, uint32 max_array_size, uint32 offset) [inline]`

API to query the list of alloc nodes. It copies the information into the provided output array.

**Parameters:**

*output\_array* the array where the data will be written

*max\_array\_size* the max number of output array elements

*offset* the offset into the alloc node list from which the data should begin.

**Returns:**

the number of valid nodes in the output array

**6.153.2.8** `const OsclMemStatsNode* OsclMemAudit::MM_GetExistingTag (const char * tagIn) [inline]`

API to add a node and zero out its counters; Note that this tag should be re-used

**Parameters:**

*tagIn* input tag

**Returns:**

true if operation succeeds;

**6.153.2.9** `uint32 OsclMemAudit::MM_GetMode (void) [inline]`

API to get the operating mode of the mm\_audit class.

**6.153.2.10** `uint32 OsclMemAudit::MM_GetNumAllocNodes () [inline]`

API to get the number of allocation nodes (records) for allocations that are being tracked individually.

**6.153.2.11** `void OsclMemAudit::MM_GetOverheadStats (MM_AuditOverheadStats & stats) [inline]`

API to get the overhead statistics for the memory used by the mm\_audit class.

**6.153.2.12** `uint32 OsclMemAudit::MM_GetPostfillPattern (void) [inline]`

API to get the postfill pattern. The pattern is used to fill the memory before freeing it.

**6.153.2.13** `uint8 OsclMemAudit::MM_GetPrefillPattern (void) [inline]`

API to get the prefill pattern. The pattern is used to fill the memory before returning it to the caller.

**6.153.2.14 int32 OsclMemAudit::MM\_GetRefCount () [inline]**

**6.153.2.15 const OsclMemStatsNode\* OsclMemAudit::MM\_GetRootNode () [inline]**

**6.153.2.16 MM\_Stats\_t\* OsclMemAudit::MM\_GetStats (const char \*const tagIn) [inline]**

API to get memory statistics through context string(tag)

**Returns:**

statistics pointer if operation succeeds

**6.153.2.17 uint32 OsclMemAudit::MM\_GetStatsInDepth (const char \* tagIn, MM\_Stats\_CB \* array\_ptr, uint32 max\_nodes) [inline]**

API to get memory statistics in detail through context string(tag) including its subtree

**Returns:**

statistics pointer array and actual number of nodes if operation succeeds

**6.153.2.18 const OsclMemStatsNode\* OsclMemAudit::MM\_GetTagName (const char \* tagIn) [inline]**

API to add a node and zero out its counters; Note that this tag should be re-used

**Parameters:**

*tagIn* input tag

**Returns:**

pointer to [OsclMemStatsNode](#) which should be passed to MM\_allocate

**6.153.2.19 uint32 OsclMemAudit::MM\_GetTreeNodes (const char \* tagIn) [inline]**

API to get the number of tree nodes including the tag node and its subtree

**Parameters:**

*tagIn* input tag

**Returns:**

the number of tree nodes ; 0 means no tag node

**6.153.2.20 void OsclMemAudit::MM\_ReleaseAllocNodeInfo (MM\_AllocQueryInfo \* info) [inline]**

**6.153.2.21 bool OsclMemAudit::MM\_SetFailurePoint (const char \* tagIn, uint32 alloc\_number) [inline]**

API to insert allocation failure deterministically according to allocation number associated with tag

**Parameters:**

*tagIn* input tag  
*alloc\_number* allocation number associated with tag

**Returns:**

true if operation succeeds;

**6.153.2.22 void OsclMemAudit::MM\_SetMode (uint32 *inMode*) [inline]**

API to set the operating mode of the mm\_audit class.

**6.153.2.23 void OsclMemAudit::MM\_SetPostfillPattern (uint8 *pattern*) [inline]**

API to set the postfill pattern.

**6.153.2.24 void OsclMemAudit::MM\_SetPrefillPattern (uint8 *pattern*) [inline]**

API to set the prefill pattern.

**6.153.2.25 void OsclMemAudit::MM\_SetTagLevel (uint32 *level*) [inline]**

API to set the maximum tag level,i.e. tag level for a.b.c.d = 4

**Parameters:**

*level* input tag level to be set

**6.153.2.26 void OsclMemAudit::MM\_UnsetFailurePoint (const char \* *tagIn*) [inline]**

API to cancel the allocation failure point associated with tag

**Parameters:**

*tagIn* input tag

**6.153.2.27 bool OsclMemAudit::MM\_Validate (const void \* *ptrIn*) [inline]**

API to check the input pointer is a valid pointer to a chunk of memory

**Parameters:**

*ptrIn* input pointer to be validated

**Returns:**

true if operation succeeds;

### 6.153.3 Friends And Related Function Documentation

#### 6.153.3.1 friend class OsclMemGlobalAuditObject [friend]

The documentation for this class was generated from the following file:

- [oscl\\_mem\\_audit.h](#)

## 6.154 OSCLMemAutoPtr< T, \_Allocator > Class Template Reference

The oscl\_auto\_ptr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or indirectly) by new. When the oscl\_auto\_ptr expires, its destructor uses delete to free the memory.

```
#include <oscl_mem_auto_ptr.h>
```

### Public Methods

- **OSCLMemAutoPtr** (T \*inPtr=0)  
*Default constructor Initializes the pointer and takes ownership.*
- **OSCLMemAutoPtr** (const OSCLMemAutoPtr< T > &\_Y)  
*Copy constructor.*
- **OSCLMemAutoPtr< T, \_Allocator > & operator=** (const OSCLMemAutoPtr< T, \_Allocator > &\_Y)  
*Assignment operator from an another oscl\_auto\_ptr.*
- **~OSCLMemAutoPtr** ()  
*Destructor.*
- **T & operator \*** () const  
*The indirection operator (\*) accesses a value indirectly, through a pointer.*
- **T \* operator ->** () const  
*The indirection operator (->) accesses a value indirectly, through a pointer.*
- **void takeOwnership** (T \*ptr)  
*The takeOwnership function assigns the value with ownership.*
- **void allocate** (**oscl\_memsize\_t** size)
- **void setWithoutOwnership** (T \*ptr)  
*The takeOwnership function assigns the value with ownership.*
- **T \* get** () const  
*get() method returns the pointer, currently owned by the class.*
- **T \* release** () const  
*release() method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.*

### Static Public Methods

- **void deallocate** (T \*ptr)

## Data Fields

- bool [\\_Ownership](#)

### 6.154.1 Detailed Description

```
template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>> class OSCLMemAuto-  
Ptr< T, _Allocator >
```

The oscl\_auto\_ptr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or indirectly) by new. When the oscl\_auto\_ptr expires, its destructor uses delete to free the memory.

The purpose of this class is to provide a way to prevent accidental memory leaks in a class or a method, due to "not remembering to delete" variables allocated on the heap. Thus if you assign an address returned by new to an oscl\_auto\_ptr object, you don't have to remember to free the memory later, it will be freed automatically when the object goes out of scope. The oscl\_auto\_ptr is an example of a smart pointer, an object that acts like a pointer, but with additional features. The class is defined so that it acts like a regular pointer in most respects

### 6.154.2 Constructor & Destructor Documentation

```
6.154.2.1 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>>  
OSCLMemAutoPtr< T, _Allocator >::OSCLMemAutoPtr (T * inPtr = 0) [inline,  
explicit]
```

Default constructor Initializes the pointer and takes ownership.

```
6.154.2.2 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>>  
OSCLMemAutoPtr< T, _Allocator >::OSCLMemAutoPtr (const OSCLMemAutoPtr<  
T > & _Y) [inline]
```

Copy constructor.

Initializes the pointer and takes ownership from another oscl\_auto\_ptr. Note that the other class does NOT own the pointer any longer, and hence it is NOT its responsibility to free it.

```
6.154.2.3 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>>  
OSCLMemAutoPtr< T, _Allocator >::~OSCLMemAutoPtr () [inline]
```

Destructor.

The pointer is deleted in case this class still has ownership

### 6.154.3 Member Function Documentation

**6.154.3.1 template<class T, class \_Allocator = Oscl\_TAlloc<T, OsclMemAllocator>> void OSCLMemAutoPtr< T, \_Allocator >::allocate (oscl\_memsize\_t size) [inline]**

**6.154.3.2 template<class T, class \_Allocator = Oscl\_TAlloc<T, OsclMemAllocator>> void OSCLMemAutoPtr< T, \_Allocator >::deallocate (T \*ptr) [inline, static]**

**6.154.3.3 template<class T, class \_Allocator = Oscl\_TAlloc<T, OsclMemAllocator>> T\* OSCLMemAutoPtr< T, \_Allocator >::get () const [inline]**

[get\(\)](#) method returns the pointer, currently owned by the class.

**6.154.3.4 template<class T, class \_Allocator = Oscl\_TAlloc<T, OsclMemAllocator>> T& OSCLMemAutoPtr< T, \_Allocator >::operator \* () const [inline]**

The indirection operator (\*) accesses a value indirectly, through a pointer.

This operator ensures that the OSCLMemAutoPtr can be used like the regular pointer that it was initialized with.

**6.154.3.5 template<class T, class \_Allocator = Oscl\_TAlloc<T, OsclMemAllocator>> T\* OSCLMemAutoPtr< T, \_Allocator >::operator -> () const [inline]**

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the OSCLMemAutoPtr can be used like the regular pointer that it was initialized with.

**6.154.3.6 template<class T, class \_Allocator = Oscl\_TAlloc<T, OsclMemAllocator>> OSCLMemAutoPtr< T, \_Allocator >& OSCLMemAutoPtr< T, \_Allocator >::operator=(const OSCLMemAutoPtr< T, \_Allocator > & \_Y) [inline]**

Assignment operator from an another oscl\_auto\_ptr.

**Parameters:**

*\_Y* The value parameter should be another oscl\_auto\_ptr

**Returns:**

Returns a reference to this oscl\_auto\_ptr instance with pointer initialized.

**Precondition:**

The input class should be non-null and should point to a valid pointer.

This assignment operator initializes the class to the contents of the oscl\_auto\_ptr given as the input parameter. The ownership of the pointer is transferred.

**6.154.3.7 template<class T, class \_Allocator = Oscl\_TAlloc<T, OsclMemAllocator>> T\* OSCLMemAutoPtr< T, \_Allocator >::release () const [inline]**

[release\(\)](#) method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.

**6.154.3.8 template<class T, class \_Allocator = Oscl\_TAlloc<T, OsclMemAllocator>> void OSCLMemAutoPtr< T, \_Allocator >::setWithoutOwnership (T \*ptr) [inline]**

The takeOwnership function assigns the value with ownership.

**6.154.3.9 template<class T, class \_Allocator = Oscl\_TAlloc<T, OsclMemAllocator>> void OSCLMemAutoPtr< T, \_Allocator >::takeOwnership (T \*ptr) [inline]**

The takeOwnership function assigns the value with ownership.

#### **6.154.4 Field Documentation**

**6.154.4.1 template<class T, class \_Allocator = Oscl\_TAlloc<T, OsclMemAllocator>> bool OSCLMemAutoPtr< T, \_Allocator >::\_Ownership**

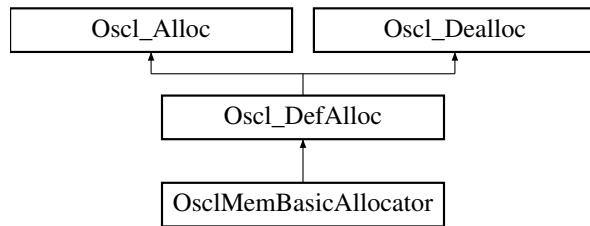
The documentation for this class was generated from the following file:

- [oscl\\_mem\\_auto\\_ptr.h](#)

## 6.155 OsclMemBasicAllocator Class Reference

```
#include <oscl_mem.h>
```

Inheritance diagram for OsclMemBasicAllocator::



### Public Methods

- [OsclAny \\* allocate \(const uint32 n\)](#)
- [void deallocate \(OsclAny \\*p\)](#)

#### 6.155.1 Detailed Description

A simple allocator class that does not use the memory management.

Note: this allocator is for internal use by Oscl only. Higher level code should use [OsclMemAllocator](#).

#### 6.155.2 Member Function Documentation

##### 6.155.2.1 [OsclAny\\* OsclMemBasicAllocator::allocate \(const uint32 n\) \[inline, virtual\]](#)

This API throws an exception when malloc returns NULL. n must be greater than 0.

###### Returns:

pointer (or Leave with OsclErrNoMemory )

Implements [Oscl\\_DefAlloc](#).

##### 6.155.2.2 [void OsclMemBasicAllocator::deallocate \(OsclAny \\*p\) \[inline, virtual\]](#)

Implements [Oscl\\_DefAlloc](#).

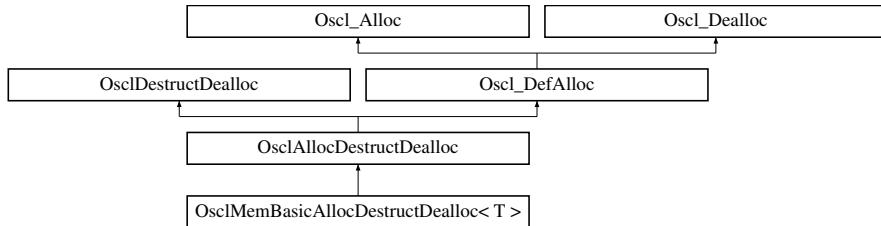
The documentation for this class was generated from the following file:

- [oscl\\_mem.h](#)

## 6.156 OsclMemBasicAllocDestructDealloc< T > Class Template Reference

```
#include <oscl_mem.h>
```

Inheritance diagram for OsclMemBasicAllocDestructDealloc< T >::



### Public Methods

- [OsclAny \\* allocate \(const uint32 size\)](#)
- [void deallocate \(OsclAny \\*p\)](#)
- [void destruct\\_and\\_dealloc \(OsclAny \\*p\)](#)

#### 6.156.1 Detailed Description

**template<class T> class OsclMemBasicAllocDestructDealloc< T >**

An [OsclAllocDestructDealloc](#) class that uses [OsclMemBasicAllocator](#).

#### 6.156.2 Member Function Documentation

**6.156.2.1 template<class T> [OsclAny\\*](#) OsclMemBasicAllocDestructDealloc< T >::allocate (const uint32 size) [inline, virtual]**

Implements [Oscl\\_DefAlloc](#).

**6.156.2.2 template<class T> void OsclMemBasicAllocDestructDealloc< T >::deallocate ([OsclAny](#) \* p) [inline, virtual]**

Implements [Oscl\\_DefAlloc](#).

**6.156.2.3 template<class T> void OsclMemBasicAllocDestructDealloc< T >::destruct\_and\_dealloc ([OsclAny](#) \* p) [inline, virtual]**

Implements [OsclDestructDealloc](#).

The documentation for this class was generated from the following file:

- [oscl\\_mem.h](#)

## 6.157 OsclMemGlobalAuditObject Class Reference

```
#include <oscl_mem.h>
```

### Public Types

- `typedef OsclMemAudit audit_type`

### Static Public Methods

- `OSCL_IMPORT_REF audit_type * getGlobalMemAuditObject ()`

### Friends

- class `OsclMem`

#### 6.157.1 Member Typedef Documentation

##### 6.157.1.1 `typedef OsclMemAudit OsclMemGlobalAuditObject::audit_type`

#### 6.157.2 Member Function Documentation

##### 6.157.2.1 `OSCL_IMPORT_REF audit_type* OsclMemGlobalAuditObject::getGlobalMemAuditObject () [static]`

returns the global audit object. For use in macros only– not a public API.

#### 6.157.3 Friends And Related Function Documentation

##### 6.157.3.1 `friend class OsclMem [friend]`

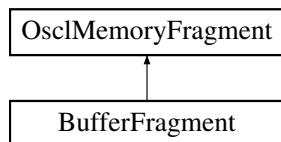
The documentation for this class was generated from the following file:

- `oscl_mem.h`

## 6.158 OsclMemoryFragment Struct Reference

```
#include <oscl_types.h>
```

Inheritance diagram for OsclMemoryFragment::



### Data Fields

- uint32 [len](#)
- void \* [ptr](#)

#### 6.158.1 Field Documentation

##### 6.158.1.1 uint32 OsclMemoryFragment::len

##### 6.158.1.2 void\* OsclMemoryFragment::ptr

The documentation for this struct was generated from the following file:

- [oscl\\_types.h](#)

## 6.159 OsclMemPoolAllocator Class Reference

```
#include <oscl_mempool_allocator.h>
```

### Public Methods

- [OsclMemPoolAllocator \(Oscl\\_DefAlloc \\*gen\\_alloc=NULL\)](#)
- [virtual ~OsclMemPoolAllocator \(\)](#)
- [OsclAny \\* CreateMemPool \(const uint32 aNumChunk=2, const uint32 aChunkSize=4\)](#)
- [void DestroyMemPool \(\)](#)
- [uint oscl\\_mem\\_aligned\\_size \(uint size\)](#)

#### 6.159.1 Constructor & Destructor Documentation

**6.159.1.1 OsclMemPoolAllocator::OsclMemPoolAllocator ([Oscl\\_DefAlloc \\* gen\\_alloc = NULL](#))**

**6.159.1.2 virtual OsclMemPoolAllocator::~OsclMemPoolAllocator () [virtual]**

#### 6.159.2 Member Function Documentation

**6.159.2.1 [OsclAny\\* OsclMemPoolAllocator::CreateMemPool \(const uint32 aNumChunk = 2, const uint32 aChunkSize = 4\)](#)**

**6.159.2.2 void OsclMemPoolAllocator::DestroyMemPool ()**

**6.159.2.3 [uint OsclMemPoolAllocator::oscl\\_mem\\_aligned\\_size \(uint size\)](#)**

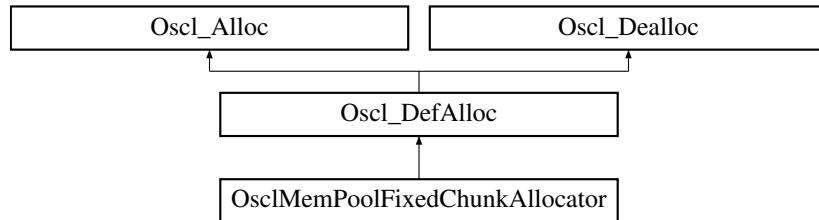
The documentation for this class was generated from the following file:

- [oscl\\_mempool\\_allocator.h](#)

## 6.160 OsclMemPoolFixedChunkAllocator Class Reference

```
#include <oscl_mem_mempool.h>
```

Inheritance diagram for OsclMemPoolFixedChunkAllocator::



### Public Methods

- OSCL\_IMPORT\_REF OsclMemPoolFixedChunkAllocator (const uint32 numchunk=1, const uint32 chunksze=0, Oscl\_DefAlloc \*gen\_alloc=NULL)
- virtual OSCL\_IMPORT\_REF void enablenullpointerreturn ()
- virtual OSCL\_IMPORT\_REF ~OsclMemPoolFixedChunkAllocator ()
- virtual OSCL\_IMPORT\_REF OsclAny \* allocate (const uint32 n)
- virtual OSCL\_IMPORT\_REF void deallocate (OsclAny \*p)
- virtual OSCL\_IMPORT\_REF void notifyfreechunkavailable (OsclMemPoolFixedChunkAllocatorObserver &obs, OsclAny \*aContextData=NULL)
- virtual OSCL\_IMPORT\_REF void CancelFreeChunkAvailableCallback ()
- OSCL\_IMPORT\_REF void addRef ()
- OSCL\_IMPORT\_REF void removeRef ()

### Protected Methods

- virtual OSCL\_IMPORT\_REF void createmempool ()
- virtual OSCL\_IMPORT\_REF void destroymempool ()

### Protected Attributes

- uint32 iNumChunk
- uint32 iChunkSize
- uint32 iChunkSizeMemAligned
- Oscl\_DefAlloc \* iMemPoolAllocator
- OsclAny \* iMemPool
- Oscl\_Vector< OsclAny \*, OsclMemAllocator > iFreeMemChunkList
- bool iCheckNextAvailableFreeChunk
- OsclMemPoolFixedChunkAllocatorObserver \* iObserver
- OsclAny \* iNextAvailableContextData
- int32 iRefCount
- bool iEnableNullPtrReturn

### 6.160.1 Constructor & Destructor Documentation

**6.160.1.1 OSCL\_IMPORT\_REF OsclMemPoolFixedChunkAllocator::OsclMemPoolFixedChunkAllocator (const uint32 numchunk = 1, const uint32 chunkszie = 0, Oscl\_DefAlloc \* gen\_alloc = NULL)**

This API throws an exception when the memory allocation for pool fails If numchunk and chunkszie parameters are not set, memory pool of 1 chunk will be created in the first call to allocate. The chunk size will be set to the n passed in for [allocate\(\)](#). If numchunk parameter is set to 0, the memory pool will use 1 for numchunk.

**Returns:**

void

**6.160.1.2 virtual OSCL\_IMPORT\_REF OsclMemPoolFixedChunkAllocator::~OsclMemPoolFixedChunkAllocator () [virtual]**

The destructor for the memory pool

### 6.160.2 Member Function Documentation

**6.160.2.1 OSCL\_IMPORT\_REF void OsclMemPoolFixedChunkAllocator::addRef ()**

Increments the reference count for this memory pool allocator

**Returns:**

void

**6.160.2.2 virtual OSCL\_IMPORT\_REF OsclAny\* OsclMemPoolFixedChunkAllocator::allocate (const uint32 n) [virtual]**

This API throws an exception when n is greater than the fixed chunk size or there are no free chunk available in the pool, if "enablenullpointerreturn" has not been called. If the memory pool hasn't been created yet, the pool will be created with chunk size equal to n so n must be greater than 0. Exception will be thrown if memory allocation for the memory pool fails.

**Returns:**

pointer to available chunk from memory pool

Implements [Oscl\\_DefAlloc](#).

**6.160.2.3 virtual OSCL\_IMPORT\_REF void OsclMemPoolFixedChunkAllocator::CancelFreeChunkAvailableCallback () [virtual]**

This API will cancel any past callback requests..

**Returns:**

void

**6.160.2.4 virtual OSCL\_IMPORT\_REF void OsclMemPoolFixedChunkAllocator::createmempool()** [protected, virtual]

**6.160.2.5 virtual OSCL\_IMPORT\_REF void OsclMemPoolFixedChunkAllocator::deallocate(  
OsclAny \*p)** [virtual]

This API throws an exception when the pointer p passed in is not part of the memory pool. Exception will be thrown if the memory pool is not set up yet.

**Returns:**

void

Implements [Oscl\\_DefAlloc](#).

**6.160.2.6 virtual OSCL\_IMPORT\_REF void OsclMemPoolFixedChunkAllocator::destroymempool()** [protected, virtual]

**6.160.2.7 virtual OSCL\_IMPORT\_REF void OsclMemPoolFixedChunkAllocator::enablenullpointerreturn()** [virtual]

This API will disable exceptions in case the memory pool runs out of memory Instead of doing "[OSCL\\_LEAVE\(OsclErrNoResources\)](#)" allocate API will return NULL.

**Returns:**

void

**6.160.2.8 virtual OSCL\_IMPORT\_REF void OsclMemPoolFixedChunkAllocator::notifyfreechunkavailable(  
OsclMemPoolFixedChunkAllocatorObserver &obs, OsclAny \*aContextData = NULL)** [virtual]

This API will set the flag to send a callback via specified observer object when the next memory chunk is deallocated by [deallocate\(\)](#) call..

**Returns:**

void

**6.160.2.9 OSCL\_IMPORT\_REF void OsclMemPoolFixedChunkAllocator::removeRef()**

Decrements the reference count for this memory pool allocator When the reference count goes to 0, this instance of the memory pool object is deleted

**Returns:**

void

### 6.160.3 Field Documentation

- 6.160.3.1 **bool OsclMemPoolFixedChunkAllocator::iCheckNextAvailableFreeChunk** [protected]
- 6.160.3.2 **uint32 OsclMemPoolFixedChunkAllocator::iChunkSize** [protected]
- 6.160.3.3 **uint32 OsclMemPoolFixedChunkAllocator::iChunkSizeMemAligned** [protected]
- 6.160.3.4 **bool OsclMemPoolFixedChunkAllocator::iEnableNullPtrReturn** [protected]
- 6.160.3.5 **Oscl\_Vector<OsclAny\*, OsclMemAllocator> OsclMemPoolFixedChunkAllocator::iFreeMemChunkList** [protected]
- 6.160.3.6 **OsclAny\* OsclMemPoolFixedChunkAllocator::iMemPool** [protected]
- 6.160.3.7 **Oscl\_DefAlloc\* OsclMemPoolFixedChunkAllocator::iMemPoolAllocator** [protected]
- 6.160.3.8 **OsclAny\* OsclMemPoolFixedChunkAllocator::iNextAvailableContextData** [protected]
- 6.160.3.9 **uint32 OsclMemPoolFixedChunkAllocator::iNumChunk** [protected]
- 6.160.3.10 **OsclMemPoolFixedChunkAllocatorObserver\* OsclMemPoolFixedChunkAllocator::iObserver** [protected]
- 6.160.3.11 **int32 OsclMemPoolFixedChunkAllocator::iRefCount** [protected]

The documentation for this class was generated from the following file:

- [oscl\\_mem\\_mempool.h](#)

## 6.161 OsclMemPoolFixedChunkAllocatorObserver Class Reference

```
#include <oscl_mem_mempool.h>
```

### Public Methods

- virtual void [freechunkavailable \(OsclAny \\*aContextData\)=0](#)
- virtual [~OsclMemPoolFixedChunkAllocatorObserver \(\)](#)

#### 6.161.1 Constructor & Destructor Documentation

**6.161.1.1** virtual [OsclMemPoolFixedChunkAllocatorObserver::~OsclMemPoolFixedChunkAllocatorObserver \(\) \[inline, virtual\]](#)

#### 6.161.2 Member Function Documentation

**6.161.2.1** virtual void [OsclMemPoolFixedChunkAllocatorObserver::freechunkavailable \(OsclAny \\* aContextData\) \[pure virtual\]](#)

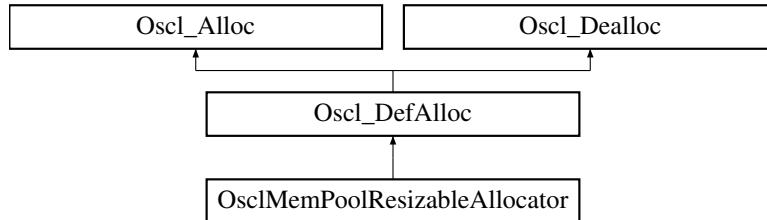
The documentation for this class was generated from the following file:

- [oscl\\_mem\\_mempool.h](#)

## 6.162 OsclMemPoolResizableAllocator Class Reference

```
#include <oscl_mem_mempool.h>
```

Inheritance diagram for OsclMemPoolResizableAllocator::



### Public Methods

- OSCL\_IMPORT\_REF [OsclMemPoolResizableAllocator](#) (uint32 aMemPoolBufferSize, uint32 aMemPoolBufferNumLimit=0, uint32 aExpectedNumBlocksPerBuffer=0, [Oscl\\_DefAlloc](#) \*gen\_alloc=NULL)
- virtual OSCL\_IMPORT\_REF void [enablenullpointerreturn](#) ()
- virtual OSCL\_IMPORT\_REF [OsclAny](#) \* [allocate](#) (const uint32 aNumBytes)
- virtual OSCL\_IMPORT\_REF void [deallocate](#) ([OsclAny](#) \*aPtr)
- virtual OSCL\_IMPORT\_REF bool [trim](#) ([OsclAny](#) \*aPtr, uint32 aBytesToFree)
- OSCL\_IMPORT\_REF uint32 [getBufferSize](#) () const
- virtual OSCL\_IMPORT\_REF uint32 [getAllocatedSize](#) () const
- virtual OSCL\_IMPORT\_REF uint32 [getAvailableSize](#) () const
- virtual OSCL\_IMPORT\_REF uint32 [getLargestContiguousFreeBlockSize](#) () const
- virtual OSCL\_IMPORT\_REF bool [setMaxSzForNewMemPoolBuffer](#) (uint32 aMaxNewMemPoolBufferSz)
- virtual OSCL\_IMPORT\_REF void [notifyfreeblockavailable](#) ([OsclMemPoolResizableAllocatorObserver](#) &aObserver, uint32 aRequestedSize=0, [OsclAny](#) \*aContextData=NULL)
- virtual OSCL\_IMPORT\_REF void [CancelFreeChunkAvailableCallback](#) ()
- virtual OSCL\_IMPORT\_REF void [notifyfreememoryavailable](#) ([OsclMemPoolResizableAllocatorMemoryObserver](#) &aObserver, uint32 aRequestedSize=0, [OsclAny](#) \*aContextData=NULL)
- OSCL\_IMPORT\_REF void [CancelFreeMemoryAvailableCallback](#) ()
- OSCL\_IMPORT\_REF void [addRef](#) ()
- OSCL\_IMPORT\_REF void [removeRef](#) ()

### Protected Methods

- virtual OSCL\_IMPORT\_REF ~[OsclMemPoolResizableAllocator](#) ()
- [MemPoolBufferInfo](#) \* [addnewmempoolbuffer](#) (uint32 aBufferSize)
- void [destroyallmempoolbuffers](#) ()
- [MemPoolBlockInfo](#) \* [findfreeblock](#) (uint32 aBlockSize)
- [OsclAny](#) \* [allocateblock](#) ([MemPoolBlockInfo](#) &aBlockPtr, uint32 aNumBytes)
- void [deallocateblock](#) ([MemPoolBlockInfo](#) &aBlockPtr)
- bool [validateblock](#) ([OsclAny](#) \*aBlockBufPtr)
- uint32 [getMemPoolBufferSize](#) ([MemPoolBufferInfo](#) \*aBufferInfo) const
- uint32 [getMemPoolBufferAllocatedSize](#) ([MemPoolBufferInfo](#) \*aBufferInfo) const
- uint32 [memoryPoolBufferMgmtOverhead](#) () const

## Protected Attributes

- uint32 `iMemPoolBufferSize`
- uint32 `iMemPoolBufferNumLimit`
- uint32 `iExpectedNumBlocksPerBuffer`
- uint32 `iMaxNewMemPoolBufferSz`
- `Oscl_DefAlloc` \* `iMemPoolBufferAllocator`
- `Oscl_Vector< MemPoolBufferInfo *, OsclMemAllocator >` `iMemPoolBufferList`
- uint32 `iBufferInfoAlignedSize`
- uint32 `iBlockInfoAlignedSize`
- bool `iCheckNextAvailable`
- uint32 `iRequestedNextAvailableSize`
- `OsclAny` \* `iNextAvailableContextData`
- `OsclMemPoolResizableAllocatorObserver` \* `iObserver`
- bool `iCheckFreeMemoryAvailable`
- uint32 `iRequestedAvailableFreeMemSize`
- `OsclAny` \* `iFreeMemContextData`
- `OsclMemPoolResizableAllocatorMemoryObserver` \* `iFreeMemPoolObserver`
- int32 `iRefCount`
- bool `iEnableNullPtrReturn`

### 6.162.1 Constructor & Destructor Documentation

**6.162.1.1 OSCL\_IMPORT\_REF OsclMemPoolResizableAllocator::OsclMemPoolResizableAllocator (uint32 *aMemPoolBufferSize*, uint32 *aMemPoolBufferNumLimit* = 0, uint32 *aExpectedNumBlocksPerBuffer* = 0, `Oscl_DefAlloc` \* *gen\_alloc* = NULL)**

Create the memory pool allocator with resizing functionality. The size of the memory pool buffer needs to be passed-in. The maximum number of memory pool buffers, expected number of blocks in a memory pool buffer, and outside allocator are optional. This API throws an exception when the memory allocation for the pool buffer fails. If memory pool buffer number limit parameter is not set, the assumption is that there is no limit and memory pool will grow as needed. If the expected number of blocks is not set or not known, the memory pool will use a default value to 10 to allocate extra memory for the block info header.

**Returns:**

`void`

**6.162.1.2 virtual OSCL\_IMPORT\_REF OsclMemPoolResizableAllocator::~OsclMemPoolResizableAllocator () [protected, virtual]**

The destructor for the memory pool. Should not be called directly. Use `removeRef()` instead.

### 6.162.2 Member Function Documentation

**6.162.2.1 `MemPoolBufferInfo*` OsclMemPoolResizableAllocator::addnewmempoolbuffer (uint32 *aBufferSize*) [protected]**

**6.162.2.2 OSCL\_IMPORT\_REF void OsclMemPoolResizableAllocator::addRef ()**

Increments the reference count for this memory pool allocator

**Returns:**

void

**6.162.2.3 virtual OSCL\_IMPORT\_REF [OsclAny](#)\* OsclMemPoolResizableAllocator::allocate  
(const uint32 aNumBytes) [virtual]**

Allocates a block from the memory pool that is at least in size requested This API throws an exception if there isn't enough memory (if "enablenullpointerreturn" has not been called) for the requested amount in the pool or if the extra pool buffer cannot be allocated.

**Returns:**

Pointer to memory buffer from memory pool

Implements [Oscl\\_DefAlloc](#).

**6.162.2.4 [OsclAny](#)\* OsclMemPoolResizableAllocator::allocateblock ([MemPoolBlockInfo](#) &  
aBlockPtr, uint32 aNumBytes) [protected]****6.162.2.5 virtual OSCL\_IMPORT\_REF void OsclMemPoolResizableAllocator::CancelFree-  
ChunkAvailableCallback () [virtual]**

This API will cancel any past callback requests..

**Returns:**

void

**6.162.2.6 OSCL\_IMPORT\_REF void OsclMemPoolResizableAllocator::CancelFreeMemory-  
AvailableCallback ()****6.162.2.7 virtual OSCL\_IMPORT\_REF void OsclMemPoolResizableAllocator::deallocate  
([OsclAny](#) \* aPtr) [virtual]**

Deallocates and returns a block back to the memory pool This API throws an exception if the pointer passed in is not part of the memory pool, aligned, or has corrupted block header.

**Returns:**

void

Implements [Oscl\\_DefAlloc](#).

**6.162.2.8 void OsclMemPoolResizableAllocator::deallocateblock ([MemPoolBlockInfo](#) &  
aBlockPtr) [protected]****6.162.2.9 void OsclMemPoolResizableAllocator::destroyallmempoolbuffers () [protected]****6.162.2.10 virtual OSCL\_IMPORT\_REF void OsclMemPoolResizable-  
Allocator::enablenullpointerreturn () [virtual]**

This API will disable exceptions in case the memory pool runs out of memory Instead of doing "[OSCL\\_LEAVE\(OsclErrNoResources\)](#)" allocate API will return NULL.

**Returns:**`void`

**6.162.2.11** `MemPoolBlockInfo* OsclMemPoolResizableAllocator::findfreeblock (uint32 aBlockSize) [protected]`

**6.162.2.12** `virtual OSCL_IMPORT_REF uint32 OsclMemPoolResizableAllocator::getAllocatedSize () [virtual]`

Returns the number of bytes allocated from the buffer<including the overhead bytes that may be allocated by the allocator to keep track of the chunks allocated>

**6.162.2.13** `virtual OSCL_IMPORT_REF uint32 OsclMemPoolResizableAllocator::getAvailableSize () [virtual]`

Returns the number of bytes available with the buffer

**6.162.2.14** `OSCL_IMPORT_REF uint32 OsclMemPoolResizableAllocator::getBufferSize ()`

Returns the size of the buffer <including the overhead bytes that may be allocated by the allocator>

**6.162.2.15** `virtual OSCL_IMPORT_REF uint32 OsclMemPoolResizableAllocator::getLargestContiguousFreeBlockSize () [virtual]`

Returns the size of the largest available chunk in the memory.

**6.162.2.16** `uint32 OsclMemPoolResizableAllocator::getMemPoolBufferAllocatedSize (MemPoolBufferInfo * aBufferInfo) const [protected]`

**6.162.2.17** `uint32 OsclMemPoolResizableAllocator::getMemPoolBufferSize (MemPoolBufferInfo * aBufferInfo) const [protected]`

**6.162.2.18** `uint32 OsclMemPoolResizableAllocator::memoryPoolBufferMgmtOverhead () [protected]`

**6.162.2.19** `virtual OSCL_IMPORT_REF void OsclMemPoolResizableAllocator::notifyfreeblockavailable (OsclMemPoolResizableAllocatorObserver & aObserver, uint32 aRequestedSize = 0, OsclAny * aContextData = NULL) [virtual]`

This API will set the flag to send a callback via specified observer object when the next memory block is deallocated by `deallocate()` call. If the optional requested size parameter is set, the callback is sent when a free memory space of requested size becomes available. The optional context data is returned with the callback and can be used by the user to differentiate between different instances of memory pool objects. This memory pool only allows one notify to be queued. Another call to this function will just overwrite the previous call.

**Returns:**`void`

**6.162.2.20** `virtual OSCL_IMPORT_REF void OsclMemPoolResizableAllocator::notifyfreememoryavailable (OsclMemPoolResizableAllocatorMemoryObserver & aObserver, uint32 aRequestedSize = 0, OsclAny * aContextData = NULL)`  
[virtual]

**6.162.2.21** `OSCL_IMPORT_REF void OsclMemPoolResizableAllocator::removeRef ()`

Decrements the reference count for this memory pool allocator When the reference count goes to 0, this instance of the memory pool object is deleted

**Returns:**

`void`

**6.162.2.22** `virtual OSCL_IMPORT_REF bool OsclMemPoolResizableAllocator::setMaxSzForNewMemPoolBuffer (uint32 aMaxNewMemPoolBufferSz)`  
[virtual]

**6.162.2.23** `virtual OSCL_IMPORT_REF bool OsclMemPoolResizableAllocator::trim (OsclAny * aPtr, uint32 aBytesToFree) [virtual]`

Returns a tail segment of a previously allocated memory block back to the memory pool. The passed-in pointer to the memory buffer is still valid after the call completes but the buffer size is smaller by by specified amount that was freed. This function allows the user to allocate a larger size block initially when the amount needed is unknown and then return the unused portion of the block when the amount becomes known. This API throws an exception if the pointer passed in is not part of the memory pool or the size to return is bigger than the size of the passed-in block. Exception will be thrown if the memory pool is not set up yet.

**Returns:**

`bool` True if trim operation successful. False if the block wasn't trimmed

**6.162.2.24** `bool OsclMemPoolResizableAllocator::validateblock (OsclAny * aBlockBufPtr)`  
[protected]

### 6.162.3 Field Documentation

**6.162.3.1** `uint32 OsclMemPoolResizableAllocator::iBlockInfoAlignedSize` [protected]

**6.162.3.2** `uint32 OsclMemPoolResizableAllocator::iBufferInfoAlignedSize` [protected]

**6.162.3.3** `bool OsclMemPoolResizableAllocator::iCheckFreeMemoryAvailable` [protected]

**6.162.3.4** `bool OsclMemPoolResizableAllocator::iCheckNextAvailable` [protected]

**6.162.3.5** `bool OsclMemPoolResizableAllocator::iEnableNullPtrReturn` [protected]

**6.162.3.6** `uint32 OsclMemPoolResizableAllocator::iExpectedNumBlocksPerBuffer`  
[protected]

**6.162.3.7** `OsclAny* OsclMemPoolResizableAllocator::iFreeMemContextData` [protected]

**6.162.3.8** `OsclMemPoolResizableAllocatorMemoryObserver* OsclMemPoolResizable-  
Allocator::iFreeMemPoolObserver` [protected]

**6.162.3.9** `uint32 OsclMemPoolResizableAllocator::iMaxNewMemPoolBufferSz` [protected]

**6.162.3.10** `Oscl_DefAlloc* OsclMemPoolResizableAllocator::iMemPoolBufferAllocator`  
[protected]

**6.162.3.11** `Oscl_Vector<MemPoolBufferInfo*, OsclMemAllocator>`  
`OsclMemPoolResizableAllocator::iMemPoolBufferList` [protected]

**6.162.3.12** `uint32 OsclMemPoolResizableAllocator::iMemPoolBufferNumLimit` [protected]

**6.162.3.13** `uint32 OsclMemPoolResizableAllocator::iMemPoolBufferSize` [protected]

**6.162.3.14** `OsclAny* OsclMemPoolResizableAllocator::iNextAvailableContextData`  
[protected]

**6.162.3.15** `OsclMemPoolResizableAllocatorObserver* OsclMemPoolResizableAllocator::i-  
Observer` [protected]

**6.162.3.16** `int32 OsclMemPoolResizableAllocator::iRefCount` [protected]

**6.162.3.17** `uint32 OsclMemPoolResizableAllocator::iRequestedAvailableFreeMemSize`  
[protected]

**6.162.3.18** `uint32 OsclMemPoolResizableAllocator::iRequestedNextAvailableSize`  
[protected]

The documentation for this class was generated from the following file:

- `oscl_mem_mempool.h`

## 6.163 OsclMemPoolResizableAllocator::MemPoolBlockInfo Struct Reference

```
#include <oscl_mem_mempool.h>
```

### Data Fields

- uint32 iBlockPreFence
- MemPoolBlockInfo \* iNextFreeBlock
- MemPoolBlockInfo \* iPrevFreeBlock
- uint32 iBlockSize
- uint8 \* iBlockBuffer
- MemPoolBufferInfo \* iParentBuffer
- uint32 iBlockPostFence

### 6.163.1 Field Documentation

**6.163.1.1 uint8\* OsclMemPoolResizableAllocator::MemPoolBlockInfo::iBlockBuffer**

**6.163.1.2 uint32 OsclMemPoolResizableAllocator::MemPoolBlockInfo::iBlockPostFence**

**6.163.1.3 uint32 OsclMemPoolResizableAllocator::MemPoolBlockInfo::iBlockPreFence**

**6.163.1.4 uint32 OsclMemPoolResizableAllocator::MemPoolBlockInfo::iBlockSize**

**6.163.1.5 MemPoolBlockInfo\* OsclMemPoolResizableAllocator::MemPoolBlockInfo::iNextFree-Block**

**6.163.1.6 MemPoolBufferInfo\* OsclMemPoolResizableAllocator::MemPoolBlockInfo::iParent-Buffer**

**6.163.1.7 MemPoolBlockInfo\* OsclMemPoolResizableAllocator::MemPoolBlockInfo::iPrevFree-Block**

The documentation for this struct was generated from the following file:

- oscl\_mem\_mempool.h

## 6.164 OsclMemPoolResizableAllocator::MemPoolBufferInfo Struct Reference

```
#include <oscl_mem_mempool.h>
```

### Data Fields

- uint32 iBufferPreFence
- [OsclAny](#) \* iStartAddr
- [OsclAny](#) \* iEndAddr
- uint32 iBufferSize
- uint32 iNumOutstanding
- [MemPoolBlockInfo](#) \* iNextFreeBlock
- uint32 iAllocatedSz
- uint32 iBufferPostFence

#### 6.164.1 Field Documentation

**6.164.1.1 uint32 OsclMemPoolResizableAllocator::MemPoolBufferInfo::iAllocatedSz**

**6.164.1.2 uint32 OsclMemPoolResizableAllocator::MemPoolBufferInfo::iBufferPostFence**

**6.164.1.3 uint32 OsclMemPoolResizableAllocator::MemPoolBufferInfo::iBufferPreFence**

**6.164.1.4 uint32 OsclMemPoolResizableAllocator::MemPoolBufferInfo::iBufferSize**

**6.164.1.5 [OsclAny](#)\* OsclMemPoolResizableAllocator::MemPoolBufferInfo::iEndAddr**

**6.164.1.6 [MemPoolBlockInfo](#)\* OsclMemPoolResizableAllocator::MemPoolBufferInfo::iNextFree-Block**

**6.164.1.7 uint32 OsclMemPoolResizableAllocator::MemPoolBufferInfo::iNumOutstanding**

**6.164.1.8 [OsclAny](#)\* OsclMemPoolResizableAllocator::MemPoolBufferInfo::iStartAddr**

The documentation for this struct was generated from the following file:

- [oscl\\_mem\\_mempool.h](#)

## 6.165 OsclMemPoolResizableAllocatorMemoryObserver Class Reference

```
#include <oscl_mem_mempool.h>
```

### Public Methods

- virtual void [freememoryavailable \(OsclAny \\*aContextData\)=0](#)
- virtual [~OsclMemPoolResizableAllocatorMemoryObserver \(\)](#)

#### 6.165.1 Constructor & Destructor Documentation

**6.165.1.1 virtual OsclMemPoolResizableAllocatorMemoryObserver::~OsclMemPoolResizableAllocatorMemoryObserver () [inline, virtual]**

#### 6.165.2 Member Function Documentation

**6.165.2.1 virtual void OsclMemPoolResizableAllocatorMemoryObserver::freememoryavailable (OsclAny \* aContextData) [pure virtual]**

The documentation for this class was generated from the following file:

- [oscl\\_mem\\_mempool.h](#)

## 6.166 OsclMemPoolResizableAllocatorObserver Class Reference

```
#include <oscl_mem_mempool.h>
```

### Public Methods

- virtual void [freeblockavailable \(OsclAny \\*aContextData\)=0](#)
- virtual [~OsclMemPoolResizableAllocatorObserver \(\)](#)

#### 6.166.1 Constructor & Destructor Documentation

**6.166.1.1** [virtual OsclMemPoolResizableAllocatorObserver::~OsclMemPoolResizableAllocatorObserver \(\) \[inline, virtual\]](#)

#### 6.166.2 Member Function Documentation

**6.166.2.1** [virtual void OsclMemPoolResizableAllocatorObserver::freeblockavailable \(OsclAny \\*\*aContextData\*\) \[pure virtual\]](#)

The documentation for this class was generated from the following file:

- [oscl\\_mem\\_mempool.h](#)

## 6.167 OsclMemStatsNode Class Reference

```
#include <oscl_mem_audit.h>
```

### Public Methods

- [OsclMemStatsNode \(\)](#)
- [void reset \(\)](#)
- [~OsclMemStatsNode \(\)](#)
- [void \\* operator new \(oscl\\_memsize\\_t size\)](#)
- [void \\* operator new \(oscl\\_memsize\\_t size, OsclMemStatsNode \\*ptr\)](#)
- [void operator delete \(void \\*ptr\) throw \(\)](#)

### Data Fields

- [MM\\_Stats\\_t \\* pMMStats](#)
- [MM\\_FailInsertParam \\* pMMFIParam](#)
- [char \\* tag](#)

#### 6.167.1 Constructor & Destructor Documentation

**6.167.1.1 OsclMemStatsNode::OsclMemStatsNode () [inline]**

**6.167.1.2 OsclMemStatsNode::~OsclMemStatsNode () [inline]**

#### 6.167.2 Member Function Documentation

**6.167.2.1 void OsclMemStatsNode::operator delete (void \*ptr) throw () [inline]**

**6.167.2.2 void\* OsclMemStatsNode::operator new (oscl\_memsize\_t size, OsclMemStatsNode \*ptr) [inline]**

**6.167.2.3 void\* OsclMemStatsNode::operator new (oscl\_memsize\_t size) [inline]**

**6.167.2.4 void OsclMemStatsNode::reset () [inline]**

#### 6.167.3 Field Documentation

**6.167.3.1 MM\_FailInsertParam\* OsclMemStatsNode::pMMFIParam**

**6.167.3.2 MM\_Stats\_t\* OsclMemStatsNode::pMMStats**

**6.167.3.3 char\* OsclMemStatsNode::tag**

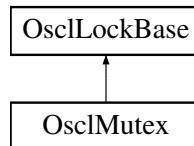
The documentation for this class was generated from the following file:

- [oscl\\_mem\\_audit.h](#)

## 6.168 OsclMutex Class Reference

```
#include <oscl_mutex.h>
```

Inheritance diagram for OsclMutex::



### Public Methods

- OSCL\_IMPORT\_REF OsclMutex ()
- virtual OSCL\_IMPORT\_REF ~OsclMutex ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Create (void)
- OSCL\_IMPORT\_REF void Lock ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError TryLock ()
- OSCL\_IMPORT\_REF void Unlock ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Close (void)

#### 6.168.1 Detailed Description

Class OsclMutex

#### 6.168.2 Constructor & Destructor Documentation

##### 6.168.2.1 OSCL\_IMPORT\_REF OsclMutex::OsclMutex ()

Class constructor

##### 6.168.2.2 virtual OSCL\_IMPORT\_REF OsclMutex::~OsclMutex () [virtual]

Class destructor

#### 6.168.3 Member Function Documentation

##### 6.168.3.1 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclMutex::Close (void)

Closes the Mutex

**Parameters:**

*It* wont take any parameters

**Returns:**

Returns the Error whether it is success or failure. Incase of failure it will return what is the specific error

**6.168.3.2 OSCL\_IMPORT\_REF [OsclProcStatus::eOsclProcError](#) OsclMutex::Create (void)**

Creates the Mutex

**Parameters:**

*No* input arguments

**Returns:**

Returns the Error whether it is success or failure. Incase of failure it will return what is the specific error

**6.168.3.3 OSCL\_IMPORT\_REF void OsclMutex::Lock () [virtual]**

Locks the Mutex

**Parameters:**

*It* wont take any parameters

**Returns:**

Returns nothing

Implements [OsclLockBase](#).

**6.168.3.4 OSCL\_IMPORT\_REF [OsclProcStatus::eOsclProcError](#) OsclMutex::TryLock ()**

Try to lock the mutex,if the Mutex is already locked calling thread immediately returns with out blocking

**Parameters:**

*It* wont take any parameters

**Returns:**

Returns SUCCESS\_ERROR if the mutex was acquired, MUTEX\_LOCKED\_ERROR if the mutex cannot be acquired without waiting, or an error code if the operation failed. Note: this function may not be supported on all platforms, and may return NOT\_IMPLEMENTED.

**6.168.3.5 OSCL\_IMPORT\_REF void OsclMutex::Unlock () [virtual]**

Releases the Mutex

**Parameters:**

*It* wont take any parameters

**Returns:**

Returns nothing

Implements [OsclLockBase](#).

The documentation for this class was generated from the following file:

- [oscl\\_mutex.h](#)

## 6.169 OsclNameString< \_\_len > Class Template Reference

```
#include <oscl_namestring.h>
```

### Public Methods

- [OsclNameString \(\)](#)
- [OsclNameString \(const char a\[ \]\)](#)
- [OsclNameString \(uint8 \\*a\)](#)
- void [Set \(uint8 \\*a\)](#)
- void [Set \(const char a\[ \]\)](#)
- uint8 \* [Str \(\) const](#)
- int32 [MaxLen \(\) const](#)

### 6.169.1 Detailed Description

```
template<int __len> class OsclNameString< __len >
```

Name string class appropriate for passing short constant ASCII strings around. All strings are automatically truncated and null-terminated.

### 6.169.2 Constructor & Destructor Documentation

**6.169.2.1 template<int \_\_len> OsclNameString< \_\_len >::OsclNameString () [inline]**

**6.169.2.2 template<int \_\_len> OsclNameString< \_\_len >::OsclNameString (const char a[ ]) [inline]**

**6.169.2.3 template<int \_\_len> OsclNameString< \_\_len >::OsclNameString (uint8 \* a) [inline]**

### 6.169.3 Member Function Documentation

**6.169.3.1 template<int \_\_len> int32 OsclNameString< \_\_len >::MaxLen () const [inline]**

**6.169.3.2 template<int \_\_len> void OsclNameString< \_\_len >::Set (const char a[ ]) [inline]**

**6.169.3.3 template<int \_\_len> void OsclNameString< \_\_len >::Set (uint8 \* a) [inline]**

Set the string to the input value. The string will be truncated to fit the storage class and automatically null-terminated.

#### Parameters:

*a* (input param): null-terminated character string.

**6.169.3.4 template<int \_\_len> uint8\* OsclNameString< \_\_len >::Str () const [inline]**

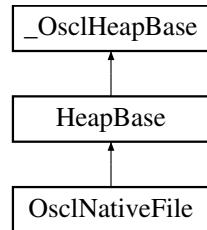
The documentation for this class was generated from the following file:

- [oscl\\_namestring.h](#)

## 6.170 OsclNativeFile Class Reference

```
#include <oscl_file_native.h>
```

Inheritance diagram for OsclNativeFile::



### Public Methods

- `OsclNativeFile ()`
- `~OsclNativeFile ()`
- `int32 Open (const OsclFileHandle &, uint32 mode, const OsclNativeFileParams &params, Oscl_FileServer &fileserv)`
- `int32 Open (const oscl_wchar *filename, uint32 mode, const OsclNativeFileParams &params, Oscl_FileServer &fileserv)`
- `int32 Open (const char *filename, uint32 mode, const OsclNativeFileParams &params, Oscl_FileServer &fileserv)`
- `uint32 Read (OsclAny *buffer, uint32 size, uint32 numelements)`
- `uint32 Write (const OsclAny *buffer, uint32 size, uint32 numelements)`
- `int32 Seek (TOsclFileOffset offset, Oscl_File::seek_type origin)`
- `TOsclFileOffset Tell ()`
- `int32 Flush ()`
- `int32 EndOfFile ()`
- `TOsclFileOffset Size ()`
- `int32 Close ()`
- `uint32 Mode ()`
- `int32 GetError ()`
- `int32 ReadAsync (OsclAny *buffer, uint32 size, uint32 numelements, OsclAOStatus &status)`
- `uint32 GetReadAsyncNumElements ()`
- `bool HasAsyncRead ()`
- `void ReadAsyncCancel ()`

## 6.170.1 Constructor & Destructor Documentation

**6.170.1.1** `OsclNativeFile::OsclNativeFile ()`

**6.170.1.2** `OsclNativeFile::~OsclNativeFile ()`

## 6.170.2 Member Function Documentation

**6.170.2.1** `int32 OsclNativeFile::Close ()`

**6.170.2.2** `int32 OsclNativeFile::EndOfFile ()`

**6.170.2.3** `int32 OsclNativeFile::Flush ()`

**6.170.2.4** `int32 OsclNativeFile::GetError ()`

**6.170.2.5** `uint32 OsclNativeFile::GetReadAsyncNumElements ()`

Get the number of elements read in the last call to ReadAsync. @returns: number of elements read.

**6.170.2.6** `bool OsclNativeFile::HasAsyncRead ()`

@returns: true if async read is supported natively.

**6.170.2.7** `uint32 OsclNativeFile::Mode () [inline]`

**6.170.2.8** `int32 OsclNativeFile::Open (const char *filename, uint32 mode, const OsclNativeFileParams &params, Oscl_FileServer &fileserv)`

**6.170.2.9** `int32 OsclNativeFile::Open (const oscl_wchar *filename, uint32 mode, const OsclNativeFileParams &params, Oscl_FileServer &fileserv)`

**6.170.2.10** `int32 OsclNativeFile::Open (const OsclFileHandle &, uint32 mode, const OsclNativeFileParams &params, Oscl_FileServer &fileserv)`

**6.170.2.11** `uint32 OsclNativeFile::Read (OsclAny *buffer, uint32 size, uint32 numelements)`

**6.170.2.12** `int32 OsclNativeFile::ReadAsync (OsclAny *buffer, uint32 size, uint32 numelements, OsclAOStatus &status)`

Asynchronous read.

### Parameters:

*buffer*: data buffer, must be at least size\*numelements bytes

*size*: size of elements

*numelements*: number of elements to read

*status*: Request status for asynchronous completion @returns: 0 for success.

**6.170.2.13 void OsclNativeFile::ReadAsyncCancel ()**

Cancel any pending async read.

**6.170.2.14 int32 OsclNativeFile::Seek ([TOsclFileOffset](#) *offset*, [Oscl\\_File::seek\\_type](#) *origin*)****6.170.2.15 [TOsclFileOffset](#) OsclNativeFile::Size ()****6.170.2.16 [TOsclFileOffset](#) OsclNativeFile::Tell ()****6.170.2.17 uint32 OsclNativeFile::Write (const [OsclAny](#) \* *buffer*, uint32 *size*, uint32 *numelements*)**

The documentation for this class was generated from the following file:

- [oscl\\_file\\_native.h](#)

## 6.171 OsclNativeFileParams Class Reference

```
#include <oscl_file_types.h>
```

### Public Methods

- [OsclNativeFileParams \(uint32 mode=0, uint32 bufsize=0, uint32 asyncsize=0\)](#)

### Data Fields

- uint32 [iNativeAccessMode](#)
- uint32 [iNativeBufferSize](#)
- uint32 [iAsyncReadBufferSize](#)

#### 6.171.1 Constructor & Destructor Documentation

**6.171.1.1 OsclNativeFileParams::OsclNativeFileParams (uint32 mode = 0, uint32 bufsize = 0, uint32 asyncsize = 0) [inline]**

#### 6.171.2 Field Documentation

**6.171.2.1 uint32 OsclNativeFileParams::iAsyncReadBufferSize**

**6.171.2.2 uint32 OsclNativeFileParams::iNativeAccessMode**

**6.171.2.3 uint32 OsclNativeFileParams::iNativeBufferSize**

The documentation for this class was generated from the following file:

- [oscl\\_file\\_types.h](#)

## 6.172 OsclNetworkAddress Class Reference

```
#include <oscl_socket_types.h>
```

### Public Methods

- [OsclNetworkAddress \(\)](#)
- [OsclNetworkAddress \(const char \\*addr, int p\)](#)
- [bool operator== \(const OsclNetworkAddress &rhs\) const](#)

### Data Fields

- [OsclNameString< PVNETWORKADDRESS\\_LEN > ipAddr](#)
- [int port](#)

#### 6.172.1 Constructor & Destructor Documentation

6.172.1.1 [OsclNetworkAddress::OsclNetworkAddress \(\) \[inline\]](#)

6.172.1.2 [OsclNetworkAddress::OsclNetworkAddress \(const char \\*addr, int p\) \[inline\]](#)

#### 6.172.2 Member Function Documentation

6.172.2.1 [bool OsclNetworkAddress::operator== \(const OsclNetworkAddress & rhs\) const \[inline\]](#)

#### 6.172.3 Field Documentation

6.172.3.1 [OsclNameString<PVNETWORKADDRESS\\_LEN> OsclNetworkAddress::ipAddr](#)

6.172.3.2 [int OsclNetworkAddress::port](#)

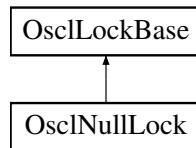
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_types.h](#)

## 6.173 OsclNullLock Class Reference

```
#include <oscl_lock_base.h>
```

Inheritance diagram for OsclNullLock::



### Public Methods

- virtual void [Lock \(\)](#)
- virtual void [Unlock \(\)](#)
- virtual [~OsclNullLock \(\)](#)

#### 6.173.1 Constructor & Destructor Documentation

**6.173.1.1 virtual OsclNullLock::~OsclNullLock () [inline, virtual]**

#### 6.173.2 Member Function Documentation

**6.173.2.1 virtual void OsclNullLock::Lock () [inline, virtual]**

Implements [OsclLockBase](#).

**6.173.2.2 virtual void OsclNullLock::Unlock () [inline, virtual]**

Implements [OsclLockBase](#).

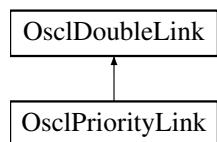
The documentation for this class was generated from the following file:

- [oscl\\_lock\\_base.h](#)

## 6.174 OsclPriorityLink Class Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OsclPriorityLink::



### Data Fields

- int32 [iPriority](#)

#### 6.174.1 Field Documentation

##### 6.174.1.1 int32 OsclPriorityLink::iPriority

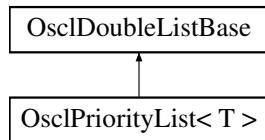
The documentation for this class was generated from the following file:

- [oscl\\_double\\_list.h](#)

## 6.175 OsclPriorityList< T > Class Template Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OsclPriorityList< T >::



### Public Methods

- OSCL\_INLINE OsclPriorityList ()
- OSCL\_INLINE OsclPriorityList (int32 anOffset)
- OSCL\_INLINE void Insert (T &aRef)
- OSCL\_INLINE bool IsHead (const T \*aPtr) const
- OSCL\_INLINE bool IsTail (const T \*aPtr) const
- OSCL\_INLINE T \* Head () const
- OSCL\_INLINE T \* Tail () const

```
template<class T> class OsclPriorityList< T >
```

#### 6.175.1 Constructor & Destructor Documentation

**6.175.1.1 template<class T> OSCL\_INLINE OsclPriorityList< T >::OsclPriorityList ()**

**6.175.1.2 template<class T> OSCL\_INLINE OsclPriorityList< T >::OsclPriorityList (int32 *anOffset*)**

#### 6.175.2 Member Function Documentation

**6.175.2.1 template<class T> OSCL\_INLINE T\* OsclPriorityList< T >::Head ()**

**6.175.2.2 template<class T> OSCL\_INLINE void OsclPriorityList< T >::Insert (T &*aRef*)**

**6.175.2.3 template<class T> OSCL\_INLINE bool OsclPriorityList< T >::IsHead (const T \**aPtr*) const**

**6.175.2.4 template<class T> OSCL\_INLINE bool OsclPriorityList< T >::IsTail (const T \**aPtr*) const**

**6.175.2.5 template<class T> OSCL\_INLINE T\* OsclPriorityList< T >::Tail ()**

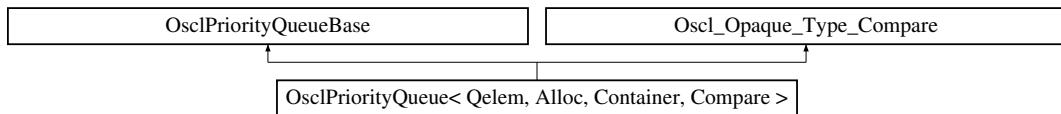
The documentation for this class was generated from the following file:

- [oscl\\_double\\_list.h](#)

## 6.176 OsclPriorityQueue< Qelem, Alloc, Container, Compare > Class Template Reference

```
#include <oscl_priqueue.h>
```

Inheritance diagram for OsclPriorityQueue< Qelem, Alloc, Container, Compare >::



### Public Types

- `typedef Container::value_type value_type`
- `typedef Container container_type`
- `typedef Container::iterator iterator`
- `typedef Container::const_reference const_reference`

### Public Methods

- `bool empty () const`
- `uint32 size () const`
- `void reserve (uint32 n)`
- `const_reference top () const`
- `const Container & vec ()`
- `void push (const value_type &input)`
- `void pop ()`
- `int remove (const value_type &input)`
- `OsclPriorityQueue ()`
- `virtual ~OsclPriorityQueue ()`

### Protected Methods

- `void push_heap (iterator first, iterator last)`
- `void pop_heap (iterator first, iterator last)`
- `iterator find_heap (const value_type &input, iterator first, iterator last)`
- `int validate ()`
- `void swap (OsclAny *dest, const OsclAny *src)`
- `int compare_LT (OsclAny *a, OsclAny *b) const`
- `int compare_EQ (const OsclAny *a, const OsclAny *b) const`

### Protected Attributes

- `Container c`
- `Compare comp`

## Friends

- class [oscl\\_priqueue\\_test](#)

```
template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> class OsclPriorityQueue< Qelem, Alloc, Container, Compare >
```

### 6.176.1 Member Typedef Documentation

- 6.176.1.1** template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **typedef Container::const\_reference OsclPriorityQueue< Qelem, Alloc, Container, Compare >::const\_reference**
- 6.176.1.2** template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **typedef Container OsclPriorityQueue< Qelem, Alloc, Container, Compare >::container\_type**
- 6.176.1.3** template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **typedef Container::iterator OsclPriorityQueue< Qelem, Alloc, Container, Compare >::iterator**
- 6.176.1.4** template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **typedef Container::value\_type OsclPriorityQueue< Qelem, Alloc, Container, Compare >::value\_type**

### 6.176.2 Constructor & Destructor Documentation

- 6.176.2.1** template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **OsclPriorityQueue< Qelem, Alloc, Container, Compare >::OsclPriorityQueue () [inline]**
- 6.176.2.2** template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **virtual OsclPriorityQueue< Qelem, Alloc, Container, Compare >::~OsclPriorityQueue () [inline, virtual]**

### 6.176.3 Member Function Documentation

- 6.176.3.1** template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **int OsclPriorityQueue< Qelem, Alloc, Container, Compare >::compare\_EQ (const OsclAny \* a, const OsclAny \* b) const [inline, protected, virtual]**

Return a==b.

Implements [Oscl\\_Opaque\\_Type\\_Compare](#).

- 6.176.3.2** template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **int OsclPriorityQueue< Qelem, Alloc, Container, Compare >::compare\_LT (OsclAny \* a, OsclAny \* b) const [inline, protected, virtual]**

Return a<b.

Implements [Oscl\\_Opaque\\_Type\\_Compare](#).

- 6.176.3.3 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> bool OsclPriorityQueue< Qelem, Alloc, Container, Compare >::empty () const [inline]`
- 6.176.3.4 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> iterator OsclPriorityQueue< Qelem, Alloc, Container, Compare >::find_heap (const value_type & input, iterator first, iterator last) [inline, protected]`
- 6.176.3.5 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> void OsclPriorityQueue< Qelem, Alloc, Container, Compare >::pop () [inline]`
- 6.176.3.6 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> void OsclPriorityQueue< Qelem, Alloc, Container, Compare >::pop_heap (iterator first, iterator last) [inline, protected]`
- 6.176.3.7 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> void OsclPriorityQueue< Qelem, Alloc, Container, Compare >::push (const value_type & input) [inline]`
- 6.176.3.8 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> void OsclPriorityQueue< Qelem, Alloc, Container, Compare >::push_heap (iterator first, iterator last) [inline, protected]`
- 6.176.3.9 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> int OsclPriorityQueue< Qelem, Alloc, Container, Compare >::remove (const value_type & input) [inline]`
- 6.176.3.10 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> void OsclPriorityQueue< Qelem, Alloc, Container, Compare >::reserve (uint32 n) [inline]`
- 6.176.3.11 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> uint32 OsclPriorityQueue< Qelem, Alloc, Container, Compare >::size () const [inline]`
- 6.176.3.12 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> void OsclPriorityQueue< Qelem, Alloc, Container, Compare >::swap (OsclAny * dest, const OsclAny * src) [inline, protected, virtual]`

Swap element at "a" with element at "b". Both pointers must be non-NULL.

Implements [Oscl\\_Opaque\\_Type\\_Compare](#).

## 6.176 OsclPriorityQueue< Qelem, Alloc, Container, Compare > Class Template Reference

- 6.176.3.13 template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **const\_reference** OsclPriorityQueue< Qelem, Alloc, Container, Compare >::top () const [inline]
- 6.176.3.14 template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> int OsclPriorityQueue< Qelem, Alloc, Container, Compare >::validate () [inline, protected]
- 6.176.3.15 template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **const Container&** OsclPriorityQueue< Qelem, Alloc, Container, Compare >::vec () [inline]

### 6.176.4 Friends And Related Function Documentation

- 6.176.4.1 template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> friend class oscl\_priqueue\_test [friend]

### 6.176.5 Field Documentation

- 6.176.5.1 template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> Container OsclPriorityQueue< Qelem, Alloc, Container, Compare >::c [protected]
- 6.176.5.2 template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> Compare OsclPriorityQueue< Qelem, Alloc, Container, Compare >::comp [protected]

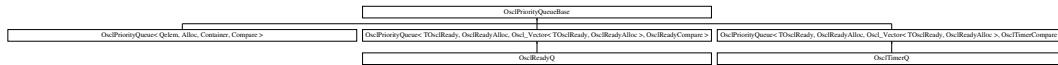
The documentation for this class was generated from the following file:

- [oscl\\_priqueue.h](#)

## 6.177 OsclPriorityQueueBase Class Reference

```
#include <oscl_priqueue.h>
```

Inheritance diagram for OsclPriorityQueueBase::



### Protected Methods

- virtual ~OsclPriorityQueueBase ()
- OSCL\_IMPORT\_REF void [push\\_heap](#) (OsclAny \*first, OsclAny \*last)
- OSCL\_IMPORT\_REF void [pop\\_heap](#) (OsclAny \*first, OsclAny \*last)
- OSCL\_IMPORT\_REF OsclAny \* [find\\_heap](#) (const OsclAny \*input, OsclAny \*first, OsclAny \*last)
- OSCL\_IMPORT\_REF int [remove](#) (const OsclAny \*input)
- void [construct](#) (Oscl\_Opaque\_Type\_Compare \*ot, Oscl\_Vector\_Base \*vec)

### 6.177.1 Detailed Description

OsclPriorityQueueBase is a non-templated base class for [OsclPriorityQueue](#). The purpose of this base class is to avoid large inline routines in the [OsclPriorityQueue](#) implementation. This class is not intended for direct instantiation except by [OsclPriorityQueue](#).

### 6.177.2 Constructor & Destructor Documentation

**6.177.2.1 virtual OsclPriorityQueueBase::~OsclPriorityQueueBase ()** [inline, protected, virtual]

### 6.177.3 Member Function Documentation

**6.177.3.1 void OsclPriorityQueueBase::construct (Oscl\_Opaque\_Type\_Compare \* ot, Oscl\_Vector\_Base \* vec)** [inline, protected]

**6.177.3.2 OSCL\_IMPORT\_REF OsclAny\* OsclPriorityQueueBase::find\_heap (const OsclAny \* input, OsclAny \* first, OsclAny \* last)** [protected]

**6.177.3.3 OSCL\_IMPORT\_REF void OsclPriorityQueueBase::pop\_heap (OsclAny \* first, OsclAny \* last)** [protected]

**6.177.3.4 OSCL\_IMPORT\_REF void OsclPriorityQueueBase::push\_heap (OsclAny \* first, OsclAny \* last)** [protected]

**6.177.3.5 OSCL\_IMPORT\_REF int OsclPriorityQueueBase::remove (const OsclAny \* input)** [protected]

The documentation for this class was generated from the following file:

- [oscl\\_priqueue.h](#)

## 6.178 OsclProcStatus Class Reference

```
#include <oscl_procstatus.h>
```

### Public Types

- enum `eOsclProcError` { `SUCCESS_ERROR` = 0, `OTHER_ERROR`, `TOO_MANY_THREADS_ERROR`, `BAD_THREADID_ADDR_ERROR`, `MAX_THRDS_REACHED_ERROR`, `INVALID_THREAD_ID_ERROR`, `NOT_ENOUGH_MEMORY_ERROR`, `OUTOFMEMORY_ERROR`, `NOT_ENOUGH_RESOURCES_ERROR`, `THREAD_1_INACTIVE_ERROR`, `ALREADY_SUSPENDED_ERROR`, `NOT_SUSPENDED_ERROR`, `INVALID_THREAD_ERROR`, `INVALID_PARAM_ERROR`, `NO_PERMISSION_ERROR`, `INVALID_PRIORITY_ERROR`, `PSHARED_NOT_ZERO_ERROR`, `EXCEED_MAX_COUNT_VARIABLE_ERROR`, `THREAD_BLOCK_ERROR`, `EXCEED_MAX_SEM_COUNT_ERROR`, `INVALID_HANDLE_ERROR`, `INVALID_OPERATION_ERROR`, `INVALID_FUNCTION_ERROR`, `INVALID_ACCESS_ERROR`, `INVALID_ARGUMENT_ERROR`, `SYSTEM_RESOURCES_UNAVAILABLE_ERROR`, `INVALID_POINTER_ERROR`, `RELOCK_MUTEX_ERROR`, `THREAD_NOT_OWN_MUTEX_ERROR`, `MUTEX_LOCKED_ERROR`, `WAIT_ABANDONED_ERROR`, `WAIT_TIMEOUT_ERROR`, `SEM_NOT_SIGNALED_ERROR`, `PSHARED_ATTRIBUTE_SETTING_ERROR`, `NOT_IMPLEMENTED` }

### 6.178.1 Detailed Description

Class OsclProcStatus

### 6.178.2 Member Enumeration Documentation

#### 6.178.2.1 enum OsclProcStatus::eOsclProcError

List of enums which contain error codes

Enumeration values:

`SUCCESS_ERROR`  
`OTHER_ERROR`  
`TOO_MANY_THREADS_ERROR`  
`BAD_THREADID_ADDR_ERROR`  
`MAX_THRDS_REACHED_ERROR`  
`INVALID_THREAD_ID_ERROR`  
`NOT_ENOUGH_MEMORY_ERROR`  
`OUTOFMEMORY_ERROR`  
`NOT_ENOUGH_RESOURCES_ERROR`  
`THREAD_1_INACTIVE_ERROR`  
`ALREADY_SUSPENDED_ERROR`  
`NOT_SUSPENDED_ERROR`  
`INVALID_THREAD_ERROR`  
`INVALID_PARAM_ERROR`  
`NO_PERMISSION_ERROR`

**INVALID\_PRIORITY\_ERROR**  
**PSHARED\_NOT\_ZERO\_ERROR**  
**EXCEED\_MAX\_COUNT\_VARIABLE\_ERROR**  
**THREAD\_BLOCK\_ERROR**  
**EXCEED\_MAX\_SEM\_COUNT\_ERROR**  
**INVALID\_HANDLE\_ERROR**  
**INVALID\_OPERATION\_ERROR**  
**INVALID\_FUNCTION\_ERROR**  
**INVALID\_ACCESS\_ERROR**  
**INVALID\_ARGUMENT\_ERROR**  
**SYSTEM\_RESOURCES\_UNAVAILABLE\_ERROR**  
**INVALID\_POINTER\_ERROR**  
**RELOCK\_MUTEX\_ERROR**  
**THREAD\_NOT\_OWN\_MUTEX\_ERROR**  
**MUTEX\_LOCKED\_ERROR**  
**WAIT\_ABANDONED\_ERROR**  
**WAIT\_TIMEOUT\_ERROR**  
**SEM\_NOT\_SIGNALED\_ERROR**  
**PSHARED\_ATTRIBUTE\_SETTING\_ERROR**  
**NOT\_IMPLEMENTED**

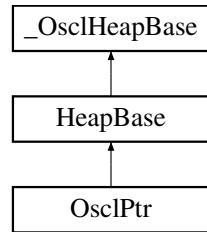
The documentation for this class was generated from the following file:

- [oscl\\_procstatus.h](#)

## 6.179 OsclPtr Class Reference

```
#include <oscl_file_async_read.h>
```

Inheritance diagram for OsclPtr::



### Public Methods

- [OsclPtr \(uint8 \\*ptr, int32 &len, int32 max\)](#)
- [OsclPtr \(const OsclPtr &d\)](#)
- [uint8 \\* Ptr \(\)](#)
- [void SetLength \(int32 l\)](#)
- [int32 Length \(\)](#)
- [void Zero \(\)](#)
- [void Set \(OsclPtr &v\)](#)
- [void Set \(uint8 \\*ptr, int32 len, int32 max\)](#)
- [void Append \(OsclPtrC &v\)](#)

#### 6.179.1 Constructor & Destructor Documentation

**6.179.1.1 OsclPtr::OsclPtr (uint8 \*ptr, int32 &len, int32 max) [inline]**

**6.179.1.2 OsclPtr::OsclPtr (const OsclPtr &d) [inline]**

#### 6.179.2 Member Function Documentation

**6.179.2.1 void OsclPtr::Append (OsclPtrC &v) [inline]**

**6.179.2.2 int32 OsclPtr::Length () [inline]**

**6.179.2.3 uint8\* OsclPtr::Ptr () [inline]**

**6.179.2.4 void OsclPtr::Set (uint8 \*ptr, int32 len, int32 max) [inline]**

**6.179.2.5 void OsclPtr::Set (OsclPtr &v) [inline]**

**6.179.2.6 void OsclPtr::SetLength (int32 l) [inline]**

**6.179.2.7 void OsclPtr::Zero () [inline]**

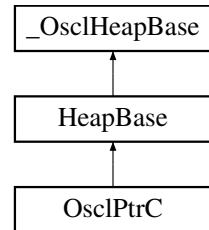
The documentation for this class was generated from the following file:

- [oscl\\_file\\_async\\_read.h](#)

## 6.180 OsclPtrC Class Reference

```
#include <oscl_file_async_read.h>
```

Inheritance diagram for OsclPtrC::



### Public Methods

- [OsclPtrC](#) (const uint8 \*ptr, int32 len, int32 max)
- [OsclPtrC](#) (const OsclPtrC &d)
- const uint8 \* [Ptr](#) ()
- void [SetLength](#) (int32 l)
- int32 [Length](#) ()
- void [Zero](#) ()
- void [Set](#) (OsclPtrC \*v)
- void [Set](#) (uint8 \*ptr, int32 len, int32 max)
- OsclPtrC [Right](#) (int32 size)
- OsclPtrC [Left](#) (int32 size)

## 6.180.1 Constructor & Destructor Documentation

**6.180.1.1** `OsclPtrC::OsclPtrC (const uint8 *ptr, int32 len, int32 max)` [inline]

**6.180.1.2** `OsclPtrC::OsclPtrC (const OsclPtrC & d)` [inline]

## 6.180.2 Member Function Documentation

**6.180.2.1** `OsclPtrC OsclPtrC::Left (int32 size)` [inline]

**6.180.2.2** `int32 OsclPtrC::Length ()` [inline]

**6.180.2.3** `const uint8* OsclPtrC::Ptr ()` [inline]

**6.180.2.4** `OsclPtrC OsclPtrC::Right (int32 size)` [inline]

**6.180.2.5** `void OsclPtrC::Set (uint8 *ptr, int32 len, int32 max)` [inline]

**6.180.2.6** `void OsclPtrC::Set (OsclPtrC *v)` [inline]

**6.180.2.7** `void OsclPtrC::SetLength (int32 l)` [inline]

**6.180.2.8** `void OsclPtrC::Zero ()` [inline]

The documentation for this class was generated from the following file:

- [oscl\\_file\\_async\\_read.h](#)

## 6.181 OsclRand Class Reference

```
#include <oscl_rand.h>
```

### Public Methods

- OSCL\_COND\_IMPORT\_REF void [Seed](#) (int32 seed)
- OSCL\_COND\_IMPORT\_REF int32 [Rand](#) ()

#### 6.181.1 Member Function Documentation

**6.181.1.1 OSCL\_COND\_IMPORT\_REF int32 OsclRand::Rand ()**

**6.181.1.2 OSCL\_COND\_IMPORT\_REF void OsclRand::Seed (int32 *seed*)**

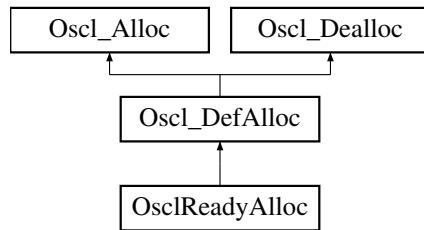
The documentation for this class was generated from the following file:

- [oscl\\_rand.h](#)

## 6.182 OsclReadyAlloc Class Reference

```
#include <oscl_scheduler_readyq.h>
```

Inheritance diagram for OsclReadyAlloc::



### Public Methods

- [OsclAny \\* allocate \(const uint32 size\)](#)
- [OsclAny \\* allocate\\_fl \(const uint32 size, const char \\*file\\_name, const int line\\_num\)](#)
- void [deallocate \(OsclAny \\*p\)](#)

#### 6.182.1 Member Function Documentation

##### 6.182.1.1 [OsclAny\\* OsclReadyAlloc::allocate \(const uint32 size\) \[virtual\]](#)

Implements [Oscl\\_DefAlloc](#).

##### 6.182.1.2 [OsclAny\\* OsclReadyAlloc::allocate\\_fl \(const uint32 size, const char \\*file\\_name, const int line\\_num\) \[virtual\]](#)

Reimplemented from [Oscl\\_DefAlloc](#).

##### 6.182.1.3 [void OsclReadyAlloc::deallocate \(OsclAny \\*p\) \[virtual\]](#)

Implements [Oscl\\_DefAlloc](#).

The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_readyq.h](#)

## 6.183 OsclReadyCompare Class Reference

```
#include <oscl_scheduler_readyq.h>
```

### Static Public Methods

- int [compare \(TOsclReady &a, TOsclReady &b\)](#)

#### 6.183.1 Member Function Documentation

##### 6.183.1.1 int OsclReadyCompare::compare (TOsclReady &*a*, TOsclReady &*b*) [static]

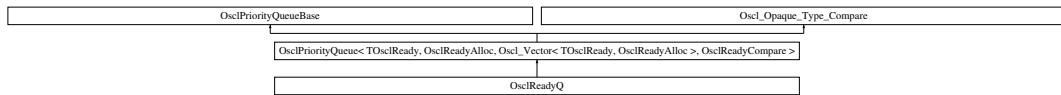
The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_readyq.h](#)

## 6.184 OsclReadyQ Class Reference

```
#include <oscl_scheduler_readyq.h>
```

Inheritance diagram for OsclReadyQ::



### Public Methods

- void [Construct](#) (int)
- void [ThreadLogon](#) ()
- void [ThreadLogoff](#) ()
- void [Remove](#) (TOscIReady)
- bool [IsIn](#) (TOscIReady)
- uint32 [Depth](#) ()
- [TOscIReady PopTop](#) ()
- [TOscIReady Top](#) ()
- [TOscIReady WaitAndPopTop](#) ()
- [TOscIReady WaitAndPopTop](#) (uint32)
- int32 [PendComplete](#) (PVActiveBase \*pvbase, int32 aReason)
- int32 [WaitForRequestComplete](#) (PVActiveBase \*)
- void [RegisterForCallback](#) (OsclSchedulerObserver \*aCallback, OsclAny \*aCallbackContext)
- void [TimerCallback](#) (uint32 aDelayMicrosec)
- [OsclSchedulerObserver \\* Callback](#) ()

## 6.184.1 Member Function Documentation

6.184.1.1 **OsclSchedulerObserver\*** OsclReadyQ::Callback () [inline]

6.184.1.2 void OsclReadyQ::Construct (int)

6.184.1.3 uint32 OsclReadyQ::Depth () [inline]

6.184.1.4 bool OsclReadyQ::IsIn (**TOsclReady**)

6.184.1.5 int32 OsclReadyQ::PendComplete (**PVActiveBase** \**pvbase*, int32 *aReason*)

6.184.1.6 **TOsclReady** OsclReadyQ::PopTop ()

6.184.1.7 void OsclReadyQ::RegisterForCallback (**OsclSchedulerObserver** \**aCallback*, **OsclAny** \**aCallbackContext*)

6.184.1.8 void OsclReadyQ::Remove (**TOsclReady**)

6.184.1.9 void OsclReadyQ::ThreadLogoff ()

6.184.1.10 void OsclReadyQ::ThreadLogon ()

6.184.1.11 void OsclReadyQ::TimerCallback (uint32 *aDelayMicrosec*)

6.184.1.12 **TOsclReady** OsclReadyQ::Top ()

6.184.1.13 **TOsclReady** OsclReadyQ::WaitAndPopTop (uint32)

6.184.1.14 **TOsclReady** OsclReadyQ::WaitAndPopTop ()

6.184.1.15 int32 OsclReadyQ::WaitForRequestComplete (**PVActiveBase** \*)

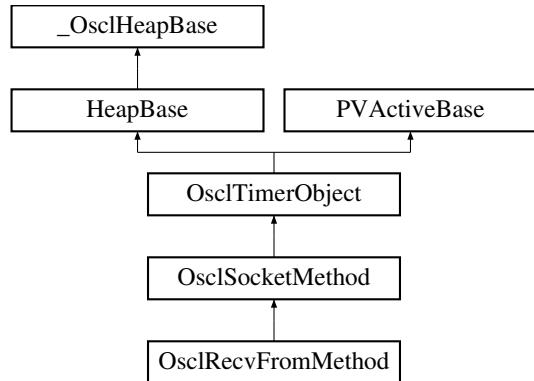
The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_readyq.h](#)

## 6.185 OsclRecvFromMethod Class Reference

```
#include <oscl_socket_recv_from.h>
```

Inheritance diagram for OsclRecvFromMethod:::



### Public Methods

- [`~OsclRecvFromMethod \(\)`](#)
- [`TPVSocketEvent RecvFrom \(uint8 \*&aPtr, uint32 aMaxLen, OsclNetworkAddress &aAddress, int32 aTimeout, uint32 aMultiMax, Oscl\_Vector< uint32, OsclMemAllocator > \*aPacketLen, Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator > \*aPacketSource\)`](#)
- [`uint8 \* GetRecvData \(int32 \*aLength\)`](#)
- [`OsclRecvFromRequest \* RecvFromRequest \(\)`](#)

### Static Public Methods

- [`OsclRecvFromMethod \* NewL \(OsclIPSocketI &c\)`](#)

#### 6.185.1 Constructor & Destructor Documentation

##### 6.185.1.1 OsclRecvFromMethod::~OsclRecvFromMethod ()

#### 6.185.2 Member Function Documentation

##### 6.185.2.1 `uint8* OsclRecvFromMethod::GetRecvData (int32 * aLength)`

##### 6.185.2.2 `OsclRecvFromMethod* OsclRecvFromMethod::NewL (OsclIPSocketI & c)` [static]

##### 6.185.2.3 `TPVSocketEvent OsclRecvFromMethod::RecvFrom (uint8 *& aPtr, uint32 aMaxLen, OsclNetworkAddress & aAddress, int32 aTimeout, uint32 aMultiMax, Oscl_Vector< uint32, OsclMemAllocator > * aPacketLen, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > * aPacketSource)`

##### 6.185.2.4 `OsclRecvFromRequest* OsclRecvFromMethod::RecvFromRequest () [inline]`

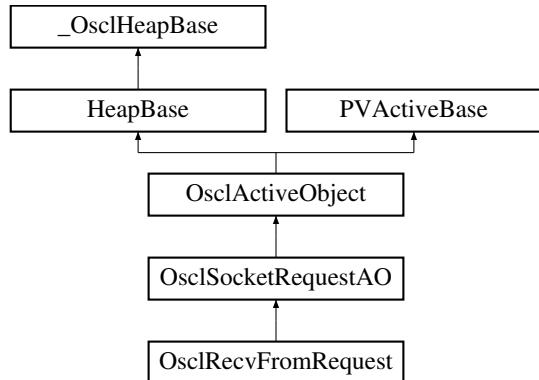
The documentation for this class was generated from the following file:

- 
- [oscl\\_socket\\_recv\\_from.h](#)

## 6.186 OsclRecvFromRequest Class Reference

```
#include <oscl_socket_recv_from.h>
```

Inheritance diagram for OsclRecvFromRequest::



### Public Methods

- `uint8 * GetRecvData (int32 *aLength)`
- `OsclRecvFromRequest (OsclSocketMethod &c)`
- `void RecvFrom (uint8 *&aPtr, uint32 aMaxLen, OsclNetworkAddress &aAddress, uint32 aMultiMax, Oscl_Vector< uint32, OsclMemAllocator > *aPacketLen, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > *aPacketSource)`
- `void Success ()`

#### 6.186.1 Detailed Description

This is the AO that interacts with the socket server

#### 6.186.2 Constructor & Destructor Documentation

**6.186.2.1 OsclRecvFromRequest::OsclRecvFromRequest (`OsclSocketMethod &c`) [inline]**

#### 6.186.3 Member Function Documentation

**6.186.3.1 `uint8* OsclRecvFromRequest::GetRecvData (int32 * aLength)`**

**6.186.3.2 `void OsclRecvFromRequest::RecvFrom (uint8 *& aPtr, uint32 aMaxLen, OsclNetworkAddress & aAddress, uint32 aMultiMax, Oscl_Vector< uint32, OsclMemAllocator > * aPacketLen, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > * aPacketSource)`**

**6.186.3.3 `void OsclRecvFromRequest::Success () [virtual]`**

Reimplemented from `OsclSocketRequestAO`.

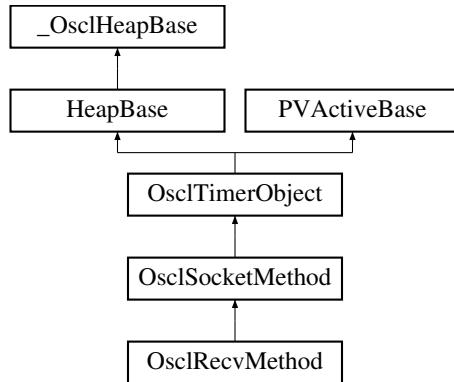
The documentation for this class was generated from the following file:

- 
- [oscl\\_socket\\_recv\\_from.h](#)

## 6.187 OsclRecvMethod Class Reference

```
#include <oscl_socket_recv.h>
```

Inheritance diagram for OsclRecvMethod::



### Public Methods

- [~OsclRecvMethod \(\)](#)
- [TPVSocketEvent Recv \(uint8 \\*&aPtr, uint32 aMaxLen, int32 aTimeout\)](#)
- [uint8 \\* GetRecvData \(int32 \\*aLength\)](#)
- [OsclRecvRequest \\* RecvRequest \(\)](#)

### Static Public Methods

- [OsclRecvMethod \\* NewL \(OsclIPSocketI &c\)](#)

#### 6.187.1 Constructor & Destructor Documentation

##### 6.187.1.1 OsclRecvMethod::~OsclRecvMethod ()

#### 6.187.2 Member Function Documentation

##### 6.187.2.1 uint8\* OsclRecvMethod::GetRecvData (int32 \* aLength)

##### 6.187.2.2 OsclRecvMethod\* OsclRecvMethod::NewL (OsclIPSocketI & c) [static]

##### 6.187.2.3 TPVSocketEvent OsclRecvMethod::Recv (uint8 \*& aPtr, uint32 aMaxLen, int32 aTimeout)

##### 6.187.2.4 OsclRecvRequest\* OsclRecvMethod::RecvRequest () [inline]

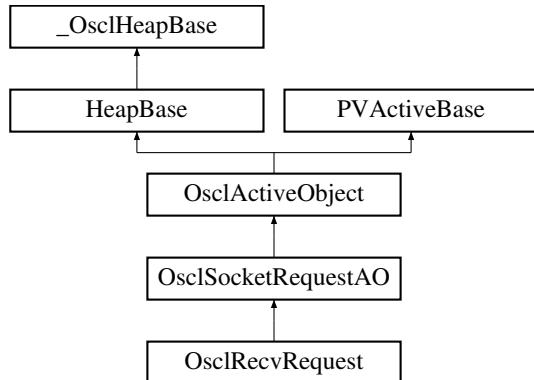
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_recv.h](#)

## 6.188 OsclRecvRequest Class Reference

```
#include <oscl_socket_recv.h>
```

Inheritance diagram for OsclRecvRequest::



### Public Methods

- `uint8 * GetRecvData (int32 *aLength)`
- `OsclRecvRequest (OsclSocketMethod &c)`
- `void Recv (uint8 *&aPtr, uint32 aMaxLen)`
- `void Success ()`

#### 6.188.1 Detailed Description

This is the AO that interacts with the socket server

#### 6.188.2 Constructor & Destructor Documentation

**6.188.2.1 OsclRecvRequest::OsclRecvRequest (`OsclSocketMethod &c`) [inline]**

#### 6.188.3 Member Function Documentation

**6.188.3.1 `uint8* OsclRecvRequest::GetRecvData (int32 * aLength)`**

**6.188.3.2 `void OsclRecvRequest::Recv (uint8 *& aPtr, uint32 aMaxLen)`**

**6.188.3.3 `void OsclRecvRequest::Success () [virtual]`**

Reimplemented from `OsclSocketRequestAO`.

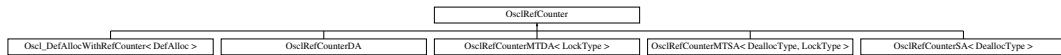
The documentation for this class was generated from the following file:

- `oscl_socket_recv.h`

## 6.189 OsclRefCounter Class Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for OsclRefCounter:::



### Public Methods

- virtual void `addRef ()=0`
- virtual void `removeRef ()=0`
- virtual uint32 `getCount ()=0`
- virtual `~OsclRefCounter ()`

#### 6.189.1 Detailed Description

Interface class for OsclRefCounter implementations

#### 6.189.2 Constructor & Destructor Documentation

**6.189.2.1 virtual OsclRefCounter::~OsclRefCounter () [inline, virtual]**

#### 6.189.3 Member Function Documentation

**6.189.3.1 virtual void OsclRefCounter::addRef () [pure virtual]**

Add to the reference count

Implemented in `OsclRefCounterDA`, `OsclRefCounterSA< DeallocType >`, `OsclRefCounterMTDA< LockType >`, `OsclRefCounterMTSA< DeallocType, LockType >`, and `Oscl_RefAllocWithRefCounter< DefAlloc >`.

**6.189.3.2 virtual uint32 OsclRefCounter::getCount () [pure virtual]**

Gets the current number of references

Implemented in `OsclRefCounterDA`, `OsclRefCounterSA< DeallocType >`, `OsclRefCounterMTDA< LockType >`, `OsclRefCounterMTSA< DeallocType, LockType >`, and `Oscl_RefAllocWithRefCounter< DefAlloc >`.

**6.189.3.3 virtual void OsclRefCounter::removeRef () [pure virtual]**

Delete from reference count

Implemented in `OsclRefCounterDA`, `OsclRefCounterSA< DeallocType >`, `OsclRefCounterMTDA< LockType >`, `OsclRefCounterMTSA< DeallocType, LockType >`, and `Oscl_RefAllocWithRefCounter< DefAlloc >`.

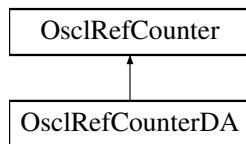
The documentation for this class was generated from the following file:

- 
- [oscl\\_refcounter.h](#)

## 6.190 OsclRefCounterDA Class Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for OsclRefCounterDA::



### Public Methods

- [OsclRefCounterDA \(OsclAny \\*p, OsclDestructDealloc \\*dealloc\)](#)
- virtual [~OsclRefCounterDA \(\)](#)
- void [addRef \(\)](#)
- void [removeRef \(\)](#)
- uint32 [getCount \(\)](#)

#### 6.190.1 Detailed Description

Implementation of an [OsclRefCounter](#) that uses a dynamically created deallocator.

#### 6.190.2 Constructor & Destructor Documentation

##### 6.190.2.1 OsclRefCounterDA::OsclRefCounterDA ([OsclAny \\*p](#), [OsclDestructDealloc \\*dealloc](#)) [inline]

Constructor Takes a pointer to the buffer to track, and a pointer to the deallocator object which will be used to delete the buffer.

When the reference count reaches zero, the buffer will be deleted by the deallocator. Also, the [OsclRefCounter](#) object (this) will self-destruct when the reference count is zero. In some cases the [OsclRefCounter](#) object will be part of the buffer being deleted. For such cases, the object pointer must be equal to the buffer pointer given at construction. If the object is not part of the buffer being deleted, it will self-destruct with a call to `delete()`.

##### Parameters:

*p* pointer to the buffer to track

*dealloc* pointer to the deallocator to use when deleting the buffer

##### 6.190.2.2 virtual OsclRefCounterDA::~OsclRefCounterDA () [inline, virtual]

Destructor empty

## 6.190.3 Member Function Documentation

### 6.190.3.1 void OsclRefCounterDA::addRef () [inline, virtual]

Add to the reference count

Implements [OsclRefCounter](#).

### 6.190.3.2 uint32 OsclRefCounterDA::getCount () [inline, virtual]

Gets the current number of references

Implements [OsclRefCounter](#).

### 6.190.3.3 void OsclRefCounterDA::removeRef () [inline, virtual]

Remove from the reference count

Implements [OsclRefCounter](#).

The documentation for this class was generated from the following file:

- [oscl\\_refcounter.h](#)

## 6.191 OsclRefCounterMemFrag Class Reference

```
#include <oscl_refcounter_memfrag.h>
```

### Public Methods

- [OsclRefCounterMemFrag \(OsclMemoryFragment &m, OsclRefCounter \\*r, uint32 in\\_capacity\)](#)
- [OsclRefCounterMemFrag \(const OsclRefCounterMemFrag &x\)](#)
- [OsclRefCounterMemFrag \(\)](#)
- [OsclRefCounterMemFrag & operator= \(const OsclRefCounterMemFrag &x\)](#)
- [~OsclRefCounterMemFrag \(\)](#)
- [OsclRefCounter \\* getRefCounter \(\)](#)
- [OsclMemoryFragment & getMemFrag \(\)](#)
- [OsclAny \\* getMemFragPtr \(\)](#)
- [uint32 getMemFragSize \(\)](#)
- [uint32 getCapacity \(\)](#)
- [uint32 getCount \(\)](#)

### 6.191.1 Detailed Description

Class to contain a memory fragment with it's associated reference counter.

### 6.191.2 Constructor & Destructor Documentation

#### 6.191.2.1 OsclRefCounterMemFrag::OsclRefCounterMemFrag ([OsclMemoryFragment & m](#), [OsclRefCounter \\* r](#), [uint32 in\\_capacity](#)) [inline]

Constructor. A valid memory fragment and reference counter are required as input. The memory fragment structure will be copied locally.

##### Parameters:

- m* reference to memory fragment
- r* pointer to the reference counter associated with the memory fragment.

#### 6.191.2.2 OsclRefCounterMemFrag::OsclRefCounterMemFrag ([const OsclRefCounterMemFrag & x](#)) [inline]

Copy constructor.

#### 6.191.2.3 OsclRefCounterMemFrag::OsclRefCounterMemFrag () [inline]

Default constructor.

#### 6.191.2.4 OsclRefCounterMemFrag::~OsclRefCounterMemFrag () [inline]

Destructor. Removes this object's reference from the reference counter. The reference counter will not be deleted. The reference counter is designed to self-delete when it's reference count reaches 0.

### 6.191.3 Member Function Documentation

#### 6.191.3.1 `uint32 OsclRefCounterMemFrag::getCapacity () [inline]`

Returns the capacity of the memory fragment

**Returns:**

#### 6.191.3.2 `uint32 OsclRefCounterMemFrag::getCount () [inline]`

Returns the reference counter's current count.

#### 6.191.3.3 `OsclMemoryFragment& OsclRefCounterMemFrag::getMemFrag () [inline]`

Returns a reference to the contained memory fragment structure.

#### 6.191.3.4 `OsclAny* OsclRefCounterMemFrag::getMemFragPtr () [inline]`

Returns a pointer to the memory fragment data.

#### 6.191.3.5 `uint32 OsclRefCounterMemFrag::getMemFragSize () [inline]`

Returns the size of the memory fragment data which equals its filled size.

**Returns:**

#### 6.191.3.6 `OsclRefCounter* OsclRefCounterMemFrag::getRefCounter () [inline]`

Returns a pointer to the contained reference counter object

#### 6.191.3.7 `OsclRefCounterMemFrag& OsclRefCounterMemFrag::operator= (const OsclRefCounterMemFrag & x) [inline]`

Assignment Operator

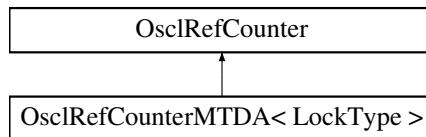
The documentation for this class was generated from the following file:

- [oscl\\_refcounter\\_memfrag.h](#)

## 6.192 OsclRefCounterMTDA< LockType > Class Template Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for OsclRefCounterMTDA< LockType >::



### Public Methods

- [OsclRefCounterMTDA \(OsclAny \\*p, OsclDestructDealloc \\*dealloc\)](#)
- virtual [~OsclRefCounterMTDA \(\)](#)
- void [addRef \(\)](#)
- void [removeRef \(\)](#)
- uint32 [getCount \(\)](#)

### 6.192.1 Detailed Description

**template<class LockType> class OsclRefCounterMTDA< LockType >**

Implementation of [OsclRefCounterDA](#) for multi-threaded use. A templated lock class must be specified.

### 6.192.2 Constructor & Destructor Documentation

**6.192.2.1 template<class LockType> OsclRefCounterMTDA< LockType >::OsclRefCounterMTDA (OsclAny \*p, OsclDestructDealloc \*dealloc) [inline]**

Constructor Takes a pointer to the buffer to track, and a pointer to the deallocator object which will be used to delete the buffer.

When the reference count reaches zero, the buffer will be deleted by the deallocator. Also, the [OsclRefCounter](#) object (this) will self-destruct when the reference count is zero. In some cases the [OsclRefCounter](#) object will be part of the buffer being deleted. For such cases, the object pointer must be equal to the buffer pointer given at construction. If the object is not part of the buffer being deleted, it will self-destruct with a call to [delete\(\)](#).

#### Parameters:

*p* pointer to the buffer to track

*dealloc* pointer to the deallocator to use when deleting the buffer

**6.192.2.2 template<class LockType> virtual OsclRefCounterMTDA< LockType >::~OsclRefCounterMTDA () [inline, virtual]**

Destructor empty

### 6.192.3 Member Function Documentation

**6.192.3.1 template<class LockType> void OsclRefCounterMTDA< LockType >::addRef ()  
[inline, virtual]**

Add to the reference count

Implements [OsclRefCounter](#).

**6.192.3.2 template<class LockType> uint32 OsclRefCounterMTDA< LockType >::getCount ()  
[inline, virtual]**

Gets the current number of references

Implements [OsclRefCounter](#).

**6.192.3.3 template<class LockType> void OsclRefCounterMTDA< LockType >::removeRef ()  
[inline, virtual]**

Remove from the reference count

Implements [OsclRefCounter](#).

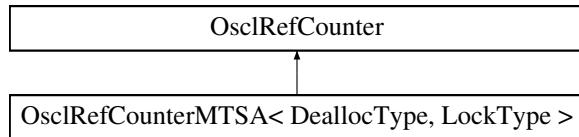
The documentation for this class was generated from the following file:

- [oscl\\_refcounter.h](#)

## 6.193 OsclRefCounterMTSA< DeallocType, LockType > Class Template Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for OsclRefCounterMTSA< DeallocType, LockType >::



### Public Methods

- [OsclRefCounterMTSA \(OsclAny \\*p\)](#)
- virtual [~OsclRefCounterMTSA \(\)](#)
- void [addRef \(\)](#)
- void [removeRef \(\)](#)
- uint32 [getCount \(\)](#)

#### 6.193.1 Detailed Description

```
template<class DeallocType, class LockType> class OsclRefCounterMTSA< DeallocType, LockType >
```

Implementation of [OsclRefCounterSA](#) for multi-threaded use. A templated lock class must be specified.

#### 6.193.2 Constructor & Destructor Documentation

**6.193.2.1 template<class DeallocType, class LockType> OsclRefCounterMTSA< DeallocType, LockType >::OsclRefCounterMTSA (OsclAny \* p) [inline]**

Constructor Takes a pointer to the buffer to track.

When the reference count reaches zero, the buffer will be deleted by the deallocator. Also, the [OsclRefCounter](#) object (this) will self-destruct when the reference count is zero. In some cases the [OsclRefCounter](#) object will be part of the buffer being deleted. For such cases, the object pointer must be equal to the buffer pointer given at construction. If the object is not part of the buffer being deleted, it will self-destruct with a call to `delete()`.

**Parameters:**

*p* pointer to the buffer to track

**6.193.2.2 template<class DeallocType, class LockType> virtual OsclRefCounterMTSA< DeallocType, LockType >::~OsclRefCounterMTSA () [inline, virtual]**

Destructor empty

### 6.193.3 Member Function Documentation

**6.193.3.1 template<class DeallocType, class LockType> void OsclRefCounterMTSA< DeallocType, LockType >::addRef () [inline, virtual]**

Add to the reference count

Implements [OsclRefCounter](#).

**6.193.3.2 template<class DeallocType, class LockType> uint32 OsclRefCounterMTSA< DeallocType, LockType >::getCount () [inline, virtual]**

Gets the current number of references

Implements [OsclRefCounter](#).

**6.193.3.3 template<class DeallocType, class LockType> void OsclRefCounterMTSA< DeallocType, LockType >::removeRef () [inline, virtual]**

Remove from the reference count

Implements [OsclRefCounter](#).

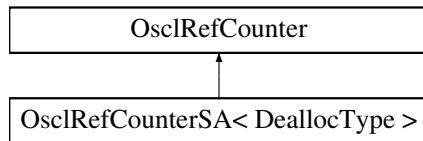
The documentation for this class was generated from the following file:

- [oscl\\_refcounter.h](#)

## 6.194 OsclRefCounterSA< DeallocType > Class Template Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for OsclRefCounterSA< DeallocType >::



### Public Methods

- [OsclRefCounterSA \(OsclAny \\*p\)](#)
- virtual [~OsclRefCounterSA \(\)](#)
- void [addRef \(\)](#)
- void [removeRef \(\)](#)
- uint32 [getCount \(\)](#)

#### 6.194.1 Detailed Description

**template<class DeallocType> class OsclRefCounterSA< DeallocType >**

Implementation of an [OsclRefCounter](#) that uses a statically created deallocator.

#### 6.194.2 Constructor & Destructor Documentation

##### 6.194.2.1 template<class DeallocType> OsclRefCounterSA< DeallocType >::OsclRefCounterSA (OsclAny \*p) [inline]

Constructor Takes a pointer to the buffer to track.

When the reference count reaches zero, the buffer will be deleted by the deallocator. Also, the [OsclRefCounter](#) object (this) will self-destruct when the reference count is zero. In some cases the [OsclRefCounter](#) object will be part of the buffer being deleted. For such cases, the object pointer must be equal to the buffer pointer given at construction. If the object is not part of the buffer being deleted, it will self-destruct with a call to [delete\(\)](#).

**Parameters:**

*p* pointer to the buffer to track

##### 6.194.2.2 template<class DeallocType> virtual OsclRefCounterSA< DeallocType >::~OsclRefCounterSA () [inline, virtual]

Destructor empty

### 6.194.3 Member Function Documentation

**6.194.3.1 template<class DeallocType> void OsclRefCounterSA< DeallocType >::addRef ()**  
[inline, virtual]

Add to the reference count

Implements [OsclRefCounter](#).

**6.194.3.2 template<class DeallocType> uint32 OsclRefCounterSA< DeallocType >::getCount ()**  
[inline, virtual]

Gets the current number of references

Implements [OsclRefCounter](#).

**6.194.3.3 template<class DeallocType> void OsclRefCounterSA< DeallocType >::removeRef ()**  
[inline, virtual]

Remove from the reference count

Implements [OsclRefCounter](#).

The documentation for this class was generated from the following file:

- [oscl\\_refcounter.h](#)

## 6.195 OsclRegistryAccessClient Class Reference

```
#include <oscl_registry_access_client.h>
```

### Public Methods

- OSCL\_IMPORT\_REF OsclRegistryAccessClient ()
- OSCL\_IMPORT\_REF ~OsclRegistryAccessClient ()
- OSCL\_IMPORT\_REF int32 Connect ()
- OSCL\_IMPORT\_REF OsclComponentFactory GetFactory (OSCL\_String &aComponent)
- OSCL\_IMPORT\_REF void GetFactories (OSCL\_String &aRegistry, Oscl\_Vector< OsclRegistryAccessElement, OsclMemAllocator > &aVec)
- OSCL\_IMPORT\_REF void Close ()

#### 6.195.1 Constructor & Destructor Documentation

**6.195.1.1 OSCL\_IMPORT\_REF OsclRegistryAccessClient::OsclRegistryAccessClient ()**

**6.195.1.2 OSCL\_IMPORT\_REF OsclRegistryAccessClient::~OsclRegistryAccessClient ()**

#### 6.195.2 Member Function Documentation

**6.195.2.1 OSCL\_IMPORT\_REF void OsclRegistryAccessClient::Close ()**

Close and cleanup session.

**6.195.2.2 OSCL\_IMPORT\_REF int32 OsclRegistryAccessClient::Connect ()**

Create a session.

**Returns:**

OsclErrNone on success.

**6.195.2.3 OSCL\_IMPORT\_REF void OsclRegistryAccessClient::GetFactories (OSCL\_String & aRegistry, Oscl\_Vector< OsclRegistryAccessElement, OsclMemAllocator > & aVec)**

Get all factories for a given registry type.

**Parameters:**

*aRegistry*: registry MIME type

*aVec*: output component factory + mimestring vector.

**6.195.2.4 OSCL\_IMPORT\_REF OsclComponentFactory OsclRegistryAccessClient::GetFactory (OSCL\_String & aComponent)**

Lookup a factory by registry and component mime type.

**Parameters:**

*aComponent*: registry+component MIME type

**Returns:**

Factory. Factory will be NULL if not found.

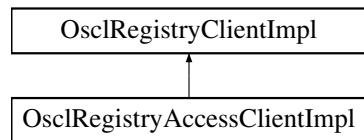
The documentation for this class was generated from the following file:

- [oscl\\_registry\\_access\\_client.h](#)

## 6.196 OsclRegistryAccessClientImpl Class Reference

```
#include <oscl_registry_client_impl.h>
```

Inheritance diagram for OsclRegistryAccessClientImpl::



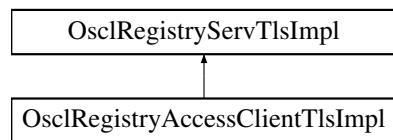
The documentation for this class was generated from the following file:

- [oscl\\_registry\\_client\\_impl.h](#)

## 6.197 OsclRegistryAccessClientTlsImpl Class Reference

```
#include <oscl_registry_client_impl.h>
```

Inheritance diagram for OsclRegistryAccessClientTlsImpl::



The documentation for this class was generated from the following file:

- [oscl\\_registry\\_client\\_impl.h](#)

## 6.198 OsclRegistryAccessElement Class Reference

```
#include <oscl_registry_types.h>
```

### Data Fields

- [OsclComponentFactory](#) iFactory
- [OSCL\\_HeapString< OsclMemAllocator >](#) iMimeType

#### 6.198.1 Detailed Description

A class used to access the registry data

#### 6.198.2 Field Documentation

##### 6.198.2.1 [OsclComponentFactory](#) OsclRegistryAccessElement::iFactory

##### 6.198.2.2 [OSCL\\_HeapString<OsclMemAllocator>](#) OsclRegistryAccessElement::iMimeType

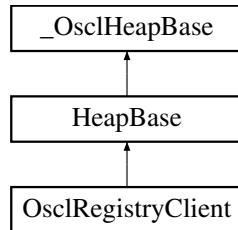
The documentation for this class was generated from the following file:

- [oscl\\_registry\\_types.h](#)

## 6.199 OsclRegistryClient Class Reference

```
#include <oscl_registry_client.h>
```

Inheritance diagram for OsclRegistryClient::



### Public Methods

- OSCL\_IMPORT\_REF OsclRegistryClient ()
- OSCL\_IMPORT\_REF ~OsclRegistryClient ()
- OSCL\_IMPORT\_REF int32 Connect (bool aPerThread=false)
- OSCL\_IMPORT\_REF int32 Register (OSCL\_String &aComponentID, OsclComponentFactory aFactory)
- OSCL\_IMPORT\_REF int32 UnRegister (OSCL\_String &aComponentID)
- OSCL\_IMPORT\_REF void Close ()

#### 6.199.1 Constructor & Destructor Documentation

**6.199.1.1 OSCL\_IMPORT\_REF OsclRegistryClient::OsclRegistryClient ()**

**6.199.1.2 OSCL\_IMPORT\_REF OsclRegistryClient::~OsclRegistryClient ()**

#### 6.199.2 Member Function Documentation

**6.199.2.1 OSCL\_IMPORT\_REF void OsclRegistryClient::Close ()**

Close and cleanup. All components registered in this session are automatically unregistered.

**6.199.2.2 OSCL\_IMPORT\_REF int32 OsclRegistryClient::Connect (bool *aPerThread* = false)**

Create a session.

**Parameters:**

*aPerThread*: Select per-thread registry instead of global registry.

**Returns:**

OsclErrNone on success.

**6.199.2.3 OSCL\_IMPORT\_REF int32 OsclRegistryClient::Register ([OSCL\\_String &](#)  
*aComponentID*, [OsclComponentFactory](#) *aFactory*)**

Register a component factory by registry ID and component ID.

**Parameters:**

*aComponentID*: registry + component mime-string.

*aFactory*: factory function pointer.

*aParam*: component Create param.

**Returns:**

OsclErrNone on success.

**6.199.2.4 OSCL\_IMPORT\_REF int32 OsclRegistryClient::UnRegister ([OSCL\\_String &](#)  
*aComponentID*)**

Unregister a previously registered component.

**Returns:**

OsclErrNone on success.

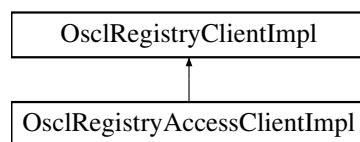
The documentation for this class was generated from the following file:

- [oscl\\_registry\\_client.h](#)

## 6.200 OsclRegistryClientImpl Class Reference

```
#include <oscl_registry_client_impl.h>
```

Inheritance diagram for OsclRegistryClientImpl:



### Protected Methods

- int32 [Connect \(\)](#)
- void [Close \(\)](#)
- int32 [Register \(OSCL\\_String &, OsclComponentFactory\)](#)
- int32 [UnRegister \(OSCL\\_String &\)](#)
- [OsclComponentFactory GetFactory \(OSCL\\_String &\)](#)
- void [GetFactories \(OSCL\\_String &, Oscl\\_Vector< OsclRegistryAccessElement, OsclMemAllocator > &\)](#)

### Friends

- class [OsclRegistryClient](#)
- class [OsclRegistryAccessClient](#)

## 6.200.1 Member Function Documentation

6.200.1.1 **void OsclRegistryClientImpl::Close (void)** [inline, protected]

6.200.1.2 **int32 OsclRegistryClientImpl::Connect ()** [inline, protected]

6.200.1.3 **void OsclRegistryClientImpl::GetFactories (OSCL\_String &, Oscl\_Vector< OsclRegistryAccessElement, OsclMemAllocator > &)** [inline, protected]

6.200.1.4 **OsclComponentFactory OsclRegistryClientImpl::GetFactory (OSCL\_String &)**  
[inline, protected]

6.200.1.5 **int32 OsclRegistryClientImpl::Register (OSCL\_String &, OsclComponentFactory)**  
[inline, protected]

6.200.1.6 **int32 OsclRegistryClientImpl::UnRegister (OSCL\_String &)** [inline,  
protected]

## 6.200.2 Friends And Related Function Documentation

6.200.2.1 **friend class OsclRegistryAccessClient** [friend]

6.200.2.2 **friend class OsclRegistryClient** [friend]

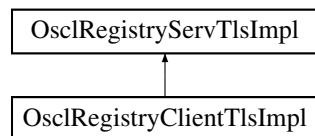
The documentation for this class was generated from the following file:

- [oscl\\_registry\\_client\\_impl.h](#)

## 6.201 OsclRegistryClientTlsImpl Class Reference

```
#include <oscl_registry_client_impl.h>
```

Inheritance diagram for OsclRegistryClientTlsImpl::



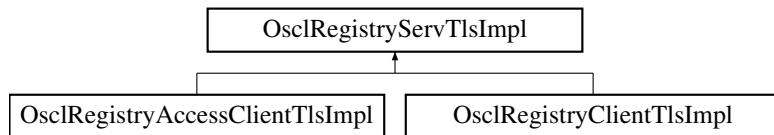
The documentation for this class was generated from the following file:

- [oscl\\_registry\\_client\\_impl.h](#)

## 6.202 OsclRegistryServTlsImpl Class Reference

```
#include <oscl_registry_serv_impl_tls.h>
```

Inheritance diagram for OsclRegistryServTlsImpl::



### Protected Methods

- [OsclRegistryServTlsImpl \(\)](#)
- virtual [~OsclRegistryServTlsImpl \(\)](#)
- int32 [Connect \(\)](#)
- void [Close \(\)](#)
- int32 [Register \(OSCL\\_String &aComponentID, OsclComponentFactory aFactory\)](#)
- int32 [UnRegister \(OSCL\\_String &aComponentID\)](#)
- [OsclComponentFactory GetFactory \(OSCL\\_String &aComponent\)](#)
- void [GetFactories \(OSCL\\_String &aRegistry, Oscl\\_Vector< OsclRegistryAccessElement, OsclMemAllocator > &aVec\)](#)

### Friends

- class [OsclRegistryClient](#)
- class [OsclRegistryAccessClient](#)

### 6.202.1 Constructor & Destructor Documentation

6.202.1.1 `OsclRegistryServTlsImpl::OsclRegistryServTlsImpl ()` [protected]

6.202.1.2 `virtual OsclRegistryServTlsImpl::~OsclRegistryServTlsImpl ()` [protected, virtual]

### 6.202.2 Member Function Documentation

6.202.2.1 `void OsclRegistryServTlsImpl::Close ()` [protected]

6.202.2.2 `int32 OsclRegistryServTlsImpl::Connect ()` [protected]

6.202.2.3 `void OsclRegistryServTlsImpl::GetFactories (OSCL_String & aRegistry, Oscl_Vector< OsclRegistryAccessElement, OsclMemAllocator > & aVec)` [protected]

6.202.2.4 `OsclComponentFactory OsclRegistryServTlsImpl::GetFactory (OSCL_String & aComponent)` [protected]

6.202.2.5 `int32 OsclRegistryServTlsImpl::Register (OSCL_String & aComponentID, OsclComponentFactory aFactory)` [protected]

6.202.2.6 `int32 OsclRegistryServTlsImpl::UnRegister (OSCL_String & aComponentID)` [protected]

### 6.202.3 Friends And Related Function Documentation

6.202.3.1 `friend class OsclRegistryAccessClient` [friend]

6.202.3.2 `friend class OsclRegistryClient` [friend]

The documentation for this class was generated from the following file:

- [oscl\\_registry\\_serv\\_impl\\_tls.h](#)

## 6.203 OsclScheduler Class Reference

```
#include <oscl_scheduler.h>
```

### Static Public Methods

- OSCL\_IMPORT\_REF void [Init](#) (const char \*name, [Oscl\\_DefAlloc](#) \*alloc=NULL, int nreserve=20)
- OSCL\_IMPORT\_REF void [Cleanup](#) ()

#### 6.203.1 Detailed Description

Per-thread scheduler initialization and cleanup.

#### 6.203.2 Member Function Documentation

##### 6.203.2.1 OSCL\_IMPORT\_REF void OsclScheduler::Cleanup () [static]

This routine uninstalls and destroys Oscl scheduler for the calling thread.

##### 6.203.2.2 OSCL\_IMPORT\_REF void OsclScheduler::Init (const char \* *name*, [Oscl\\_DefAlloc](#) \* *alloc* = NULL, int *nreserve* = 20) [static]

This routine creates and installs a scheduler in the calling thread.

#### Parameters:

- name*:** (input param) scheduler name.
- alloc*:** (input param) optional allocator to use for the internal implementation.
- nreserve*:** (input param) optional value for ready queue reserve size.

The documentation for this class was generated from the following file:

- [oscl\\_scheduler.h](#)

## 6.204 OsclSchedulerObserver Class Reference

```
#include <oscl_scheduler.h>
```

### Public Methods

- virtual void **OsclSchedulerTimerCallback** (**OsclAny** \*aContext, uint32 aDelayMsec)=0
- virtual void **OsclSchedulerReadyCallback** (**OsclAny** \*aContext)=0
- virtual ~**OsclSchedulerObserver** ()

### 6.204.1 Detailed Description

OsclSchedulerObserver is an observer class for use when running scheduler in non-blocking mode. The scheduler observer can register for callbacks so it will be notified when it is necessary to run scheduler again. Note: non-blocking mode and scheduler callbacks are not supported on Symbian.

### 6.204.2 Constructor & Destructor Documentation

6.204.2.1 virtual OsclSchedulerObserver::~OsclSchedulerObserver () [inline, virtual]

### 6.204.3 Member Function Documentation

6.204.3.1 virtual void OsclSchedulerObserver::OsclSchedulerReadyCallback (**OsclAny** \* *aContext*) [pure virtual]

OsclSchedulerReadyCallback is called when the ready queue is updated, meaning an AO is ready to run. Scheduler needs to be run ASAP. Calling context may be any thread, so be careful!

The current observer is cleared before making the callback, so the observer must call RegisterForCallback again if it wants further notifications.

6.204.3.2 virtual void OsclSchedulerObserver::OsclSchedulerTimerCallback (**OsclAny** \* *aContext*, uint32 *aDelayMsec*) [pure virtual]

OsclSchedulerTimerCallback is called when the front of the timer queue is updated. This means the minimum delay has changed and scheduler needs to be run again after aDelayMsec. Calling context is in-thread.

The current observer is cleared before making the callback, so the observer must call RegisterForCallback again if it wants further notifications.

The documentation for this class was generated from the following file:

- [oscl\\_scheduler.h](#)

## 6.205 OsclScopedLock< LockClass > Class Template Reference

The OsclScopedLock class is a template class that handles unlocking an abstract class on destruction. This is very useful for ensuring that the lock is released when the OsclScopedLock goes out of scope.

```
#include <oscl_lock_base.h>
```

### Public Methods

- [OsclScopedLock \(LockClass &inLock\)](#)  
*Default constructor Initializes the pointer and takes ownership.*
- [~OsclScopedLock \(\)](#)  
*Destructor.*

### 6.205.1 Detailed Description

**template<class LockClass> class OsclScopedLock< LockClass >**

The OsclScopedLock class is a template class that handles unlocking an abstract class on destruction. This is very useful for ensuring that the lock is released when the OsclScopedLock goes out of scope.

The purpose of this class is to provide a way to prevent accidental resource leaks in a class or a method, due to "not remembering to unlock" variables which might lead to deadlock conditions.

### 6.205.2 Constructor & Destructor Documentation

**6.205.2.1 template<class LockClass> OsclScopedLock< LockClass >::OsclScopedLock (LockClass & inLock) [inline, explicit]**

Default constructor Initializes the pointer and takes ownership.

**6.205.2.2 template<class LockClass> OsclScopedLock< LockClass >::~OsclScopedLock () [inline]**

Destructor.

The pointer is deleted in case this class still has ownership

The documentation for this class was generated from the following file:

- [oscl\\_lock\\_base.h](#)

## 6.206 OsclSelect Class Reference

```
#include <oscl_init.h>
```

### Public Methods

- [OsclSelect \(\)](#)
- [OsclSelect \(Oscl\\_DefAlloc \\*erralloc, Oscl\\_DefAlloc \\*schedalloc, const char \\*name, int32 reserve=10, bool heapcheck=false, FILE \\*output=NULL\)](#)

### Data Fields

- bool [iOsclBase](#)
- bool [iOsclMemory](#)
- bool [iOsclErrorTrap](#)
- bool [iOsclLogger](#)
- bool [iOsclScheduler](#)
- [Oscl\\_DefAlloc \\* iErrAlloc](#)
- [Oscl\\_DefAlloc \\* iSchedulerAlloc](#)
- const char \* [iSchedulerName](#)
- int32 [iSchedulerReserve](#)
- bool [iHeapCheck](#)
- FILE \* [iOutputFile](#)

### 6.206.1 Detailed Description

Oscl Module selection and Init/Cleanup options.

## 6.206.2 Constructor & Destructor Documentation

**6.206.2.1 OsclSelect::OsclSelect () [inline]**

**6.206.2.2 OsclSelect::OsclSelect ([Oscl\\_DefAlloc](#) \* *erralloc*, [Oscl\\_DefAlloc](#) \* *schedalloc*, const char \* *name*, int32 *reserve* = 10, bool *heapcheck* = false, FILE \* *output* = NULL) [inline]**

## 6.206.3 Field Documentation

**6.206.3.1 [Oscl\\_DefAlloc](#)\* OsclSelect::iErrAlloc**

**6.206.3.2 bool OsclSelect::iHeapCheck**

**6.206.3.3 bool OsclSelect::iOsclBase**

**6.206.3.4 bool OsclSelect::iOsclErrorTrap**

**6.206.3.5 bool OsclSelect::iOsclLogger**

**6.206.3.6 bool OsclSelect::iOsclMemory**

**6.206.3.7 bool OsclSelect::iOsclScheduler**

**6.206.3.8 FILE\* OsclSelect::iOutputFile**

**6.206.3.9 [Oscl\\_DefAlloc](#)\* OsclSelect::iSchedulerAlloc**

**6.206.3.10 const char\* OsclSelect::iSchedulerName**

**6.206.3.11 int32 OsclSelect::iSchedulerReserve**

The documentation for this class was generated from the following file:

- [oscl\\_init.h](#)

## 6.207 OsclSemaphore Class Reference

```
#include <oscl_semaphore.h>
```

### Public Methods

- OSCL\_IMPORT\_REF OsclSemaphore ()
- OSCL\_IMPORT\_REF ~OsclSemaphore ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Create (uint32 initVal=0)
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Close ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Wait ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Wait (uint32 timeout\_msec)
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError TryWait ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Signal ()

### 6.207.1 Detailed Description

Class Semaphore

### 6.207.2 Constructor & Destructor Documentation

#### 6.207.2.1 OSCL\_IMPORT\_REF OsclSemaphore::OsclSemaphore ()

Class constructor

#### 6.207.2.2 OSCL\_IMPORT\_REF OsclSemaphore::~OsclSemaphore ()

Class destructor

### 6.207.3 Member Function Documentation

#### 6.207.3.1 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclSemaphore::Close ()

Closes the Semaphore

**Parameters:**

*It* wont take any parameters

**Returns:**

Returns the Error whether it is success or failure incase of failure it will return what is the specific error

#### 6.207.3.2 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclSemaphore::Create (uint32 initVal = 0)

Creates the Semaphore

**Parameters:**

*Intialcount*

**Returns:**

Returns the Error whether it is success or failure incase of failure it will return what is the specific error

**6.207.3.3 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclSemaphore::Signal ()**

Signals that the thread is finished with the Semaphore

**Parameters:**

*It* wont take any parameters

**Returns:**

Returns the Error whether it is success or failure incase of failure it will return what is the specific error

**6.207.3.4 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclSemaphore::TryWait ()**

Try to acquire semaphore ,if the semaphore is already acquired by another thread, calling thread immediately returns with out blocking

**Parameters:**

*It* wont take any parameters

**Returns:**

Returns SUCCESS\_ERROR if the semaphore was acquired, SEM\_LOCKED\_ERROR if the semaphore cannot be acquired without waiting, or an error code if the operation failed. Note: this function may not be supported on all platforms, and may return NOT\_IMPLEMENTED.

**6.207.3.5 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclSemaphore::Wait (uint32 timeout\_msec)**

Makes the thread to wait on the Semaphore, with a timeout.

**Parameters:**

*timeout* in milliseconds.

**Returns:**

Returns SUCCESS\_ERROR if the semaphore was aquired, WAIT\_TIMEOUT\_ERROR if the timeout expired without acquiring the semaphore, or an error code if the operation failed. Note: this function may not be supported on all platforms, and may return NOT\_IMPLEMENTED.

**6.207.3.6 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclSemaphore::Wait ()**

Makes the thread to wait on the Semaphore

**Parameters:**

*It* wont take any parameters

**Returns:**

Returns the Error whether it is success or failure incase of failure it will return what is the specific error

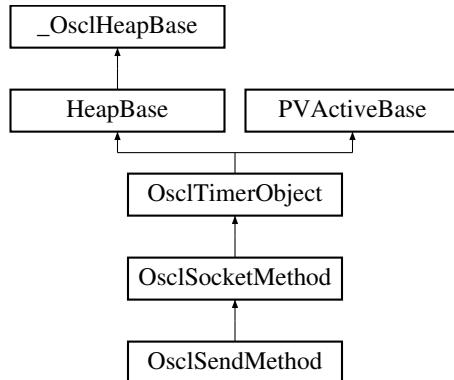
The documentation for this class was generated from the following file:

- [oscl\\_semaphore.h](#)

## 6.208 OsclSendMethod Class Reference

```
#include <oscl_socket_send.h>
```

Inheritance diagram for OsclSendMethod::



### Public Methods

- [~OsclSendMethod \(\)](#)
- [TPVSocketEvent Send \(const uint8 \\*aPtr, uint32 aLen, int32 aTimeout\)](#)
- [uint8 \\* GetSendData \(int32 \\*aLength\)](#)
- [OsclSendRequest \\* SendRequest \(\)](#)

### Static Public Methods

- [OsclSendMethod \\* NewL \(OsclIPSocketI &c\)](#)

#### 6.208.1 Constructor & Destructor Documentation

##### 6.208.1.1 OsclSendMethod::~OsclSendMethod ()

#### 6.208.2 Member Function Documentation

##### 6.208.2.1 uint8\* OsclSendMethod::GetSendData (int32 \* aLength)

##### 6.208.2.2 OsclSendMethod\* OsclSendMethod::NewL (OsclIPSocketI & c) [static]

##### 6.208.2.3 TPVSocketEvent OsclSendMethod::Send (const uint8 \*& aPtr, uint32 aLen, int32 aTimeout)

##### 6.208.2.4 OsclSendRequest\* OsclSendMethod::SendRequest () [inline]

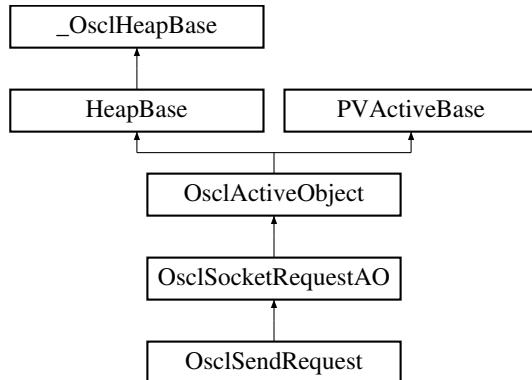
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_send.h](#)

## 6.209 OsclSendRequest Class Reference

```
#include <oscl_socket_send.h>
```

Inheritance diagram for OsclSendRequest::



### Public Methods

- [OsclSendRequest \(OsclSocketMethod &c\)](#)
- void [Send \(const uint8 \\*&aPtr, uint32 aLen\)](#)
- void [Success \(\)](#)
- uint8 \* [GetSendData \(int32 \\*aLength\)](#)

#### 6.209.1 Constructor & Destructor Documentation

**6.209.1.1 OsclSendRequest::OsclSendRequest (OsclSocketMethod & c) [inline]**

#### 6.209.2 Member Function Documentation

**6.209.2.1 uint8\* OsclSendRequest::GetSendData (int32 \* aLength)**

**6.209.2.2 void OsclSendRequest::Send (const uint8 \*& aPtr, uint32 aLen)**

**6.209.2.3 void OsclSendRequest::Success () [virtual]**

Reimplemented from [OsclSocketRequestAO](#).

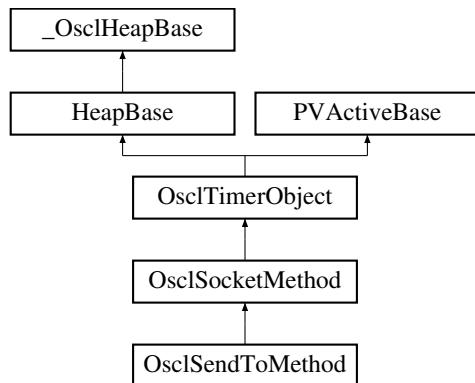
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_send.h](#)

## 6.210 OsclSendToMethod Class Reference

```
#include <oscl_socket_send_to.h>
```

Inheritance diagram for OsclSendToMethod:::



### Public Methods

- [~OsclSendToMethod \(\)](#)
- [TPVSocketEvent SendTo \(const uint8 \\*&aPtr, uint32 aLen, OsclNetworkAddress &aAddress, int32 aTimeout\)](#)
- [uint8 \\* GetSendData \(int32 \\*aLength\)](#)
- [OsclSendToRequest \\* SendToRequest \(\)](#)

### Static Public Methods

- [OsclSendToMethod \\* NewL \(OsclIPSocketI &c\)](#)

#### 6.210.1 Constructor & Destructor Documentation

##### 6.210.1.1 OsclSendToMethod::~OsclSendToMethod ()

#### 6.210.2 Member Function Documentation

##### 6.210.2.1 uint8\* OsclSendToMethod::GetSendData (int32 \* aLength)

##### 6.210.2.2 OsclSendToMethod\* OsclSendToMethod::NewL (OsclIPSocketI & c) [static]

##### 6.210.2.3 TPVSocketEvent OsclSendToMethod::SendTo (const uint8 \*& aPtr, uint32 aLen, OsclNetworkAddress & aAddress, int32 aTimeout)

##### 6.210.2.4 OsclSendToRequest\* OsclSendToMethod::SendToRequest () [inline]

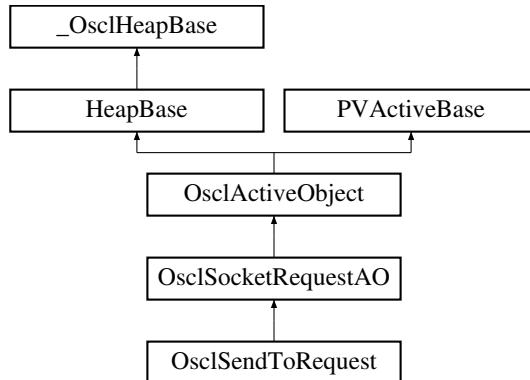
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_send\\_to.h](#)

## 6.211 OsclSendToRequest Class Reference

```
#include <oscl_socket_send_to.h>
```

Inheritance diagram for OsclSendToRequest::



### Public Methods

- [OsclSendToRequest \(OsclSocketMethod &c\)](#)
- void [SendTo \(const uint8 \\*&aPtr, uint32 aLen, OsclNetworkAddress &aAddress\)](#)
- void [Success \(\)](#)
- uint8 \* [GetSendData \(int32 \\*aLength\)](#)

#### 6.211.1 Detailed Description

This is the AO that interacts with the socket server

#### 6.211.2 Constructor & Destructor Documentation

**6.211.2.1 OsclSendToRequest::OsclSendToRequest (OsclSocketMethod & c) [inline]**

#### 6.211.3 Member Function Documentation

**6.211.3.1 uint8\* OsclSendToRequest::GetSendData (int32 \* aLength)**

**6.211.3.2 void OsclSendToRequest::SendTo (const uint8 \*& aPtr, uint32 aLen, OsclNetworkAddress & aAddress)**

**6.211.3.3 void OsclSendToRequest::Success () [virtual]**

Reimplemented from [OsclSocketRequestAO](#).

The documentation for this class was generated from the following file:

- [oscl\\_socket\\_send\\_to.h](#)

## 6.212 OsclSharedPtr< TheClass > Class Template Reference

A parameterized smart pointer class.

```
#include <oscl_shared_ptr.h>
```

### Public Methods

- **OsclSharedPtr ()**  
*Constructor.*
- **OsclSharedPtr (TheClass \*inClassPtr, OsclRefCounter \*in\_refcnt)**  
*Constructor.*
- **OsclSharedPtr (const OsclSharedPtr &inSharedPtr)**  
*Copy constructor.*
- **virtual ~OsclSharedPtr ()**  
*Destructor.*
- **TheClass \* operator → ()**  
**TheClass & operator \* ()**  
*The indirection operator returns a reference to an object of the parameterized type.*
- **operator TheClass \* ()**  
*Casting operator.*
- **TheClass \* GetRep ()**  
*Use this function to get a pointer to the wrapped object.*
- **OsclRefCounter \* GetRefCounter ()**  
*Get the refcount pointer. This should primarily be used for conversion operations.*
- **int get\_count ()**  
*Get a count of how many references to the object exist.*
- **void Bind (const OsclSharedPtr &inHandle)**  
*Use this function to bind an existing OsclSharedPtr to a already-wrapped object.*
- **void Bind (TheClass \*ptr, OsclRefCounter \*in\_refcnt)**  
*Use this function to bind an existing OsclSharedPtr to a new (unwrapped) object.*
- **void Unbind ()**  
*Use this function of unbind an existing OsclSharedPtr.*
- **OsclSharedPtr & operator= (const OsclSharedPtr &inSharedPtr)**  
*Assignment operator.*
- **bool operator== (const OsclSharedPtr &b) const**  
*Test for equality to see if two PVHandles wrap the same object.*

### 6.212.1 Detailed Description

**template<class TheClass> class OsclSharedPtr< TheClass >**

A parameterized smart pointer class.

### 6.212.2 Constructor & Destructor Documentation

**6.212.2.1 template<class TheClass> OsclSharedPtr< TheClass >::OsclSharedPtr () [inline]**

Constructor.

**6.212.2.2 template<class TheClass> OsclSharedPtr< TheClass >::OsclSharedPtr (TheClass \* *inClassPtr*, OsclRefCounter \* *in\_refcnt*) [inline]**

Constructor.

**Parameters:**

*inClassPtr* A pointer to an instance of the parameterized type that the new OsclSharedPtr will wrap.

**6.212.2.3 template<class TheClass> OsclSharedPtr< TheClass >::OsclSharedPtr (const OsclSharedPtr< TheClass > & *inSharedPtr*) [inline]**

Copy constructor.

**6.212.2.4 template<class TheClass> virtual OsclSharedPtr< TheClass >::~OsclSharedPtr () [inline, virtual]**

Destructor.

### 6.212.3 Member Function Documentation

**6.212.3.1 template<class TheClass> int OsclSharedPtr< TheClass >::get\_count () [inline]**

Get a count of how many references to the object exist.

**6.212.3.2 template<class TheClass> OsclRefCounter\* OsclSharedPtr< TheClass >::GetRefCounter () [inline]**

Get the refcount pointer. This should primarily be used for conversion operations.

**6.212.3.3 template<class TheClass> TheClass\* OsclSharedPtr< TheClass >::GetRep () [inline]**

Use this function to get a pointer to the wrapped object.

**6.212.3.4 template<class TheClass> TheClass& OsclSharedPtr< TheClass >::operator \* () [inline]**

The indirection operator returns a reference to an object of the parameterized type.

**6.212.3.5 template<class TheClass> OsclSharedPtr< TheClass >::operator TheClass \* () [inline]**

Casting operator.

**6.212.3.6 template<class TheClass> TheClass\* OsclSharedPtr< TheClass >::operator -> () [inline]**

The dereferencing operator returns a pointer to the parameterized type and can be used to access member elements of TheClass.

**6.212.3.7 template<class TheClass> OsclSharedPtr& OsclSharedPtr< TheClass >::operator=(const OsclSharedPtr< TheClass > & *inSharedPtr*) [inline]**

Assignment operator.

**6.212.3.8 template<class TheClass> void OsclSharedPtr< TheClass >::Unbind () [inline]**

Use this function of unbind an existing OsclSharedPtr.

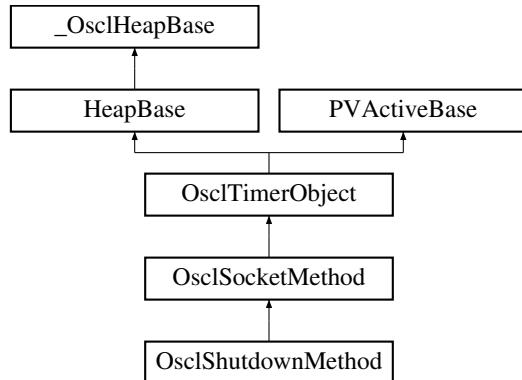
The documentation for this class was generated from the following file:

- [oscl\\_shared\\_ptr.h](#)

## 6.213 OsclShutdownMethod Class Reference

```
#include <oscl_socket_shutdown.h>
```

Inheritance diagram for OsclShutdownMethod::



### Public Methods

- [~OsclShutdownMethod \(\)](#)
- [TPVSocketEvent Shutdown \(TPVSocketShutdown aHow, int32 aTimeout\)](#)
- [OsclShutdownRequest \\* ShutdownRequest \(\)](#)

### Static Public Methods

- [OsclShutdownMethod \\* NewL \(OsclIPSocketI &c\)](#)

#### 6.213.1 Constructor & Destructor Documentation

##### 6.213.1.1 OsclShutdownMethod::~OsclShutdownMethod ()

#### 6.213.2 Member Function Documentation

##### 6.213.2.1 OsclShutdownMethod\* OsclShutdownMethod::NewL (OsclIPSocketI &c) [static]

##### 6.213.2.2 TPVSocketEvent OsclShutdownMethod::Shutdown (TPVSocketShutdown aHow, int32 aTimeout)

##### 6.213.2.3 OsclShutdownRequest\* OsclShutdownMethod::ShutdownRequest () [inline]

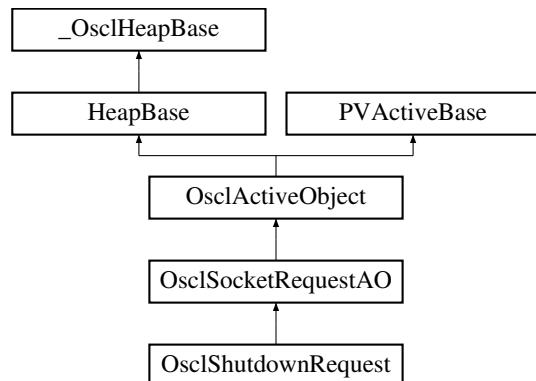
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_shutdown.h](#)

## 6.214 OsclShutdownRequest Class Reference

```
#include <oscl_socket_shutdown.h>
```

Inheritance diagram for OsclShutdownRequest::



### Public Methods

- [OsclShutdownRequest \(OsclSocketMethod &c\)](#)
- [void Shutdown \(TPVSocketShutdown aHow\)](#)

#### 6.214.1 Detailed Description

This is the AO that interacts with the socket server

#### 6.214.2 Constructor & Destructor Documentation

6.214.2.1 [OsclShutdownRequest::OsclShutdownRequest \(OsclSocketMethod & c\) \[inline\]](#)

#### 6.214.3 Member Function Documentation

6.214.3.1 [void OsclShutdownRequest::Shutdown \(TPVSocketShutdown aHow\)](#)

The documentation for this class was generated from the following file:

- [oscl\\_socket\\_shutdown.h](#)

## 6.215 OsclSingleton< T, ID, Registry > Class Template Reference

```
#include <oscl_singleton.h>
```

### Public Methods

- `OsclSingleton ()`
- `~OsclSingleton ()`
- `T & operator * () const`

*The indirection operator (\*) accesses a value indirectly, through a pointer.*

- `T * operator -> () const`

*The indirection operator (->) accesses a value indirectly, through a pointer.*

- `bool set ()`

*set() method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.*

### Protected Attributes

- `T * _Ptr`

```
template<class T, uint32 ID, class Registry = OsclSingletonRegistry> class OsclSingleton< T, ID, Registry >
```

#### 6.215.1 Constructor & Destructor Documentation

**6.215.1.1 template<class T, uint32 ID, class Registry = OsclSingletonRegistry> OsclSingleton< T, ID, Registry >::OsclSingleton () [inline]**

**6.215.1.2 template<class T, uint32 ID, class Registry = OsclSingletonRegistry> OsclSingleton< T, ID, Registry >::~OsclSingleton () [inline]**

#### 6.215.2 Member Function Documentation

**6.215.2.1 template<class T, uint32 ID, class Registry = OsclSingletonRegistry> T& OsclSingleton< T, ID, Registry >::operator \* () const [inline]**

The indirection operator (\*) accesses a value indirectly, through a pointer.

This operator ensures that the OsclSingleton can be used like the regular pointer that it was initialized with.

**6.215.2.2 template<class T, uint32 ID, class Registry = OsclSingletonRegistry> T\* OsclSingleton< T, ID, Registry >::operator -> () const [inline]**

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the OsclSingleton can be used like the regular pointer that it was initialized with.

**6.215.2.3 template<class T, uint32 ID, class Registry = OsclSingletonRegistry> bool OsclSingleton< T, ID, Registry >::set () [inline]**

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

### 6.215.3 Field Documentation

**6.215.3.1 template<class T, uint32 ID, class Registry = OsclSingletonRegistry> T\* OsclSingleton< T, ID, Registry >::\_Ptr [protected]**

The documentation for this class was generated from the following file:

- [oscl\\_singleton.h](#)

## 6.216 OsclSingletonRegistry Class Reference

```
#include <oscl_singleton.h>
```

### Static Public Methods

- OSCL\_IMPORT\_REF [OsclAny](#) \* getInstance (uint32 ID, int32 &error)
- OSCL\_IMPORT\_REF void registerInstance ([OsclAny](#) \*ptr, uint32 ID, int32 &error)
- OSCL\_IMPORT\_REF [OsclAny](#) \* lockAndGetInstance (uint32 ID, int32 &error)
- OSCL\_IMPORT\_REF void registerInstanceAndUnlock ([OsclAny](#) \*ptr, uint32 ID, int32 &error)

### Friends

- class [OsclBase](#)

### 6.216.1 Member Function Documentation

**6.216.1.1 OSCL\_IMPORT\_REF [OsclAny](#)\* OsclSingletonRegistry::getInstance (uint32 *ID*, int32 & *error*) [static]**

**6.216.1.2 OSCL\_IMPORT\_REF [OsclAny](#)\* OsclSingletonRegistry::lockAndGetInstance (uint32 *ID*, int32 & *error*) [static]**

**6.216.1.3 OSCL\_IMPORT\_REF void OsclSingletonRegistry::registerInstance ([OsclAny](#) \* *ptr*, uint32 *ID*, int32 & *error*) [static]**

**6.216.1.4 OSCL\_IMPORT\_REF void OsclSingletonRegistry::registerInstanceAndUnlock ([OsclAny](#) \* *ptr*, uint32 *ID*, int32 & *error*) [static]**

### 6.216.2 Friends And Related Function Documentation

**6.216.2.1 friend class OsclBase [friend]**

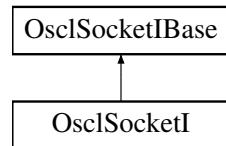
The documentation for this class was generated from the following file:

- [oscl\\_singleton.h](#)

## 6.217 OsclSocketI Class Reference

```
#include <oscl_socket_imp_pv.h>
```

Inheritance diagram for OsclSocketI::



### Public Methods

- `~OsclSocketI ()`
- `int32 Open (OsclSocketServI &aServer, uint32 addrFamily, uint32 sockType, uint32 protocol)`
- `int32 Open (OsclSocketServI &aServer)`
- `int32 Bind (OsclNetworkAddress &anAddr)`
- `int32 Join (OsclNetworkAddress &anAddr)`
- `int32 Close ()`
- `int32 Listen (uint32 qSize)`
- `int32 SetRecvBufferSize (uint32 size)`
- `void Connect (ConnectParam &, OsclSocketRequestAO &)`
- `void Accept (AcceptParam &, OsclSocketRequestAO &)`
- `void Shutdown (ShutdownParam &, OsclSocketRequestAO &)`
- `void Send (SendParam &, OsclSocketRequestAO &)`
- `void SendSuccess (SendParam &)`
- `void SendTo (SendToParam &, OsclSocketRequestAO &)`
- `void SendToSuccess (SendToParam &)`
- `void Recv (RecvParam &, OsclSocketRequestAO &)`
- `void RecvSuccess (RecvParam &)`
- `void RecvFrom (RecvFromParam &, OsclSocketRequestAO &)`
- `void RecvFromSuccess (RecvFromParam &)`
- `TOsclSocket Socket ()`
- `void ProcessConnect (OsclSocketServRequestQElem *)`
- `void ProcessShutdown (OsclSocketServRequestQElem *)`
- `void ProcessAccept (OsclSocketServRequestQElem *)`
- `void ProcessSendTo (OsclSocketServRequestQElem *)`
- `void ProcessRecvFrom (OsclSocketServRequestQElem *)`
- `void ProcessSend (OsclSocketServRequestQElem *)`
- `void ProcessRecv (OsclSocketServRequestQElem *)`
- `PVLogger * Logger ()`

### Static Public Methods

- `OsclSocketI * NewL (Oscl_DefAlloc &a)`
- `bool MakeAddr (OsclNetworkAddress &in, TOsclSockAddr &addr)`
- `void MakeAddr (TOsclSockAddr &in, OsclNetworkAddress &addr)`

## Friends

- class [OsclAcceptRequest](#)
- class [OsclConnectRequest](#)
- class [OsclRecvRequest](#)
- class [OsclRecvFromRequest](#)
- class [OsclSendRequest](#)
- class [OsclSendToRequest](#)
- class [OsclShutdownRequest](#)
- class [OsclUDPSocket](#)
- class [OsclTCPSocket](#)

### 6.217.1 Detailed Description

Socket implementation class

### 6.217.2 Constructor & Destructor Documentation

#### 6.217.2.1 `OsclSocketI::~OsclSocketI ()`

### 6.217.3 Member Function Documentation

#### 6.217.3.1 `void OsclSocketI::Accept (AcceptParam &, OsclSocketRequestAO &) [virtual]`

Implements [OsclSocketIBase](#).

#### 6.217.3.2 `int32 OsclSocketI::Bind (OsclNetworkAddress & anAddr) [virtual]`

Implements [OsclSocketIBase](#).

#### 6.217.3.3 `int32 OsclSocketI::Close () [virtual]`

Implements [OsclSocketIBase](#).

#### 6.217.3.4 `void OsclSocketI::Connect (ConnectParam &, OsclSocketRequestAO &) [virtual]`

Implements [OsclSocketIBase](#).

#### 6.217.3.5 `int32 OsclSocketI::Join (OsclNetworkAddress & anAddr) [virtual]`

Implements [OsclSocketIBase](#).

#### 6.217.3.6 `int32 OsclSocketI::Listen (uint32 qSize) [virtual]`

Implements [OsclSocketIBase](#).

6.217.3.7 **PVLogger\*** OsclSocketI::Logger () [inline]

6.217.3.8 **void** OsclSocketI::MakeAddr (**TOsclSockAddr** & *in*, **OsclNetworkAddress** & *addr*)  
[static]

6.217.3.9 **bool** OsclSocketI::MakeAddr (**OsclNetworkAddress** & *in*, **TOsclSockAddr** & *addr*)  
[static]

6.217.3.10 **OsclSocketI\*** OsclSocketI::NewL (**Oscl\_DefAlloc** & *a*) [static]

6.217.3.11 **int32** OsclSocketI::Open (**OsclSocketServI** & *aServer*) [virtual]

Implements [OsclSocketIBase](#).

6.217.3.12 **int32** OsclSocketI::Open (**OsclSocketServI** & *aServer*, **uint32** *addrFamily*, **uint32**  
*sockType*, **uint32** *protocol*) [virtual]

Implements [OsclSocketIBase](#).

6.217.3.13 **void** OsclSocketI::ProcessAccept (**OsclSocketServRequestQElem** \*)

6.217.3.14 **void** OsclSocketI::ProcessConnect (**OsclSocketServRequestQElem** \*)

6.217.3.15 **void** OsclSocketI::ProcessRecv (**OsclSocketServRequestQElem** \*)

6.217.3.16 **void** OsclSocketI::ProcessRecvFrom (**OsclSocketServRequestQElem** \*)

6.217.3.17 **void** OsclSocketI::ProcessSend (**OsclSocketServRequestQElem** \*)

6.217.3.18 **void** OsclSocketI::ProcessSendTo (**OsclSocketServRequestQElem** \*)

6.217.3.19 **void** OsclSocketI::ProcessShutdown (**OsclSocketServRequestQElem** \*)

6.217.3.20 **void** OsclSocketI::Recv (**RecvParam** &, **OsclSocketRequestAO** &) [virtual]

Implements [OsclSocketIBase](#).

6.217.3.21 **void** OsclSocketI::RecvFrom (**RecvFromParam** &, **OsclSocketRequestAO** &)  
[virtual]

Implements [OsclSocketIBase](#).

6.217.3.22 **void** OsclSocketI::RecvFromSuccess (**RecvFromParam** &) [virtual]

Implements [OsclSocketIBase](#).

6.217.3.23 **void** OsclSocketI::RecvSuccess (**RecvParam** &) [virtual]

Implements [OsclSocketIBase](#).

**6.217.3.24 void OsclSocketI::Send ([SendParam](#) &, [OsclSocketRequestAO](#) &) [virtual]**

Implements [OsclSocketIBase](#).

**6.217.3.25 void OsclSocketI::SendSuccess ([SendParam](#) &) [virtual]**

Implements [OsclSocketIBase](#).

**6.217.3.26 void OsclSocketI::SendTo ([SendToParam](#) &, [OsclSocketRequestAO](#) &) [virtual]**

Implements [OsclSocketIBase](#).

**6.217.3.27 void OsclSocketI::SendToSuccess ([SendToParam](#) &) [virtual]**

Implements [OsclSocketIBase](#).

**6.217.3.28 int32 OsclSocketI::SetRecvBufferSize (uint32 *size*)**

**6.217.3.29 void OsclSocketI::Shutdown ([ShutdownParam](#) &, [OsclSocketRequestAO](#) &) [virtual]**

Implements [OsclSocketIBase](#).

**6.217.3.30 [TOsclSocket](#) OsclSocketI::Socket () [inline]**

## 6.217.4 Friends And Related Function Documentation

**6.217.4.1 friend class OsclAcceptRequest [friend]**

**6.217.4.2 friend class OsclConnectRequest [friend]**

**6.217.4.3 friend class OsclRecvFromRequest [friend]**

**6.217.4.4 friend class OsclRecvRequest [friend]**

**6.217.4.5 friend class OsclSendRequest [friend]**

**6.217.4.6 friend class OsclSendToRequest [friend]**

**6.217.4.7 friend class OsclShutdownRequest [friend]**

**6.217.4.8 friend class OsclTCPSocket [friend]**

Reimplemented from [OsclSocketIBase](#).

**6.217.4.9 friend class OsclUDPSocket [friend]**

Reimplemented from [OsclSocketIBase](#).

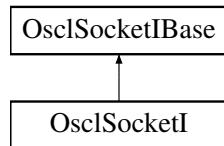
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_imp\\_pv.h](#)

## 6.218 OsclSocketIBase Class Reference

```
#include <oscl_socket_imp_base.h>
```

Inheritance diagram for OsclSocketIBase::



### Public Methods

- virtual ~OsclSocketIBase ()
- virtual int32 Open (OsclSocketServI &aServer, uint32 addrFamily, uint32 sockType, uint32 proto-col)=0
- virtual int32 Open (OsclSocketServI &aServer)=0
- virtual int32 Bind (OsclNetworkAddress &anAddr)=0
- virtual int32 Join (OsclNetworkAddress &anAddr)=0
- virtual int32 Close ()=0
- virtual int32 Listen (uint32 qSize)=0
- virtual void Connect (ConnectParam &, OsclSocketRequestAO &)=0
- virtual void Accept (AcceptParam &, OsclSocketRequestAO &)=0
- virtual void Shutdown (ShutdownParam &, OsclSocketRequestAO &)=0
- virtual void Send (SendParam &, OsclSocketRequestAO &)=0
- virtual void SendSuccess (SendParam &)=0
- virtual void SendTo (SendToParam &, OsclSocketRequestAO &)=0
- virtual void SendToSuccess (SendToParam &)=0
- virtual void Recv (RecvParam &, OsclSocketRequestAO &)=0
- virtual void RecvSuccess (RecvParam &)=0
- virtual void RecvFrom (RecvFromParam &, OsclSocketRequestAO &)=0
- virtual void RecvFromSuccess (RecvFromParam &)=0
- virtual void BindAsync (BindParam &, OsclSocketRequestAO &)
- virtual void ListenAsync (ListenParam &, OsclSocketRequestAO &)
- void CancelFxn (TPVSocketFxn)

### Static Public Methods

- bool HasAsyncBind ()
- bool HasAsyncListen ()

### Protected Methods

- OsclSocketIBase (Oscl\_DefAlloc &a)
- virtual void CancelConnect ()=0
- virtual void CancelAccept ()=0
- virtual void CancelShutdown ()=0
- virtual void CancelSend ()=0

- virtual void [CancelSendTo \(\)=0](#)
- virtual void [CancelRecv \(\)=0](#)
- virtual void [CancelRecvFrom \(\)=0](#)
- virtual void [CancelBind \(\)](#)
- virtual void [CancelListen \(\)](#)
- virtual bool [IsOpen \(\)=0](#)

## Static Protected Methods

- int [GetShutdown \(TPVSocketShutdown aOsclVal\)](#)

## Protected Attributes

- [Oscl\\_DefAlloc & iAlloc](#)
- [OsclSocketServI \\* iSocketServ](#)

## Friends

- class [OsclSocketRequest](#)
- class [OsclSocketMethod](#)
- class [OsclSocketRequestAO](#)
- class [OsclUDPSocket](#)
- class [OsclTCPSocket](#)

## 6.218.1 Detailed Description

Socket implementation base class

## 6.218.2 Constructor & Destructor Documentation

**6.218.2.1** virtual OsclSocketIBase::~OsclSocketIBase () [virtual]

**6.218.2.2** OsclSocketIBase::OsclSocketIBase ([Oscl\\_DefAlloc & a](#)) [protected]

## 6.218.3 Member Function Documentation

**6.218.3.1** virtual void OsclSocketIBase::Accept ([AcceptParam &](#), [OsclSocketRequestAO &](#)) [pure virtual]

Implemented in [OsclSocketI](#).

**6.218.3.2** virtual int32 OsclSocketIBase::Bind ([OsclNetworkAddress & anAddr](#)) [pure virtual]

Implemented in [OsclSocketI](#).

- 6.218.3.3 **virtual void OsclSocketIBase::BindAsync (BindParam &, OsclSocketRequestAO &)**  
[inline, virtual]
- 6.218.3.4 **virtual void OsclSocketIBase::CancelAccept ()** [protected, pure virtual]
- 6.218.3.5 **virtual void OsclSocketIBase::CancelBind ()** [inline, protected, virtual]
- 6.218.3.6 **virtual void OsclSocketIBase::CancelConnect ()** [protected, pure virtual]
- 6.218.3.7 **void OsclSocketIBase::CancelFxn (TPVSocketFxn)**
- 6.218.3.8 **virtual void OsclSocketIBase::CancelListen ()** [inline, protected, virtual]
- 6.218.3.9 **virtual void OsclSocketIBase::CancelRecv ()** [protected, pure virtual]
- 6.218.3.10 **virtual void OsclSocketIBase::CancelRecvFrom ()** [protected, pure virtual]
- 6.218.3.11 **virtual void OsclSocketIBase::CancelSend ()** [protected, pure virtual]
- 6.218.3.12 **virtual void OsclSocketIBase::CancelSendTo ()** [protected, pure virtual]
- 6.218.3.13 **virtual void OsclSocketIBase::CancelShutdown ()** [protected, pure virtual]
- 6.218.3.14 **virtual int32 OsclSocketIBase::Close ()** [pure virtual]

Implemented in [OsclSocketI](#).

- 6.218.3.15 **virtual void OsclSocketIBase::Connect (ConnectParam &, OsclSocketRequestAO &)**  
[pure virtual]

Implemented in [OsclSocketI](#).

- 6.218.3.16 **int OsclSocketIBase::GetShutdown (TPVSocketShutdown aOsclVal)** [static, protected]
- 6.218.3.17 **bool OsclSocketIBase::HasAsyncBind ()** [static]
- 6.218.3.18 **bool OsclSocketIBase::HasAsyncListen ()** [static]
- 6.218.3.19 **virtual bool OsclSocketIBase::IsOpen ()** [protected, pure virtual]
- 6.218.3.20 **virtual int32 OsclSocketIBase::Join (OsclNetworkAddress & anAddr)** [pure virtual]

Implemented in [OsclSocketI](#).

- 6.218.3.21 **virtual int32 OsclSocketIBase::Listen (uint32 qSize)** [pure virtual]

Implemented in [OsclSocketI](#).

**6.218.3.22** `virtual void OsclSocketIBase::ListenAsync (ListenParam &, OsclSocketRequestAO &)`  
[inline, virtual]

**6.218.3.23** `virtual int32 OsclSocketIBase::Open (OsclSocketServI & aServer)` [pure  
virtual]

Implemented in [OsclSocketI](#).

**6.218.3.24** `virtual int32 OsclSocketIBase::Open (OsclSocketServI & aServer, uint32 addrFamily,  
uint32 sockType, uint32 protocol)` [pure virtual]

Implemented in [OsclSocketI](#).

**6.218.3.25** `virtual void OsclSocketIBase::Recv (RecvParam &, OsclSocketRequestAO &)` [pure  
virtual]

Implemented in [OsclSocketI](#).

**6.218.3.26** `virtual void OsclSocketIBase::RecvFrom (RecvFromParam &, OsclSocketRequestAO  
&)` [pure virtual]

Implemented in [OsclSocketI](#).

**6.218.3.27** `virtual void OsclSocketIBase::RecvFromSuccess (RecvFromParam &)` [pure  
virtual]

Implemented in [OsclSocketI](#).

**6.218.3.28** `virtual void OsclSocketIBase::RecvSuccess (RecvParam &)` [pure virtual]

Implemented in [OsclSocketI](#).

**6.218.3.29** `virtual void OsclSocketIBase::Send (SendParam &, OsclSocketRequestAO &)` [pure  
virtual]

Implemented in [OsclSocketI](#).

**6.218.3.30** `virtual void OsclSocketIBase::SendSuccess (SendParam &)` [pure virtual]

Implemented in [OsclSocketI](#).

**6.218.3.31** `virtual void OsclSocketIBase::SendTo (SendToParam &, OsclSocketRequestAO &)`  
[pure virtual]

Implemented in [OsclSocketI](#).

**6.218.3.32 virtual void OsclSocketIBase::SendToSuccess ([SendToParam](#) &)** [pure virtual]

Implemented in [OsclSocketI](#).

**6.218.3.33 virtual void OsclSocketIBase::Shutdown ([ShutdownParam](#) &, [OsclSocketRequestAO](#) &)** [pure virtual]

Implemented in [OsclSocketI](#).

## 6.218.4 Friends And Related Function Documentation

**6.218.4.1 friend class OsclSocketMethod** [friend]

**6.218.4.2 friend class OsclSocketRequest** [friend]

**6.218.4.3 friend class OsclSocketRequestAO** [friend]

**6.218.4.4 friend class OsclTCPSocket** [friend]

Reimplemented in [OsclSocketI](#).

**6.218.4.5 friend class OsclUDPSocket** [friend]

Reimplemented in [OsclSocketI](#).

## 6.218.5 Field Documentation

**6.218.5.1 [Oscl\\_DefAlloc](#)& OsclSocketIBase::iAlloc** [protected]

**6.218.5.2 [OsclSocketServI](#)\* OsclSocketIBase::iSocketServ** [protected]

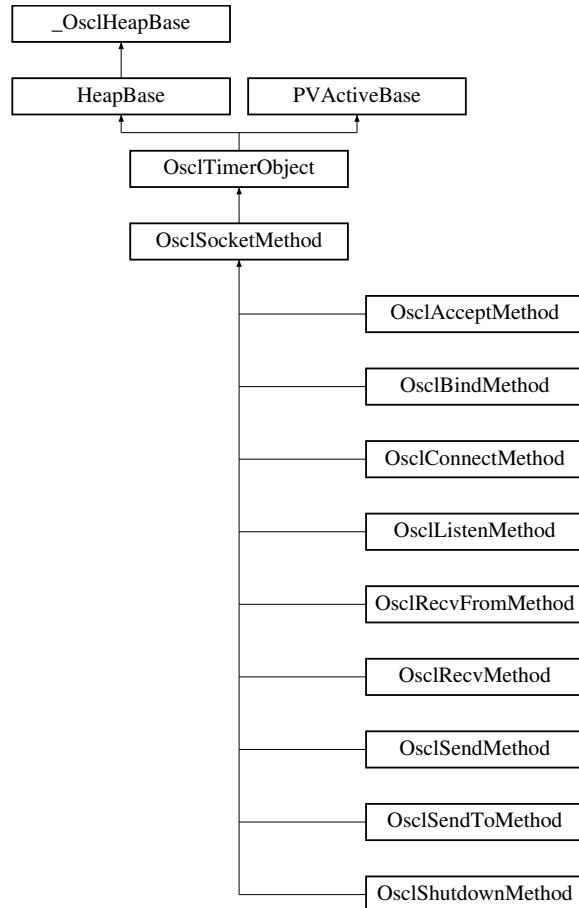
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_imp\\_base.h](#)

## 6.219 OsclSocketMethod Class Reference

```
#include <oscl_socket_method.h>
```

Inheritance diagram for OsclSocketMethod::



### Public Methods

- [OsclSocketMethod \(OsclIPSocketI &aContainer, const char \\*name, TPVSocketFxn fxn\)](#)
- virtual [~OsclSocketMethod \(\)](#)
- void [Abort \(\)](#)
- void [AbortAll \(\)](#)
- void [CancelMethod \(\)](#)
- [Oscl\\_DefAlloc & Alloc \(\)](#)

### Data Fields

- [OsclIPSocketI & iContainer](#)
- [TPVSocketFxn iSocketFxn](#)

## Protected Methods

- void [ConstructL \(OsclSocketRequestAO \\*aAO\)](#)
- bool [StartMethod \(int32 aTimeoutMsec\)](#)
- void [MethodDone \(\)](#)
- void [Run \(\)](#)

## Protected Attributes

- [OsclSocketRequestAO \\* iSocketRequestAO](#)

### 6.219.1 Detailed Description

OsclSocketMethod is the base class for all socket methods. Two AOs are required for each socket operation— one to provide a timeout, and one to detect request completion. The OsclSocketMethod class implements the timeout and contains the request completion AO.

### 6.219.2 Constructor & Destructor Documentation

**6.219.2.1 OsclSocketMethod::OsclSocketMethod** ([OsclIPSocketI & aContainer](#), [const char \\* name](#), [TPVSocketFxn ffn](#)) [inline]

**6.219.2.2 virtual OsclSocketMethod::~OsclSocketMethod ()** [inline, virtual]

### 6.219.3 Member Function Documentation

**6.219.3.1 void OsclSocketMethod::Abort ()** [inline]

**6.219.3.2 void OsclSocketMethod::AbortAll ()** [inline]

**6.219.3.3 Oscl\_DefAlloc& OsclSocketMethod::Alloc ()** [inline]

**6.219.3.4 void OsclSocketMethod::CancelMethod ()** [inline]

**6.219.3.5 void OsclSocketMethod::ConstructL (OsclSocketRequestAO \* aAO)** [inline, protected]

**6.219.3.6 void OsclSocketMethod::MethodDone ()** [inline, protected]

**6.219.3.7 void OsclSocketMethod::Run ()** [protected, virtual]

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's `WaitForAnyRequest()` function completes.

Before calling this active object's `Run()` function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request

2. marked this active object's request as complete (i.e. the request is no longer outstanding)

[Run\(\)](#) runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls [ExecError\(\)](#) to handle the leave.

Note that once the active scheduler's [Start\(\)](#) function has been called, all user code is run under one of the program's active object's [Run\(\)](#) or [RunError\(\)](#) functions.

Implements [PVActiveBase](#).

**6.219.3.8 bool OsclSocketMethod::StartMethod (int32 *aTimeoutMsec*) [protected]**

## 6.219.4 Field Documentation

**6.219.4.1 OsclIPSocketI& OsclSocketMethod::iContainer**

**6.219.4.2 TPVSocketFxn OsclSocketMethod::iSocketFxn**

**6.219.4.3 OsclSocketRequestAO\* OsclSocketMethod::iSocketRequestAO [protected]**

The documentation for this class was generated from the following file:

- [oscl\\_socket\\_method.h](#)

## 6.220 OsclSocketObserver Class Reference

```
#include <oscl_socket_types.h>
```

### Public Methods

- virtual OSCL\_IMPORT\_REF void [HandleSocketEvent](#) (int32 *aId*, [TPVSocketFxn](#) *aFxn*, [TPVSocketEvent](#) *aEvent*, int32 *aError*)=0
- virtual ~[OsclSocketObserver](#) ()

#### 6.220.1 Detailed Description

Socket event observer. The client implements this to get asynchronous command completion.

#### 6.220.2 Constructor & Destructor Documentation

**6.220.2.1 virtual OsclSocketObserver::~OsclSocketObserver () [inline, virtual]**

#### 6.220.3 Member Function Documentation

**6.220.3.1 virtual OSCL\_IMPORT\_REF void OsclSocketObserver::HandleSocketEvent (int32 *aId*, [TPVSocketFxn](#) *aFxn*, [TPVSocketEvent](#) *aEvent*, int32 *aError*) [pure virtual]**

Socket Event callback.

##### Parameters:

***aId***: The ID that was supplied when the socket was created.

***aFxn***: Type of socket function call.

***aEvent***: Function completion event. Will be EPVSocketSuccess, EPVSocketTimeout, or EPVSocketFailure.

***aError***: When the event is EPVSocketFailure, this may contain a platform-specific error code, or zero if none is available.

The documentation for this class was generated from the following file:

- [oscl\\_socket\\_types.h](#)

## 6.221 OsclSocketRequest Class Reference

```
#include <oscl_socket_request.h>
```

### Public Methods

- [OsclSocketRequest \(\)](#)
- [TPVSocketFxn Fxn \(\)](#)
- void [CancelRequest \(\)](#)
- void [Activate \(SocketRequestParam \\*iParam, OsclSocketRequestAO &a\)](#)
- void [Complete \(OsclSocketServRequestQElem \\*, int32 aStatus, int32 aSockErr=0\)](#)

### Data Fields

- [OsclSocketRequestAO \\* iSocketRequestAO](#)
- [SocketRequestParam \\* iParam](#)
- [OsclSocketI \\* iSocketI](#)

#### 6.221.1 Detailed Description

This class defines the request interface to the PV socket server.

#### 6.221.2 Constructor & Destructor Documentation

##### 6.221.2.1 [OsclSocketRequest::OsclSocketRequest \(\) \[inline\]](#)

#### 6.221.3 Member Function Documentation

##### 6.221.3.1 [void OsclSocketRequest::Activate \(SocketRequestParam \\* iParam, OsclSocketRequestAO & a\)](#)

##### 6.221.3.2 [void OsclSocketRequest::CancelRequest \(\)](#)

##### 6.221.3.3 [void OsclSocketRequest::Complete \(OsclSocketServRequestQElem \\*, int32 aStatus, int32 aSockErr = 0\)](#)

##### 6.221.3.4 [TPVSocketFxn OsclSocketRequest::Fxn \(\) \[inline\]](#)

#### 6.221.4 Field Documentation

##### 6.221.4.1 [SocketRequestParam\\* OsclSocketRequest::iParam](#)

##### 6.221.4.2 [OsclSocketI\\* OsclSocketRequest::iSocketI](#)

##### 6.221.4.3 [OsclSocketRequestAO\\* OsclSocketRequest::iSocketRequestAO](#)

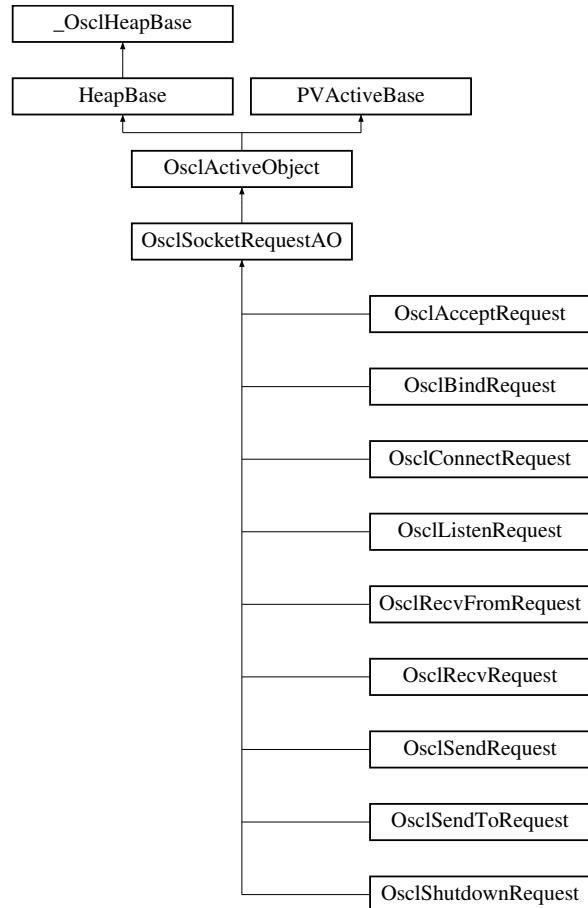
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_request.h](#)

## 6.222 OsclSocketRequestAO Class Reference

```
#include <oscl_socket_method.h>
```

Inheritance diagram for OsclSocketRequestAO::



### Public Methods

- void [ConstructL \(\)](#)

### Protected Methods

- [OsclSocketRequestAO \(OsclSocketMethod &aContainer, const char \\*name\)](#)
- virtual [~OsclSocketRequestAO \(\)](#)
- [OsclAny \\* NewRequest \(const uint32 size\)](#)
- void [CleanupParam \(bool deallocate=false\)](#)
- void [Abort \(\)](#)
- void [RequestDone \(\)](#)
- int [GetSocketError \(\)](#)
- void [DoCancel \(\)](#)
- void [Run \(\)](#)

- virtual void [Success \(\)](#)
- [OsclSocketI \\* SocketI \(\)](#)
- [OsclSocketObserver \\* SocketObserver \(\)](#)
- uint32 [Id \(\)](#)
- [Oscl\\_DefAlloc & Alloc \(\)](#)

## Protected Attributes

- [OsclSocketMethod & iContainer](#)
- int32 [iSocketError](#)
- [SocketRequestParam \\* iParam](#)
- uint32 [iParamSize](#)

## Friends

- class [OsclSocketI](#)
- class [OsclSocketMethod](#)
- class [OsclSocketRequest](#)

### 6.222.1 Detailed Description

This is the base class for all the AOs that interact with the socket server. This object is contained within an [OsclSocketMethod](#) object

### 6.222.2 Constructor & Destructor Documentation

**6.222.2.1 OsclSocketRequestAO::OsclSocketRequestAO ([OsclSocketMethod & aContainer](#), const char \* *name*) [inline, protected]**

**6.222.2.2 virtual OsclSocketRequestAO::~OsclSocketRequestAO () [inline, protected, virtual]**

### 6.222.3 Member Function Documentation

**6.222.3.1 void OsclSocketRequestAO::Abort () [inline, protected]**

**6.222.3.2 [Oscl\\_DefAlloc& OsclSocketRequestAO::Alloc \(\) \[inline, protected\]](#)**

**6.222.3.3 void OsclSocketRequestAO::CleanupParam (bool *deallocate* = false) [protected]**

**6.222.3.4 void OsclSocketRequestAO::ConstructL () [inline]**

**6.222.3.5 void OsclSocketRequestAO::DoCancel () [inline, protected, virtual]**

Cancel request handler. This gets called by scheduler when the request is cancelled. The default routine will complete the request. If any additional action is needed, the derived class may override this. If the derived class does override DoCancel, it must complete the request.

Reimplemented from [OsclActiveObject](#).

6.222.3.6 **int OsclSocketRequestAO::GetSocketError ()** [inline, protected]

6.222.3.7 **uint32 OsclSocketRequestAO::Id ()** [inline, protected]

6.222.3.8 **OsclAny\* OsclSocketRequestAO::NewRequest (const uint32 size)** [protected]

6.222.3.9 **void OsclSocketRequestAO::RequestDone ()** [inline, protected]

6.222.3.10 **void OsclSocketRequestAO::Run ()** [protected, virtual]

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's WaitForAnyRequest() function completes.

Before calling this active object's Run() function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request
2. marked this active object's request as complete (i.e. the request is no longer outstanding)

Run() runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls ExecError() to handle the leave.

Note that once the active scheduler's Start() function has been called, all user code is run under one of the program's active object's Run() or RunError() functions.

Implements [PVActiveBase](#).

6.222.3.11 **OsclSocketI\* OsclSocketRequestAO::SocketI ()** [inline, protected]

6.222.3.12 **OsclSocketObserver\* OsclSocketRequestAO::SocketObserver ()** [inline, protected]

6.222.3.13 **virtual void OsclSocketRequestAO::Success ()** [inline, protected, virtual]

Reimplemented in [OsclRecvRequest](#), [OsclRecvFromRequest](#), [OsclSendRequest](#), and [OsclSendToRequest](#).

## 6.222.4 Friends And Related Function Documentation

6.222.4.1 **friend class OsclSocketI** [friend]

6.222.4.2 **friend class OsclSocketMethod** [friend]

6.222.4.3 **friend class OsclSocketRequest** [friend]

## 6.222.5 Field Documentation

6.222.5.1 **OsclSocketMethod& OsclSocketRequestAO::iContainer** [protected]

6.222.5.2 **SocketRequestParam\* OsclSocketRequestAO::iParam** [protected]

6.222.5.3 **uint32 OsclSocketRequestAO::iParamSize** [protected]

6.222.5.4 **int32 OsclSocketRequestAO::iSocketError** [protected]

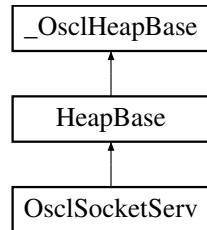
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_method.h](#)

## 6.223 OsclSocketServ Class Reference

```
#include <oscl_socket.h>
```

Inheritance diagram for OsclSocketServ::



### Public Methods

- OSCL\_IMPORT\_REF ~OsclSocketServ ()
- OSCL\_IMPORT\_REF int32 Connect (uint32 aMessageSlots=8)
- OSCL\_IMPORT\_REF void Close (bool aCleanup=true)

### Static Public Methods

- OSCL\_IMPORT\_REF OsclSocketServ \* NewL (Oscl\_DefAlloc &alloc)

### Friends

- class OsclTCPSocket
- class OsclUDPSocket
- class OsclDNS

### 6.223.1 Constructor & Destructor Documentation

#### 6.223.1.1 OSCL\_IMPORT\_REF OsclSocketServ::~OsclSocketServ ()

Destructor. The server object must be deleted using the same allocator used in the NewL call.

### 6.223.2 Member Function Documentation

#### 6.223.2.1 OSCL\_IMPORT\_REF void OsclSocketServ::Close (bool aCleanup = true)

Close socket server. This is a synchronous method.

##### Parameters:

*aCleanup*: cleanup the socket system? the socket cleanup should only be done when all parts of the application are done using sockets.

**6.223.2.2 OSCL\_IMPORT\_REF int32 OsclSocketServ::Connect (uint32 *aMessageSlots* = 8)**

Connect to socket server. This is a synchronous method.

**Parameters:**

*Number* of message slots.

**Returns:**

Returns OsclErrNone for success, or a platform-specific code.

**6.223.2.3 OSCL\_IMPORT\_REF OsclSocketServ\* OsclSocketServ::NewL (Oscl\_DefAlloc & *alloc*)  
[static]**

Create a socket server. May leave if failure.

**Parameters:**

*alloc*: Memory allocator.

**Returns:**

Returns pointer to socket server

**6.223.3 Friends And Related Function Documentation****6.223.3.1 friend class OsclDNS [friend]****6.223.3.2 friend class OsclTCPSocket [friend]****6.223.3.3 friend class OsclUDPSocket [friend]**

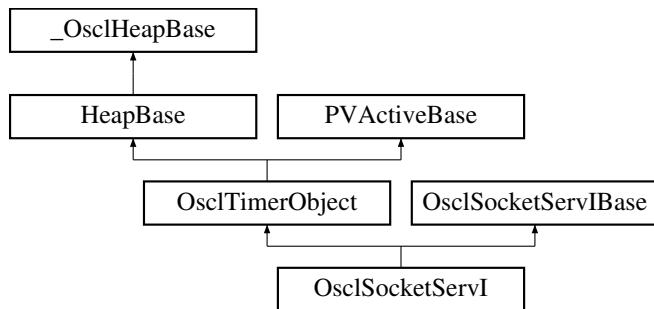
The documentation for this class was generated from the following file:

- [oscl\\_socket.h](#)

## 6.224 OsclSocketServI Class Reference

```
#include <oscl_socket_serv_imp_pv.h>
```

Inheritance diagram for OsclSocketServI::



### Public Methods

- int32 [Connect](#) (uint32 aMessageSlots)
- void [Close](#) (bool)
- bool [IsServerThread](#) ()

### Static Public Methods

- OsclSocketServI \* [NewL](#) (Oscl\_DefAlloc &a)

### Friends

- class [OsclSocketServRequestList](#)
- class [LoopbackSocket](#)
- class [OsclTCPSocketI](#)
- class [OsclUDPSocketI](#)
- class [OsclSocketI](#)
- class [OsclDNSI](#)
- class [OsclSocketRequest](#)
- class [OsclSocketServ](#)

### 6.224.1 Detailed Description

PV socket server implementation

### 6.224.2 Member Function Documentation

#### 6.224.2.1 void OsclSocketServI::Close (bool) [virtual]

Implements [OsclSocketServIBase](#).

6.224.2.2 **int32 OsclSocketServI::Connect (uint32 *aMessageSlots*)** [virtual]

Implements [OsclSocketServIBase](#).

6.224.2.3 **bool OsclSocketServI::IsServerThread ()**

6.224.2.4 **OsclSocketServI\* OsclSocketServI::NewL (Oscl\_DefAlloc & *a*)** [static]

### 6.224.3 Friends And Related Function Documentation

6.224.3.1 **friend class LoopbackSocket** [friend]

6.224.3.2 **friend class OsclDNSI** [friend]

6.224.3.3 **friend class OsclSocketI** [friend]

6.224.3.4 **friend class OsclSocketRequest** [friend]

6.224.3.5 **friend class OsclSocketServ** [friend]

6.224.3.6 **friend class OsclSocketServRequestList** [friend]

6.224.3.7 **friend class OsclTCPSocketI** [friend]

6.224.3.8 **friend class OsclUDPSocketI** [friend]

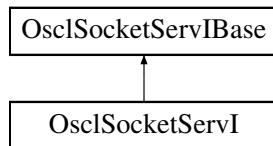
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_serv\\_imp\\_pv.h](#)

## 6.225 OsclSocketServIBase Class Reference

```
#include <oscl_socket_serv_imp_base.h>
```

Inheritance diagram for OsclSocketServIBase::



### Public Methods

- virtual ~[OsclSocketServIBase](#) ()
- virtual int32 [Connect](#) (uint32 aMessageSlots)=0
- virtual void [Close](#) (bool)=0

### Data Fields

- [PVLogger \\* iLogger](#)

### Protected Types

- enum [TSocketServState](#) { [ESocketServ\\_Idle](#), [ESocketServ\\_Connected](#), [ESocketServ\\_Error](#) }

### Protected Methods

- [OsclSocketServIBase \(Oscl\\_DefAlloc &a\)](#)
- [TSocketServState State \(\) const](#)
- [bool IsServConnected \(\) const](#)

### Protected Attributes

- [Oscl\\_DefAlloc & iAlloc](#)
- [TSocketServState iServState](#)
- [int iServError](#)

### 6.225.1 Detailed Description

Base class common to all implementations

### 6.225.2 Member Enumeration Documentation

#### 6.225.2.1 enum OsclSocketServIBase::TSocketServState [protected]

Enumeration values:

[ESocketServ\\_Idle](#)

**ESocketServ\_Connected**

**ESocketServ\_Error**

### 6.225.3 Constructor & Destructor Documentation

**6.225.3.1 virtual OsclSocketServIBase::~OsclSocketServIBase () [inline, virtual]**

**6.225.3.2 OsclSocketServIBase::OsclSocketServIBase ([Oscl\\_DefAlloc](#) & *a*) [inline, protected]**

### 6.225.4 Member Function Documentation

**6.225.4.1 virtual void OsclSocketServIBase::Close (bool) [pure virtual]**

Implemented in [OsclSocketServI](#).

**6.225.4.2 virtual int32 OsclSocketServIBase::Connect (uint32 *aMessageSlots*) [pure virtual]**

Implemented in [OsclSocketServI](#).

**6.225.4.3 bool OsclSocketServIBase::IsServConnected () const [inline, protected]**

**6.225.4.4 [TSocketServState](#) OsclSocketServIBase::State () const [inline, protected]**

### 6.225.5 Field Documentation

**6.225.5.1 [Oscl\\_DefAlloc](#)& OsclSocketServIBase::iAlloc [protected]**

**6.225.5.2 [PVLogger](#)\* OsclSocketServIBase::iLogger**

**6.225.5.3 int OsclSocketServIBase::iServerError [protected]**

**6.225.5.4 [TSocketServState](#) OsclSocketServIBase::iServState [protected]**

The documentation for this class was generated from the following file:

- [oscl\\_socket\\_serv\\_imp\\_base.h](#)

## 6.226 OsclSocketServRequestList Class Reference

```
#include <oscl_socket_serv_imp_reqlist.h>
```

### Public Methods

- [OsclSocketServRequestList \(\)](#)
- [void Add \(OsclSocketRequest \\*\)](#)
- [void StartCancel \(OsclSocketRequest \\*\)](#)
- [void Open \(OsclSocketServI \\*s\)](#)
- [void Close \(\)](#)
- [void Wakeup \(\)](#)
- [void WaitOnRequests \(\)](#)
- [void Remove \(OsclSocketServRequestQElem \\*aElem\)](#)

### Friends

- class [OsclSocketServI](#)

### 6.226.1 Detailed Description

PV socket server request queue

### 6.226.2 Constructor & Destructor Documentation

#### 6.226.2.1 OsclSocketServRequestList::OsclSocketServRequestList ()

### 6.226.3 Member Function Documentation

#### 6.226.3.1 void OsclSocketServRequestList::Add ([OsclSocketRequest \\*](#))

#### 6.226.3.2 void OsclSocketServRequestList::Close ()

#### 6.226.3.3 void OsclSocketServRequestList::Open ([OsclSocketServI \\* s](#))

#### 6.226.3.4 void OsclSocketServRequestList::Remove ([OsclSocketServRequestQElem \\* aElem](#)) [inline]

#### 6.226.3.5 void OsclSocketServRequestList::StartCancel ([OsclSocketRequest \\*](#))

#### 6.226.3.6 void OsclSocketServRequestList::WaitOnRequests ()

#### 6.226.3.7 void OsclSocketServRequestList::Wakeup ()

### 6.226.4 Friends And Related Function Documentation

#### 6.226.4.1 friend class OsclSocketServI [friend]

The documentation for this class was generated from the following file:

- 
- [oscl\\_socket\\_serv\\_imp\\_reqlist.h](#)

## 6.227 OsclSocketServRequestQElem Class Reference

```
#include <oscl_socket_serv_imp_reqlist.h>
```

### Public Methods

- [OsclSocketServRequestQElem \(OsclSocketRequest \\*r\)](#)

### Data Fields

- [OsclSocketRequest \\* iSocketRequest](#)
- [uint8 iSelect](#)
- [bool iCancel](#)

#### 6.227.1 Constructor & Destructor Documentation

6.227.1.1 [OsclSocketServRequestQElem::OsclSocketServRequestQElem \(OsclSocketRequest \\* r\)](#)  
[inline]

#### 6.227.2 Field Documentation

6.227.2.1 [bool OsclSocketServRequestQElem::iCancel](#)

6.227.2.2 [uint8 OsclSocketServRequestQElem::iSelect](#)

6.227.2.3 [OsclSocketRequest\\* OsclSocketServRequestQElem::iSocketRequest](#)

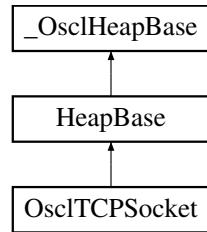
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_serv\\_imp\\_reqlist.h](#)

## 6.228 OsclTCPSocket Class Reference

```
#include <oscl_socket.h>
```

Inheritance diagram for OsclTCPSocket::



### Public Methods

- OSCL\_IMPORT\_REF ~OsclTCPSocket ()
- OSCL\_IMPORT\_REF int32 Close ()
- OSCL\_IMPORT\_REF int32 Bind (OsclNetworkAddress &aAddress)
- OSCL\_IMPORT\_REF TPVSocketEvent BindAsync (OsclNetworkAddress &aAddress, int32 aTimeoutMsec=(-1))
- OSCL\_IMPORT\_REF void CancelBind ()
- OSCL\_IMPORT\_REF int32 Listen (int32 aQueueSize)
- OSCL\_IMPORT\_REF TPVSocketEvent ListenAsync (int32 aQueueSize, int32 aTimeoutMsec=(-1))
- OSCL\_IMPORT\_REF void CancelListen ()
- OSCL\_IMPORT\_REF OsclTCPSocket \* GetAcceptedSocketL (uint32 aId)
- OSCL\_IMPORT\_REF uint8 \* GetRecvData (int32 \*aLength)
- OSCL\_IMPORT\_REF uint8 \* GetSendData (int32 \*aLength)
- OSCL\_IMPORT\_REF TPVSocketEvent Connect (OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void CancelConnect ()
- OSCL\_IMPORT\_REF TPVSocketEvent Shutdown (TPVSocketShutdown aHow, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void CancelShutdown ()
- OSCL\_IMPORT\_REF TPVSocketEvent Accept (int32 aTimeout=-1)
- OSCL\_IMPORT\_REF void CancelAccept ()
- OSCL\_IMPORT\_REF TPVSocketEvent Send (const uint8 \*aPtr, uint32 aLen, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void CancelSend ()
- OSCL\_IMPORT\_REF TPVSocketEvent Recv (uint8 \*aPtr, uint32 aMaxLen, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void CancelRecv ()

### Static Public Methods

- OSCL\_IMPORT\_REF OsclTCPSocket \* NewL (Oscl\_DefAlloc &alloc, OsclSocketServ &aServ, OsclSocketObserver \*aObserver, uint32 aId)

### 6.228.1 Detailed Description

The TCP Socket class

### 6.228.2 Constructor & Destructor Documentation

#### 6.228.2.1 OSCL\_IMPORT\_REF OsclTCPSocket::~OsclTCPSocket ()

Destructor. The object must be deleted using the same allocator used in the NewL call.

### 6.228.3 Member Function Documentation

#### 6.228.3.1 OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::Accept (int32 *aTimeout* = -1)

Accept incoming connections. This is an asynchronous method.

**Parameters:**

*aTimeoutMsec*: Timeout in milliseconds, or (-1) for infinite wait.

**Returns:**

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

#### 6.228.3.2 OSCL\_IMPORT\_REF int32 OsclTCPSocket::Bind ([OsclNetworkAddress](#) & *aAddress*)

Bind a TCP socket to an address. This is a synchronous method.

**Parameters:**

*aAddress*: Bind address.

**Returns:**

Returns OsclErrNone for success, or a platform-specific error code.

#### 6.228.3.3 OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::BindAsync ([OsclNetworkAddress](#) & *aAddress*, int32 *aTimeoutMsec* = (-1))

Bind a TCP socket to an address. This is an asynchronous method.

**Parameters:**

*aAddress*: Bind address.

*aTimeoutMsec*: Optional timeout. Use a negative value for infinite wait.

**Returns:**

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

**6.228.3.4 OSCL\_IMPORT\_REF void OsclTCPSocket::CancelAccept ()**

Cancel Accept

This method will cancel any pending Accept operation on the current socket, causing the Accept to complete with error EPVSocketCancel. If there is no pending Accept operation, this method will have no effect.

**6.228.3.5 OSCL\_IMPORT\_REF void OsclTCPSocket::CancelBind ()**

Cancel Bind

This method will cancel any pending BindAsync operation on the current socket, causing the BindAsync to complete with error EPVSocketCancel. If there is no pending BindAsync operation, this method will have no effect.

**6.228.3.6 OSCL\_IMPORT\_REF void OsclTCPSocket::CancelConnect ()**

Cancel Connect

This method will cancel any pending Connect operation on the current socket, causing the Connect to complete with error EPVSocketCancel. If there is no pending Connect operation, this method will have no effect.

**6.228.3.7 OSCL\_IMPORT\_REF void OsclTCPSocket::CancelListen ()**

Cancel Async Listen

This method will cancel any pending ListenAsync operation on the current socket, causing the Listen to complete with error EPVSocketCancel. If there is no pending Listen operation, this method will have no effect.

**6.228.3.8 OSCL\_IMPORT\_REF void OsclTCPSocket::CancelRecv ()**

Cancel Recv

This method will cancel any pending Recv operation on the current socket, causing the Recv to complete with error EPVSocketCancel. If there is no pending Recv operation, this method will have no effect.

**6.228.3.9 OSCL\_IMPORT\_REF void OsclTCPSocket::CancelSend ()**

Cancel Send

This method will cancel any pending Send operation on the current socket, causing the Send to complete with error EPVSocketCancel. If there is no pending Send operation, this method will have no effect.

**6.228.3.10 OSCL\_IMPORT\_REF void OsclTCPSocket::CancelShutdown ()**

Cancel Shutdown

This method will cancel any pending Shutdown operation on the current socket, causing the Shutdown to complete with error EPVSocketCancel. If there is no pending Shutdown operation, this method will have no effect.

### 6.228.3.11 OSCL\_IMPORT\_REF int32 OsclTCPSocket::Close ()

Close a TCP socket. This is a synchronous method.

Once it is closed a socket cannot be re-opened. Sockets are automatically closed when they are deleted. This method may be used to see any error code returned from the platform's socket close call.

**Returns:**

Returns OsclErrNone for success, or a platform-specific error code.

### 6.228.3.12 OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::Connect (OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1)

Connect to an address. This is an asynchronous method.

**Parameters:**

*aAddress*: a network address.

*aTimeoutMsec*: Timeout in milliseconds, or (-1) for infinite wait.

**Returns:**

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

### 6.228.3.13 OSCL\_IMPORT\_REF OsclTCPSocket\* OsclTCPSocket::GetAcceptedSocketL (uint32 aId)

Retrieve the accept socket after a successful Accept operation. This is a synchronous method.

**Parameters:**

*aId*: Socket ID. The caller must assign an ID to each socket. The ID is used to identify the socket in observer callbacks.

**Returns:**

Returns pointer to socket, or NULL if error. Note: The caller is responsible for deleting any accepted socket that it retrieves.

### 6.228.3.14 OSCL\_IMPORT\_REF uint8\* OsclTCPSocket::GetRecvData (int32 \* aLength)

Retrieve the received data after a successful Recv operation. This is a synchronous method.

**Parameters:**

*aLength*: (output) number of bytes of data received.

**Returns:**

Returns pointer to received data, or NULL if none.

**6.228.3.15 OSCL\_IMPORT\_REF uint8\* OsclTCPSocket::GetSendData (int32 \* aLength)**

Retrieve the sent data after a successful Send operation. This is a synchronous method.

**Parameters:**

*aLength*: (output) number of bytes of data sent.

**Returns:**

Returns pointer to sent data, or NULL if none.

**6.228.3.16 OSCL\_IMPORT\_REF int32 OsclTCPSocket::Listen (int32 aQueueSize)**

Listen. This is a synchronous method.

**Parameters:**

*aQueueSize*: Queue size.

**Returns:**

Returns OsclErrNone for success, or a platform-specific error code.

**6.228.3.17 OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::ListenAsync (int32 aQueueSize, int32 aTimeoutMsec = (-1))**

ListenAsync This is an asynchronous method.

**Parameters:**

*aQueueSize*: Queue size.

*aTimeoutMsec*: Optional timeout. Use a negative value for infinite wait.

**Returns:**

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

**6.228.3.18 OSCL\_IMPORT\_REF OsclTCPSocket\* OsclTCPSocket::NewL (Oscl\_DefAlloc & alloc, OsclSocketServ & aServ, OsclSocketObserver \* aObserver, uint32 aId) [static]**

Create a TCP Socket. May leave if failure.

**Parameters:**

*alloc*: Memory allocator.

*aServ*: Socket server. Must be connected.

*aObserver*: Socket observer.

*aId*: Socket ID. The caller must assign an ID to each socket. The ID is used to identify the socket in observer callbacks.

**Returns:**

Returns pointer to socket.

**6.228.3.19 OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::Recv (uint8 \* aPtr, uint32 aMaxLen, int32 aTimeoutMsec = -1)**

Receive Data. This is an asynchronous method.

**Parameters:**

*aPtr*: Buffer for received data.

*aMaxLen*: Length of buffer.

*aTimeoutMsec*: Timeout in milliseconds, or (-1) for infinite wait.

**Returns:**

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

**6.228.3.20 OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::Send (const uint8 \* aPtr, uint32 aLen, int32 aTimeoutMsec = -1)**

Send Data. This is an asynchronous method.

**Parameters:**

*aPtr*: Data to send.

*aLen*: Length of data to send.

*aTimeoutMsec*: Timeout in milliseconds, or (-1) for infinite wait.

**Returns:**

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

**6.228.3.21 OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::Shutdown (TPVSocketShutdown aHow, int32 aTimeoutMsec = -1)**

Shutdown a socket. This is an asynchronous method.

**Parameters:**

*aHow*: type of shutdown

*aTimeoutMsec*: Timeout in milliseconds, or (-1) for infinite wait.

**Returns:**

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

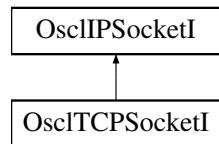
The documentation for this class was generated from the following file:

- [oscl\\_socket.h](#)

## 6.229 OsclTCPSocketI Class Reference

```
#include <oscl_tcp_socket.h>
```

Inheritance diagram for OsclTCPSocketI::



### Public Methods

- virtual ~OsclTCPSocketI ()
- int32 [Close \(\)](#)
- int32 [Listen \(int aQueueSize\)](#)
- OsclTCPSocketI \* [GetAcceptedSocketL \(uint32 aId\)](#)
- uint8 \* [GetRecvData \(int32 \\*aLength\)](#)
- uint8 \* [GetSendData \(int32 \\*aLength\)](#)
- [TPVSocketEvent BindAsync \(OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1\)](#)
- void [CancelBind \(\)](#)
- [TPVSocketEvent ListenAsync \(uint32 qsize, int32 aTimeoutMsec=-1\)](#)
- void [CancelListen \(\)](#)
- [TPVSocketEvent Connect \(OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1\)](#)
- void [CancelConnect \(\)](#)
- [TPVSocketEvent Shutdown \(TPVSocketShutdown aHow, int32 aTimeoutMsec=-1\)](#)
- void [CancelShutdown \(\)](#)
- [TPVSocketEvent Accept \(int32 aTimeout=-1\)](#)
- void [CancelAccept \(\)](#)
- [TPVSocketEvent Send \(const uint8 \\*&aPtr, uint32 aLen, int32 aTimeoutMsec=-1\)](#)
- void [CancelSend \(\)](#)
- [TPVSocketEvent Recv \(uint8 \\*&aPtr, uint32 aMaxLen, int32 aTimeoutMsec=-1\)](#)
- void [CancelRecv \(\)](#)

### Static Public Methods

- OsclTCPSocketI \* [NewL \(Oscl\\_DefAlloc &a, OsclSocketServI \\*aServ, OsclSocketObserver \\*aObserver, uint32 aId\)](#)

#### 6.229.1 Detailed Description

Internal implementation class for [OsclTCPSocket](#)

## 6.229.2 Constructor & Destructor Documentation

6.229.2.1 **virtual OsclTCPSocketI::~OsclTCPSocketI () [virtual]**

## 6.229.3 Member Function Documentation

6.229.3.1 **TPVSocketEvent OsclTCPSocketI::Accept (int32 *aTimeout* = -1) [inline]**

6.229.3.2 **TPVSocketEvent OsclTCPSocketI::BindAsync (OsclNetworkAddress & *aAddress*, int32 *aTimeoutMsec* = -1) [inline]**

6.229.3.3 **void OsclTCPSocketI::CancelAccept () [inline]**

6.229.3.4 **void OsclTCPSocketI::CancelBind () [inline]**

6.229.3.5 **void OsclTCPSocketI::CancelConnect () [inline]**

6.229.3.6 **void OsclTCPSocketI::CancelListen () [inline]**

6.229.3.7 **void OsclTCPSocketI::CancelRecv () [inline]**

6.229.3.8 **void OsclTCPSocketI::CancelSend () [inline]**

6.229.3.9 **void OsclTCPSocketI::CancelShutdown () [inline]**

6.229.3.10 **int32 OsclTCPSocketI::Close () [virtual]**

Implements [OsclIPSocketI](#).

6.229.3.11 **TPVSocketEvent OsclTCPSocketI::Connect (OsclNetworkAddress & *aAddress*, int32 *aTimeoutMsec* = -1) [inline]**

6.229.3.12 **OsclTCPSocketI\* OsclTCPSocketI::GetAcceptedSocketL (uint32 *aId*)**

6.229.3.13 **uint8 \* OsclTCPSocketI::GetRecvData (int32 \* *aLength*) [inline, virtual]**

Implements [OsclIPSocketI](#).

6.229.3.14 **uint8 \* OsclTCPSocketI::GetSendData (int32 \* *aLength*) [inline, virtual]**

Implements [OsclIPSocketI](#).

- 6.229.3.15 **int32** OsclTCPSocketI::Listen (*int aQueueSize*) [inline]
- 6.229.3.16 **TPVSocketEvent** OsclTCPSocketI::ListenAsync (*uint32 qsize, int32 aTimeoutMsec = -1*) [inline]
- 6.229.3.17 OsclTCPSocketI\* OsclTCPSocketI::NewL (**Oscl\_DefAlloc** & *a*, **OsclSocketServI** \* *aServ*, **OsclSocketObserver** \* *aObserver*, *uint32 aId*) [static]
- 6.229.3.18 **TPVSocketEvent** OsclTCPSocketI::Recv (*uint8 \*& aPtr, uint32 aMaxLen, int32 aTimeoutMsec = -1*) [inline]
- 6.229.3.19 **TPVSocketEvent** OsclTCPSocketI::Send (*const uint8 \*& aPtr, uint32 aLen, int32 aTimeoutMsec = -1*) [inline]
- 6.229.3.20 **TPVSocketEvent** OsclTCPSocketI::Shutdown (**TPVSocketShutdown** *aHow, int32 aTimeoutMsec = -1*) [inline]

The documentation for this class was generated from the following file:

- [oscl\\_tcp\\_socket.h](#)

## 6.230 OsclThread Class Reference

```
#include <oscl_thread.h>
```

### Public Methods

- OSCL\_IMPORT\_REF OsclThread ()
- OSCL\_IMPORT\_REF ~OsclThread ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Create (TOsclThreadFuncPtr func, int32 stack\_size, TOsclThreadFuncArg argument, OsclThread\_State state=Start\_on\_creation, bool oIsJoinable=false)
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError GetPriority (OsclThreadPriority &refThreadPriority)
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError SetPriority (OsclThreadPriority ePriority)
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Suspend ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Resume ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Terminate (OsclAny \*exitcode)

### Static Public Methods

- OSCL\_IMPORT\_REF void Exit (OsclAny \*exitcode)
- OSCL\_IMPORT\_REF void EnableKill ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError GetId (TOsclThreadId &refThreadId)
- OSCL\_IMPORT\_REF bool CompareId (TOsclThreadId &t1, TOsclThreadId &t2)
- OSCL\_IMPORT\_REF void SleepMillisec (const int32 msec)

### 6.230.1 Detailed Description

Thread Class. A subset of Thread APIs. It implements platform independent APIs for thread creation, exiting, suspend, resume, priority and termination. With the use of proper defines it implements the basic thread features. It provides an opaque layer through which user doesn't need to worry about OS specific data.

### 6.230.2 Constructor & Destructor Documentation

#### 6.230.2.1 OSCL\_IMPORT\_REF OsclThread::OsclThread ()

Class constructor

#### 6.230.2.2 OSCL\_IMPORT\_REF OsclThread::~OsclThread ()

Class destructor

### 6.230.3 Member Function Documentation

#### 6.230.3.1 OSCL\_IMPORT\_REF bool OsclThread::CompareId (TOsclThreadId & t1, TOsclThreadId & t2) [static]

Static routine to compare whether two thread ID's are equal.

**Parameters:**

*t1, t2*: thread ID passed by the application

**Returns:**

true if equal.

### 6.230.3.2 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclThread::Create (TOsclThreadFuncPtr *func*, int32 *stack\_size*, TOsclThreadFuncArg *argument*, *OsclThread\_State state* = Start\_on\_creation, bool *oIsJoinable* = false)

This routine will create a thread. The thread may be launched immediately or may be created in a suspended state and launched with a Resume call.

**Parameters:**

*func* = Name of the thread Function *stack\_size* = Size of the thread stack. If zero, then the platform-specific default stack size will be used. *argument* = Argument to be passed to thread function *state* = Enumeration which specifies the state of the thread on creation with values Running and Suspend. Note: the Suspend option may not be available on all platforms. If it is not supported, the Create call will return INVALID\_PARAM\_ERROR. *oIsJoinable* = A boolean, which when set to true, creates a Joinable thread. The default value for this is false, which creates a Detached thread. Note 1: When a joinable thread is created, it is imperative to call thread Terminate. Otherwise, it would cause a memory leak. Note 2: This is currently available only for platforms that have support for pthreads.

**Returns:**

eOsclProcError

### 6.230.3.3 OSCL\_IMPORT\_REF void OsclThread::EnableKill () [static]

EnableKill is a static function which can be called by the thread routine in order to enable thread termination without waiting for cancellation points. EnableKill only applies to pthread implementations. For other implementations this function will do nothing.

**Returns:**

None

### 6.230.3.4 OSCL\_IMPORT\_REF void OsclThread::Exit (OsclAny \* *exitcode*) [static]

Exit is a static function which is used to end the current thread. When called it just ends the execution of the current thread.

**Parameters:**

*exitcode* = Exitcode of the thread. This can be used by other threads to know the exit status of this thread.

**Returns:**

None

**6.230.3.5 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclThread::GetId  
(TOsclThreadId & refThreadId) [static]**

Static routine to retrieve ID of calling thread.

**Parameters:**

*Thread* ID passed by the application

**Returns:**

Error code

**6.230.3.6 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclThread::GetPriority  
(OsclThreadPriority & refThreadPriority)**

GetThreadPriority gets the priority of the thread. It takes reference of the input argument and assigns priority to it from one of the already defined priorities.

**Parameters:**

*int16&* refThreadPriority : Output Priority value

**Returns:**

Error code

**6.230.3.7 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclThread::Resume ()**

ResumeThread resumes the suspended thread and brings it into execution.

**Parameters:**

*None*

**Returns:**

Error code Note: this function may not be supported on all platforms, and may return NOT\_ - IMPLEMENTED.

**6.230.3.8 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclThread::SetPriority  
(OsclThreadPriority ePriority)**

SetThreadPriority sets the priority of the thread. It takes priority as the input argument and assigns it to the thread referred.

**Parameters:**

*ePriorityLevel* : Input Priority value

**Returns:**

Error code Note: this function may not be supported on all platforms, and may return NOT\_ - IMPLEMENTED.

**6.230.3.9 OSCL\_IMPORT\_REF void OsclThread::SleepMillisec (const int32 msec) [static]**

Suspend current thread execution for specified time.

**Parameters:**

*msec, t2*: sleep time in milliseconds.

**6.230.3.10 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclThread::Suspend ()**

This API suspends the thread being referred. The thread can later be brought into execution by calling OSCL\_ResumeThread() on it.

**Parameters:**

*None*

**Returns:**

Error code Note: this function may not be supported on all platforms, and may return NOT\_IMPLEMENTED.

**6.230.3.11 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclThread::Terminate  
(OsclAny \* exitcode)**

Terminate a thread other than the calling thread.

Note: for pthread implementations, the Terminate call will block until the thread has terminated. By default, threads will not terminate until a cancellation point is reached. The EnableKill method may be used to override this default behavior and allow immediate termination.

**Parameters:**

*exitcode* = Exitcode of the thread.

**Returns:**

Error code

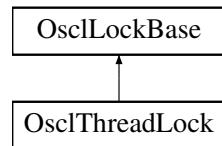
The documentation for this class was generated from the following file:

- [oscl\\_thread.h](#)

## 6.231 OsclThreadLock Class Reference

```
#include <oscl_mutex.h>
```

Inheritance diagram for OsclThreadLock::



### Public Methods

- OSCL\_IMPORT\_REF OsclThreadLock ()
- virtual OSCL\_IMPORT\_REF ~OsclThreadLock ()
- OSCL\_IMPORT\_REF void Lock ()
- OSCL\_IMPORT\_REF void Unlock ()

#### 6.231.1 Detailed Description

An implementation of [OsclLockBase](#) using a mutex

#### 6.231.2 Constructor & Destructor Documentation

##### 6.231.2.1 OSCL\_IMPORT\_REF OsclThreadLock::OsclThreadLock ()

**6.231.2.2 virtual OSCL\_IMPORT\_REF OsclThreadLock::~OsclThreadLock () [virtual]**

#### 6.231.3 Member Function Documentation

##### 6.231.3.1 OSCL\_IMPORT\_REF void OsclThreadLock::Lock () [virtual]

Implements [OsclLockBase](#).

##### 6.231.3.2 OSCL\_IMPORT\_REF void OsclThreadLock::Unlock () [virtual]

Implements [OsclLockBase](#).

The documentation for this class was generated from the following file:

- [oscl\\_mutex.h](#)

## 6.232 OsclTickCount Class Reference

```
#include <oscl_tickcount.h>
```

### Static Public Methods

- uint32 [TickCount \(\)](#)
- uint32 [TickCountFrequency \(\)](#)
- uint32 [TickCountPeriod \(\)](#)
- uint32 [TicksToMsec \(uint32 ticks\)](#)
- uint32 [MsecToTicks \(uint32 msec\)](#)

### 6.232.1 Detailed Description

OsclTickCount class is used to retrieve the system tick count and the tick counter's frequency. The maximum tick count value is equivalent to the maximum uint32 value.

### 6.232.2 Member Function Documentation

#### 6.232.2.1 uint32 OsclTickCount::MsecToTicks (uint32 *msec*) [static]

This function converts milliseconds to ticks

**Returns:**

ticks

#### 6.232.2.2 uint32 OsclTickCount::TickCount () [static]

This function returns the current system tick count

**Returns:**

returns the tick count

#### 6.232.2.3 uint32 OsclTickCount::TickCountFrequency () [static]

This function returns the tick frequency in ticks per second

**Returns:**

ticks per second

#### 6.232.2.4 uint32 OsclTickCount::TickCountPeriod () [static]

This function returns the tick period in microseconds per tick

**Returns:**

microseconds per tick

**6.232.2.5 uint32 OsclTickCount::TicksToMsec (uint32 *ticks*) [static]**

This function converts ticks to milliseconds

**Returns:**

milliseconds

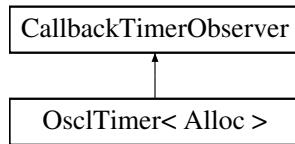
The documentation for this class was generated from the following file:

- [oscl\\_tickcount.h](#)

## 6.233 OsclTimer< Alloc > Class Template Reference

```
#include <oscl_timer.h>
```

Inheritance diagram for OsclTimer< Alloc >::



### Public Types

- `typedef CallbackTimer< Alloc > callback_timer_type`

### Public Methods

- `OsclTimer (const char *name, uint32 frequency=1, int32 priority=OsclActiveObject::EPriorityNominal)`
- `virtual ~OsclTimer ()`
- `void SetObserver (OsclTimerObserver *obs)`
- `void SetFrequency (uint32 frequency)`
- `void SetExactFrequency (uint32 frequency)`
- `void Request (int32 timerID, int32 timeoutInfo, int32 cycles, OsclTimerObserver *obs=0, bool recurring=0)`
- `void Cancel (int32 timerID, int32 timeoutInfo=-1)`
- `void Clear ()`

### Protected Methods

- `void TimerBaseElapsed ()`

### Friends

- `class CallbackTimer< Alloc >`

template<class Alloc> class OsclTimer< Alloc >

### 6.233.1 Member Typedef Documentation

6.233.1.1 template<class Alloc> typedef CallbackTimer<Alloc> OsclTimer< Alloc >::callback\_timer\_type

### 6.233.2 Constructor & Destructor Documentation

6.233.2.1 template<class Alloc> OsclTimer< Alloc >::OsclTimer (const char \* *name*, uint32 *frequency* = 1, int32 *priority* = OsclActiveObject::EPriorityNominal)

Constructor

**Parameters:**

*frequency* The frequency of the timer in cycles/second. A value of 1 means the timer will cycle in 1 second intervals.

6.233.2.2 template<class Alloc> OsclTimer< Alloc >::~OsclTimer () [virtual]

### 6.233.3 Member Function Documentation

6.233.3.1 template<class Alloc> void OsclTimer< Alloc >::Cancel (int32 *timerID*, int32 *timeoutInfo* = -1)

Cancel a timer

**Parameters:**

*timerID* used to identify the timer to cancel.

*timeoutInfo* if not set to -1, this value will be used as additional matching criteria to cancel a timer.

6.233.3.2 template<class Alloc> void OsclTimer< Alloc >::Clear ()

Cancel all pending timers.

6.233.3.3 template<class Alloc> void OsclTimer< Alloc >::Request (int32 *timerID*, int32 *timeoutInfo*, int32 *cycles*, OsclTimerObserver \* *obs* = 0, bool *recurring* = 0)

Request a timer

**Parameters:**

*timerID* used to identify the timer for cancellation. This value will be returned as part of the timeout event.

*timeoutInfo* for user info. Returned to the observer on a timeout event

*cycles* the number of cycles to wait before a timeout event. If the timer frequency is 1 and the cycles are set to 2, then the timeout event will occur in 2 seconds.

*obs* a local observer object to be called on a timeout event. This observer overides the global observer if set.

**6.233.3.4 template<class Alloc> void OsclTimer< Alloc >::SetExactFrequency (uint32 *frequency*)**

Set the exact frequency of the timer in microsecond.

**Parameters:**

*frequency* A value of 1 means the timer will cycle in one microsecond intervals, 1000 means millisecond intervals, etc.

**6.233.3.5 template<class Alloc> void OsclTimer< Alloc >::SetFrequency (uint32 *frequency*)**

Set the frequency of the timer in cycles/second.

**Parameters:**

*frequency* A value of 1 means the timer will cycle in one second intervals, 1000 means millisecond intervals, etc.

**6.233.3.6 template<class Alloc> void OsclTimer< Alloc >::SetObserver ([OsclTimerObserver](#) \* *obs*) [inline]**

Set the global observer. Each timer can request a local observer, which if set overrides the global observer.

**Parameters:**

*obs* observer object.

**6.233.3.7 template<class Alloc> void OsclTimer< Alloc >::TimerBaseElapsed () [protected, virtual]**

Implements [CallbackTimerObserver](#).

## 6.233.4 Friends And Related Function Documentation

**6.233.4.1 template<class Alloc> friend class [CallbackTimer](#)< Alloc > [friend]**

The documentation for this class was generated from the following file:

- [oscl\\_timer.h](#)

## 6.234 OsclTimerCompare Class Reference

```
#include <oscl_scheduler_readyq.h>
```

### Static Public Methods

- int [compare \(TOsclReady &a, TOsclReady &b\)](#)

#### 6.234.1 Member Function Documentation

##### 6.234.1.1 int OsclTimerCompare::compare (TOsclReady &*a*, TOsclReady &*b*) [static]

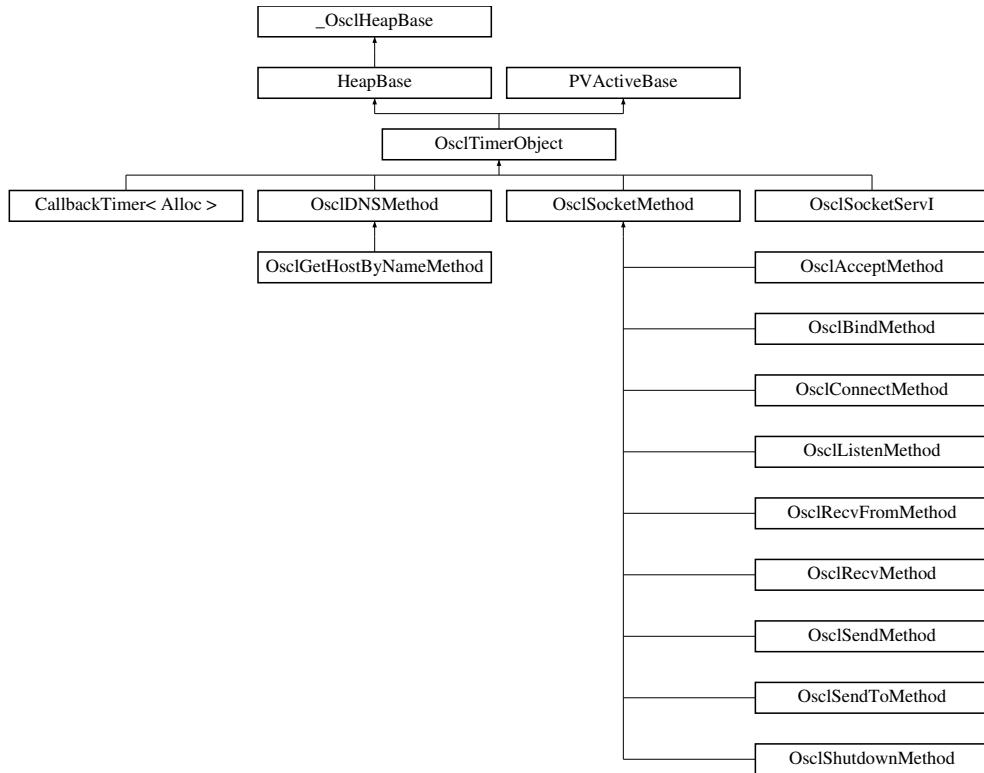
The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_readyq.h](#)

## 6.235 OsclTimerObject Class Reference

```
#include <oscl_scheduler_ao.h>
```

Inheritance diagram for OsclTimerObject::



### Public Methods

- OSCL\_IMPORT\_REF [OsclTimerObject](#) (int32 aPriority, const char name[ ])
- virtual OSCL\_IMPORT\_REF [~OsclTimerObject](#) ()
- OSCL\_IMPORT\_REF void [AddToScheduler](#) ()
- OSCL\_IMPORT\_REF void [RemoveFromScheduler](#) ()
- OSCL\_IMPORT\_REF void [After](#) (int32 aDelayMicrosec)
- OSCL\_IMPORT\_REF void [RunIfNotReady](#) (uint32 aDelayMicrosec=0)
- OSCL\_IMPORT\_REF void [SetBusy](#) ()
- OSCL\_IMPORT\_REF bool [IsBusy](#) () const
- OSCL\_IMPORT\_REF void [Cancel](#) ()
- OSCL\_IMPORT\_REF int32 [Priority](#) () const
- OSCL\_IMPORT\_REF int32 [Status](#) () const
- OSCL\_IMPORT\_REF void [SetStatus](#) (int32)
- OSCL\_IMPORT\_REF [OsclAOStatus](#) & [StatusRef](#) ()

### Protected Methods

- virtual OSCL\_IMPORT\_REF void [DoCancel](#) ()
- virtual OSCL\_IMPORT\_REF int32 [RunError](#) (int32 aError)

### 6.235.1 Detailed Description

User base class for execution objects. OsclTimerObject defines an exec object with a timer.

### 6.235.2 Constructor & Destructor Documentation

#### 6.235.2.1 OSCL\_IMPORT\_REF OsclTimerObject::OsclTimerObject (int32 *aPriority*, const char *name*[ ])

Constructor.

**Parameters:**

*aPriority* (input param): scheduling priority

*name* (input param): optional name for this AO.

#### 6.235.2.2 virtual OSCL\_IMPORT\_REF OsclTimerObject::~OsclTimerObject () [virtual]

Destructor.

### 6.235.3 Member Function Documentation

#### 6.235.3.1 OSCL\_IMPORT\_REF void OsclTimerObject::AddToScheduler ()

Add this AO to the current thread's scheduler.

Reimplemented from [PVActiveBase](#).

#### 6.235.3.2 OSCL\_IMPORT\_REF void OsclTimerObject::After (int32 *aDelayMicrosec*)

'After' sets the request ready, with request status OSCL\_REQUEST\_STATUS\_PENDING, and starts a timer. When the timer expires, the request will complete with status OSCL\_REQUEST\_ERR\_NONE. Must be called from the same thread in which the active object is scheduled. Will leave if the request is already readied, the object is not added to any scheduler, or the calling thread does not match the scheduling thread.

**Parameters:**

*anInterval*: timeout interval in microseconds.

#### 6.235.3.3 OSCL\_IMPORT\_REF void OsclTimerObject::Cancel ()

Cancel any active request. If the request is pending, this will call the DoCancel routine, wait for the request to cancel, then set the request idle. The AO will not run. If the request is not active, it does nothing. Request must be canceled from the same thread in which it is scheduled.

Reimplemented from [PVActiveBase](#).

**6.235.3.4 virtual OSCL\_IMPORT\_REF void OsclTimerObject::DoCancel () [protected, virtual]**

Cancel request handler. This gets called by scheduler when the request is cancelled. The default routine will cancel the timer. If any additional action is needed, the derived class may override this. If the derived class does override this, it should explicitly call [OsclTimerObject::DoCancel](#) in its own DoCancel routine.

Implements [PVActiveBase](#).

**6.235.3.5 OSCL\_IMPORT\_REF bool OsclTimerObject::IsBusy ()**

Return true if this AO is active, false otherwise.

**6.235.3.6 OSCL\_IMPORT\_REF int32 OsclTimerObject::Priority ()**

Return scheduling priority of this exec object.

**6.235.3.7 OSCL\_IMPORT\_REF void OsclTimerObject::RemoveFromScheduler ()**

Remove this AO from its scheduler. Will leave if the calling thread context does not match the scheduling thread. Cancels any pending request before removing.

Reimplemented from [PVActiveBase](#).

**6.235.3.8 virtual OSCL\_IMPORT\_REF int32 OsclTimerObject::RunError (int32 *aError*) [protected, virtual]**

Run Leave handler. This gets called by scheduler when the Run routine leaves. The default implementation simply returns the leave code. If the derived class wants to handle errors from Run, it may override this. The ExecError should return OsclErrNone if it handles the error, otherwise it should return the input error code.

**Parameters:**

*aError*: the leave code generated by the Run.

Implements [PVActiveBase](#).

**6.235.3.9 OSCL\_IMPORT\_REF void OsclTimerObject::RunIfNotReady (uint32 *aDelayMicrosec* = 0)**

Complete the request after a time interval. RunIfNotReady is identical to [After\(\)](#) except that it first checks the request status, and if it is already readied, it does nothing.

**Parameters:**

*aDelayMicrosec* (input param): delay in microseconds.

**6.235.3.10 OSCL\_IMPORT\_REF void OsclTimerObject::SetBusy ()**

Set request ready for this AO. Will leave if the request is already readied, or the exec object is not added to any scheduler, or the calling thread context does not match the scheduler thread.

**6.235.3.11 OSCL\_IMPORT\_REF void OsclTimerObject::SetStatus (int32)**

**6.235.3.12 OSCL\_IMPORT\_REF int32 OsclTimerObject::Status ()**

Request status access

**6.235.3.13 OSCL\_IMPORT\_REF OsclAOStatus& OsclTimerObject::StatusRef ()**

The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_ao.h](#)

## 6.236 OsclTimerObserver Class Reference

```
#include <oscl_timer.h>
```

### Public Methods

- virtual void [TimeoutOccurred](#) (int32 timerID, int32 timeoutInfo)=0
- virtual [~OsclTimerObserver](#) ()

#### 6.236.1 Detailed Description

The observer class to receive timeout callbacks

#### 6.236.2 Constructor & Destructor Documentation

**6.236.2.1** virtual OsclTimerObserver::[~OsclTimerObserver](#) () [inline, virtual]

#### 6.236.3 Member Function Documentation

**6.236.3.1** virtual void OsclTimerObserver::[TimeoutOccurred](#) (int32 *timerID*, int32 *timeoutInfo*) [pure virtual]

This function will be called when the timer associated with this observer is executed

##### Parameters:

*timerID* The ID given at timer request.

*timeoutInfo* Any extra info given at timer request.

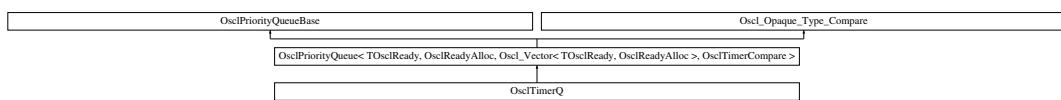
The documentation for this class was generated from the following file:

- [oscl\\_timer.h](#)

## 6.237 OsclTimerQ Class Reference

```
#include <oscl_scheduler_readyq.h>
```

Inheritance diagram for OsclTimerQ::



### Public Methods

- void [Construct](#) (int)
- void [Add](#) ([TOsclReady](#))
- void [Remove](#) ([TOsclReady](#))
- [TOsclReady PopTop](#) ()
- [TOsclReady Top](#) ()
- void [Pop](#) ([TOsclReady](#))
- bool [IsIn](#) ([TOsclReady](#))

#### 6.237.1 Member Function Documentation

**6.237.1.1 void OsclTimerQ::Add ([TOsclReady](#))**

**6.237.1.2 void OsclTimerQ::Construct (int)**

**6.237.1.3 bool OsclTimerQ::IsIn ([TOsclReady](#))**

**6.237.1.4 void OsclTimerQ::Pop ([TOsclReady](#))**

**6.237.1.5 [TOsclReady](#) OsclTimerQ::PopTop ()**

**6.237.1.6 void OsclTimerQ::Remove ([TOsclReady](#))**

**6.237.1.7 [TOsclReady](#) OsclTimerQ::Top ()**

The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_readyq.h](#)

## 6.238 OsclTLS< T, ID, Registry > Class Template Reference

```
#include <oscl_tls.h>
```

### Public Methods

- `OsclTLS ()`
- `~OsclTLS ()`
- `T & operator * () const`

*The indirection operator (\*) accesses a value indirectly, through a pointer.*

- `T * operator -> () const`

*The indirection operator (->) accesses a value indirectly, through a pointer.*

- `bool set ()`

*set() method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.*

### Protected Attributes

- `T * _Ptr`

```
template<class T, uint32 ID, class Registry = OsclTLSRegistry> class OsclTLS< T, ID, Registry >
```

#### 6.238.1 Constructor & Destructor Documentation

**6.238.1.1 template<class T, uint32 ID, class Registry = OsclTLSRegistry> OsclTLS< T, ID, Registry >::OsclTLS () [inline]**

**6.238.1.2 template<class T, uint32 ID, class Registry = OsclTLSRegistry> OsclTLS< T, ID, Registry >::~OsclTLS () [inline]**

#### 6.238.2 Member Function Documentation

**6.238.2.1 template<class T, uint32 ID, class Registry = OsclTLSRegistry> T& OsclTLS< T, ID, Registry >::operator \* () const [inline]**

The indirection operator (\*) accesses a value indirectly, through a pointer.

This operator ensures that the OsclTLS can be used like the regular pointer that it was initialized with.

**6.238.2.2 template<class T, uint32 ID, class Registry = OsclTLSRegistry> T\* OsclTLS< T, ID, Registry >::operator -> () const [inline]**

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the OsclTLS can be used like the regular pointer that it was initialized with.

**6.238.2.3 template<class T, uint32 ID, class Registry = OsclTLSRegistry> bool OsclTLS< T, ID, Registry >::set () [inline]**

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

**6.238.3 Field Documentation****6.238.3.1 template<class T, uint32 ID, class Registry = OsclTLSRegistry> T\* OsclTLS< T, ID, Registry >::\_Ptr [protected]**

The documentation for this class was generated from the following file:

- [oscl\\_tls.h](#)

## 6.239 OsclTLSEEx< T, ID, Registry > Class Template Reference

```
#include <oscl_error.h>
```

### Public Methods

- `OsclTLSEEx ()`
- `~OsclTLSEEx ()`
- `T & operator * () const`

*The indirection operator (\*) accesses a value indirectly, through a pointer.*

- `T * operator -> () const`

*The indirection operator (->) accesses a value indirectly, through a pointer.*

- `bool set ()`

*set() method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.*

### Protected Attributes

- `T * _Ptr`

```
template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> class OsclTLSEEx< T, ID, Registry >
```

#### 6.239.1 Constructor & Destructor Documentation

**6.239.1.1 template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> OsclTLSEEx< T, ID, Registry >::OsclTLSEEx () [inline]**

**6.239.1.2 template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> OsclTLSEEx< T, ID, Registry >::~OsclTLSEEx () [inline]**

#### 6.239.2 Member Function Documentation

**6.239.2.1 template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> T& OsclTLSEEx< T, ID, Registry >::operator \* () const [inline]**

The indirection operator (\*) accesses a value indirectly, through a pointer.

This operator ensures that the `OsclTLS` can be used like the regular pointer that it was initialized with.

**6.239.2.2 template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> T\* OsclTLSEEx< T, ID, Registry >::operator -> () const [inline]**

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the `OsclTLS` can be used like the regular pointer that it was initialized with.

### 6.239.2.3 template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> bool OsclTLSEEx< T, ID, Registry >::set () [inline]

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

## 6.239.3 Field Documentation

### 6.239.3.1 template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> T\* OsclTLSEEx< T, ID, Registry >::\_Ptr [protected]

The documentation for this class was generated from the following file:

- [oscl\\_error.h](#)

## 6.240 OsclTLSRegistry Class Reference

```
#include <oscl_tls.h>
```

### Static Public Methods

- OSCL\_IMPORT\_REF [OsclAny](#) \* getInstance (uint32 ID, int32 &error)
- OSCL\_IMPORT\_REF void [registerInstance](#) ([OsclAny](#) \*ptr, uint32 ID, int32 &error)

### Friends

- class [OsclBase](#)

#### 6.240.1 Member Function Documentation

**6.240.1.1 OSCL\_IMPORT\_REF [OsclAny](#)\* OsclTLSRegistry::getInstance (uint32 *ID*, int32 & *error*) [static]**

**6.240.1.2 OSCL\_IMPORT\_REF void OsclTLSRegistry::registerInstance ([OsclAny](#) \* *ptr*, uint32 *ID*, int32 & *error*) [static]**

#### 6.240.2 Friends And Related Function Documentation

**6.240.2.1 friend class [OsclBase](#) [friend]**

The documentation for this class was generated from the following file:

- [oscl\\_tls.h](#)

## 6.241 OsclTLSRegistryEx Class Reference

```
#include <oscl_error.h>
```

### Static Public Methods

- [OsclAny \\* getInstance \(uint32 ID\)](#)
- [void registerInstance \(OsclAny \\*ptr, uint32 ID\)](#)

#### 6.241.1 Member Function Documentation

**6.241.1.1 [OsclAny\\* OsclTLSRegistryEx::getInstance \(uint32 ID\)](#) [inline, static]**

**6.241.1.2 [void OsclTLSRegistryEx::registerInstance \(OsclAny \\*ptr, uint32 ID\)](#) [inline, static]**

The documentation for this class was generated from the following file:

- [oscl\\_error.h](#)

## 6.242 OsclTrapItem Class Reference

```
#include <oscl_heapbase.h>
```

### Public Methods

- OSCL\_INLINE [OsclTrapItem \(OsclTrapOperation anOperation\)](#)
- OSCL\_INLINE [OsclTrapItem \(OsclTrapOperation anOperation, OsclAny \\*aPtr\)](#)

### Friends

- class [OsclTrapStackItem](#)
- class [OsclTrapStack](#)

#### 6.242.1 Constructor & Destructor Documentation

6.242.1.1 OSCL\_INLINE OsclTrapItem::OsclTrapItem ([OsclTrapOperation anOperation](#))

6.242.1.2 OSCL\_INLINE OsclTrapItem::OsclTrapItem ([OsclTrapOperation anOperation,](#)  
[OsclAny \\* aPtr](#))

#### 6.242.2 Friends And Related Function Documentation

6.242.2.1 friend class [OsclTrapStack](#) [friend]

6.242.2.2 friend class [OsclTrapStackItem](#) [friend]

The documentation for this class was generated from the following file:

- [oscl\\_heapbase.h](#)

## 6.243 OsclTrapStack Class Reference

```
#include <oscl_error_trapcleanup.h>
```

### Friends

- class [OsclError](#)
- class [OsclErrorTrap](#)
- class [OsclErrorTrapImp](#)

### 6.243.1 Detailed Description

A common type for cleanup stack and trap mark stack. for internal use only.

### 6.243.2 Friends And Related Function Documentation

**6.243.2.1 friend class OsclError [friend]**

**6.243.2.2 friend class OsclErrorTrap [friend]**

**6.243.2.3 friend class OsclErrorTrapImp [friend]**

The documentation for this class was generated from the following file:

- [oscl\\_error\\_trapcleanup.h](#)

## 6.244 OsclTrapStackItem Class Reference

```
#include <oscl_error_trapcleanup.h>
```

### Public Methods

- [OsclTrapStackItem \(\)](#)
- [OsclTrapStackItem \(\\_OsclHeapBase \\*aCBase\)](#)
- [OsclTrapStackItem \(OsclAny \\*aTAny\)](#)
- [OsclTrapStackItem \(OsclTrapItem aItem\)](#)

### Data Fields

- [\\_OsclHeapBase \\* iCBase](#)
- [OsclAny \\* iTAny](#)
- [OsclTrapOperation iTrapOperation](#)
- [OsclTrapStackItem \\* iNext](#)

#### 6.244.1 Detailed Description

Internal cleanup stack item type.

#### 6.244.2 Constructor & Destructor Documentation

**6.244.2.1 OsclTrapStackItem::OsclTrapStackItem () [inline]**

**6.244.2.2 OsclTrapStackItem::OsclTrapStackItem (\_OsclHeapBase \* aCBase) [inline]**

**6.244.2.3 OsclTrapStackItem::OsclTrapStackItem (OsclAny \* aTAny) [inline]**

**6.244.2.4 OsclTrapStackItem::OsclTrapStackItem (OsclTrapItem aItem) [inline]**

#### 6.244.3 Field Documentation

**6.244.3.1 \_OsclHeapBase\* OsclTrapStackItem::iCBase**

**6.244.3.2 OsclTrapStackItem\* OsclTrapStackItem::iNext**

**6.244.3.3 OsclAny\* OsclTrapStackItem::iTAny**

**6.244.3.4 OsclTrapOperation OsclTrapStackItem::iTrapOperation**

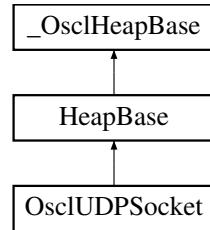
The documentation for this class was generated from the following file:

- [oscl\\_error\\_trapcleanup.h](#)

## 6.245 OsclUDPSocket Class Reference

```
#include <oscl_socket.h>
```

Inheritance diagram for OsclUDPSocket::



### Public Methods

- OSCL\_IMPORT\_REF ~OsclUDPSocket ()
- OSCL\_IMPORT\_REF int32 Close ()
- OSCL\_IMPORT\_REF int32 Bind (OsclNetworkAddress &aAddress)
- OSCL\_IMPORT\_REF int32 Join (OsclNetworkAddress &aAddress)
- OSCL\_IMPORT\_REF TPVSocketEvent BindAsync (OsclNetworkAddress &aAddress, int32 aTimeoutMsec=(-1))
- OSCL\_IMPORT\_REF void CancelBind ()
- OSCL\_IMPORT\_REF uint8 \* GetRecvData (int32 \*aLength)
- OSCL\_IMPORT\_REF uint8 \* GetSendData (int32 \*aLength)
- OSCL\_IMPORT\_REF TPVSocketEvent SendTo (const uint8 \*aPtr, uint32 aLen, OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void CancelSendTo ()
- OSCL\_IMPORT\_REF TPVSocketEvent RecvFrom (uint8 \*aPtr, uint32 aMaxLen, OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1, uint32 aMultiRecvLimit=0, Oscl\_Vector< uint32, OsclMemAllocator > \*aPacketLen=NULL, Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator > \*aPacketSource=NULL)
- OSCL\_IMPORT\_REF void CancelRecvFrom ()
- OSCL\_IMPORT\_REF int32 SetRecvBufferSize (uint32 size)

### Static Public Methods

- OSCL\_IMPORT\_REF OsclUDPSocket \* NewL (Oscl\_DefAlloc &alloc, OsclSocketServ &aServ, OsclSocketObserver \*aObserver, uint32 aId)

#### 6.245.1 Detailed Description

The UDP Socket class

#### 6.245.2 Constructor & Destructor Documentation

##### 6.245.2.1 OSCL\_IMPORT\_REF OsclUDPSocket::~OsclUDPSocket ()

Destructor. The object must be deleted using the same allocator used in the NewL call.

### 6.245.3 Member Function Documentation

#### 6.245.3.1 OSCL\_IMPORT\_REF int32 OsclUDPSocket::Bind ([OsclNetworkAddress](#) & *aAddress*)

Bind a UDP socket to an address. This is a synchronous method.

**Parameters:**

*aAddress*: Bind address.

**Returns:**

Returns OsclErrNone for success, or a platform-specific error code.

#### 6.245.3.2 OSCL\_IMPORT\_REF [TPVSocketEvent](#) OsclUDPSocket::BindAsync ([OsclNetworkAddress](#) & *aAddress*, int32 *aTimeoutMsec* = (-1))

Bind a UDP socket to an address. This is an asynchronous method.

**Parameters:**

*aAddress*: Bind address.

*aTimeoutMsec*: Optional timeout. Use a negative value for infinite wait.

**Returns:**

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

#### 6.245.3.3 OSCL\_IMPORT\_REF void OsclUDPSocket::CancelBind ()

Cancel Bind

This method will cancel any pending BindAsync operation on the current socket, causing the BindAsync to complete with error EPVSocketCancel. If there is no pending BindAsync operation, this method will have no effect.

#### 6.245.3.4 OSCL\_IMPORT\_REF void OsclUDPSocket::CancelRecvFrom ()

Cancel RecvFrom

This method will cancel any pending RecvFrom operation on the current socket, causing the RecvFrom to complete with error EPVSocketCancel. If there is no pending RecvFrom operation, this method will have no effect.

#### 6.245.3.5 OSCL\_IMPORT\_REF void OsclUDPSocket::CancelSendTo ()

Cancel SendTo

This method will cancel any pending SendTo operation on the current socket, causing the SendTo to complete with error EPVSocketCancel. If there is no pending SendTo operation, this method will have no effect.

**6.245.3.6 OSCL\_IMPORT\_REF int32 OsclUDPSocket::Close ()**

Close a UDP socket. This is a synchronous method.

Once it is closed a socket cannot be re-opened. Sockets are automatically closed when they are deleted. This method may be used to see any error code returned from the platform's socket close call.

**Returns:**

Returns OsclErrNone for success, or a platform-specific error code.

**6.245.3.7 OSCL\_IMPORT\_REF uint8\* OsclUDPSocket::GetRecvData (int32 \* aLength)**

Retrieve the received data after a successful RecvFrom operation. This is a synchronous method.

**Parameters:**

*aLength*: (output) number of bytes of data received.

**Returns:**

Returns pointer to received data, or NULL if none.

**6.245.3.8 OSCL\_IMPORT\_REF uint8\* OsclUDPSocket::GetSendData (int32 \* aLength)**

Retrieve the sent data after a successful SendTo operation. This is a synchronous method.

**Parameters:**

*aLength*: (output) number of bytes of data sent.

**Returns:**

Returns pointer to sent data, or NULL if none.

**6.245.3.9 OSCL\_IMPORT\_REF int32 OsclUDPSocket::Join ([OsclNetworkAddress](#) & aAddress)**

Bind a UDP socket to an address and Join the multicast group. This is a synchronous method.

**Parameters:**

*aAddress*: Bind address.

**Returns:**

Returns OsclErrNone for success, or a platform-specific error code. May throw an OsclErrNotSupported Exception

**6.245.3.10 OSCL\_IMPORT\_REF OsclUDPSocket\* OsclUDPSocket::NewL ([Oscl\\_DefAlloc](#) & alloc, [OsclSocketServ](#) & aServ, [OsclSocketObserver](#) \* aObserver, uint32 aId) [static]**

Create a UDP Socket. May leave if failure.

**Parameters:**

*alloc*: Memory allocator.

**aServ:** Socket server. Must be connected.

**aObserver:** Socket observer.

**aId:** Socket ID. The caller must assign an ID to each socket. The ID is used to identify the socket in observer callbacks.

**Returns:**

Returns pointer to socket.

**6.245.3.11 OSCL\_IMPORT\_REF TPVSocketEvent OsclUDPSocket::RecvFrom (uint8 \* aPtr, uint32 aMaxLen, OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1, uint32 aMultiRecvLimit = 0, Oscl\_Vector< uint32, OsclMemAllocator > \* aPacketLen = NULL, Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator > \* aPacketSource = NULL)**

Receive Data. This is an asynchronous method.

**Parameters:**

**aPtr:** Buffer to receive incoming data

**aMaxLen:** Length of buffer.

**aAddress:** (output) Source address.

**aTimeoutMsec:** Timeout in milliseconds, or (-1) for infinite wait.

**aMultiRecvLimit** (optional input): Configures multiple packet receive mode. As long as there are packets queued at the socket and at least aMultiRecvLimit bytes are available in the buffer, recvfrom operations will continue. A value of zero disabled multiple packet mode. The individual packet lengths can be retrieved in the aPacketLen parameter; and the individual packet source addresses can be retrieved in the aPacketSource parameter.

**aPacketLen:** (optional output) a vector of packet lengths, in case multiple packets were received.

**aPacketSource:** (optional output) a vector of source addresses, in case multiple packets were received.

**Returns:**

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

**6.245.3.12 OSCL\_IMPORT\_REF TPVSocketEvent OsclUDPSocket::SendTo (const uint8 \* aPtr, uint32 aLen, OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1)**

Send Data. This is an asynchronous method.

**Parameters:**

**aPtr:** Data to send.

**aLen:** Length of data to send.

**aAddress:** Destination address.

**aTimeoutMsec:** Timeout in milliseconds, or (-1) for infinite wait.

**Returns:**

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

**6.245.3.13 OSCL\_IMPORT\_REF int32 OsclUDPSocket::SetRecvBufferSize (uint32 *size*)**

Set the buffer size of the socket This is a synchronous method.

**Parameters:**

*size*: buffer size

**Returns:**

Returns OsclErrNone for success, or a platform-specific error code. May throw an OsclErrNotSupported Exception.

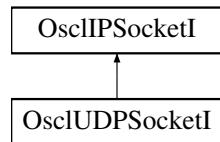
The documentation for this class was generated from the following file:

- [oscl\\_socket.h](#)

## 6.246 OsclUDPSocketI Class Reference

```
#include <oscl_udp_socket.h>
```

Inheritance diagram for OsclUDPSocketI::



### Public Methods

- virtual ~OsclUDPSocketI ()
- int32 [Close \(\)](#)
- uint8 \* [GetRecvData \(int32 \\*aLength\)](#)
- uint8 \* [GetSendData \(int32 \\*aLength\)](#)
- [TPVSocketEvent BindAsync \(OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1\)](#)
- void [CancelBind \(\)](#)
- [TPVSocketEvent SendTo \(const uint8 \\*&aPtr, uint32 aLen, OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1\)](#)
- void [CancelSendTo \(\)](#)
- [TPVSocketEvent RecvFrom \(uint8 \\*&aPtr, uint32 aMaxLen, OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1, uint32 aMultiMaxLen=0, Oscl\\_Vector< uint32, OsclMemAllocator > \\*aPacketLen=NULL, Oscl\\_Vector< OsclNetworkAddress, OsclMemAllocator > \\*aPacketSource=NULL\)](#)
- void [CancelRecvFrom \(\)](#)

### Static Public Methods

- OsclUDPSocketI \* [NewL \(Oscl\\_DefAlloc &a, OsclSocketServI \\*aServ, OsclSocketObserver \\*aObserver, uint32 aId\)](#)

#### 6.246.1 Detailed Description

Internal implementation class for [OsclUDPSocket](#)

## 6.246.2 Constructor & Destructor Documentation

6.246.2.1 `virtual OsclUDPSocketI::~OsclUDPSocketI () [virtual]`

## 6.246.3 Member Function Documentation

6.246.3.1 `TPVSocketEvent OsclUDPSocketI::BindAsync (OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1) [inline]`

6.246.3.2 `void OsclUDPSocketI::CancelBind () [inline]`

6.246.3.3 `void OsclUDPSocketI::CancelRecvFrom () [inline]`

6.246.3.4 `void OsclUDPSocketI::CancelSendTo () [inline]`

6.246.3.5 `int32 OsclUDPSocketI::Close () [virtual]`

Implements [OsclIPSocketI](#).

6.246.3.6 `uint8 * OsclUDPSocketI::GetRecvData (int32 * aLength) [inline, virtual]`

Implements [OsclIPSocketI](#).

6.246.3.7 `uint8 * OsclUDPSocketI::GetSendData (int32 * aLength) [inline, virtual]`

Implements [OsclIPSocketI](#).

6.246.3.8 `OsclUDPSocketI* OsclUDPSocketI::NewL (Oscl_DefAlloc & a, OsclSocketServI * aServ, OsclSocketObserver * aObserver, uint32 aId) [static]`

6.246.3.9 `TPVSocketEvent OsclUDPSocketI::RecvFrom (uint8 *& aPtr, uint32 aMaxLen, OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1, uint32 aMultiMaxLen = 0, Oscl_Vector< uint32, OsclMemAllocator > * aPacketLen = NULL, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > * aPacketSource = NULL) [inline]`

6.246.3.10 `TPVSocketEvent OsclUDPSocketI::SendTo (const uint8 *& aPtr, uint32 aLen, OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1) [inline]`

The documentation for this class was generated from the following file:

- [oscl\\_udp\\_socket.h](#)

## 6.247 OsclUuid Struct Reference

```
#include <oscl_uuid.h>
```

### Public Methods

- [OsclUuid \(\)](#)
- [OsclUuid \(uint32 l, uint16 w1, uint16 w2, uint8 b1, uint8 b2, uint8 b3, uint8 b4, uint8 b5, uint8 b6, uint8 b7, uint8 b8\)](#)
- [OsclUuid \(const char \\*aUuidString\)](#)
- [OsclUuid \(const OsclUuid &uuid\)](#)
- [OsclUuid & operator= \(const OsclUuid &src\)](#)
- [bool operator== \(const OsclUuid &src\) const](#)
- [bool operator!= \(const OsclUuid &src\) const](#)

### Data Fields

- [uint32 data1](#)
- [uint16 data2](#)
- [uint16 data3](#)
- [uint8 data4 \[BYTES\\_IN\\_UUID\\_ARRAY\]](#)

#### 6.247.1 Detailed Description

OSCL UUID structure used for unique identification of modules and interfaces.

## 6.247.2 Constructor & Destructor Documentation

**6.247.2.1 OsclUuid::OsclUuid () [inline]**

**6.247.2.2 OsclUuid::OsclUuid (uint32 *l*, uint16 *w1*, uint16 *w2*, uint8 *b1*, uint8 *b2*, uint8 *b3*, uint8 *b4*, uint8 *b5*, uint8 *b6*, uint8 *b7*, uint8 *b8*) [inline]**

**6.247.2.3 OsclUuid::OsclUuid (const char \* *aUuidString*) [inline]**

**6.247.2.4 OsclUuid::OsclUuid (const OsclUuid & *uuid*) [inline]**

## 6.247.3 Member Function Documentation

**6.247.3.1 bool OsclUuid::operator!= (const OsclUuid & *src*) const [inline]**

**6.247.3.2 OsclUuid& OsclUuid::operator= (const OsclUuid & *src*) [inline]**

**6.247.3.3 bool OsclUuid::operator== (const OsclUuid & *src*) const [inline]**

## 6.247.4 Field Documentation

**6.247.4.1 uint32 OsclUuid::data1**

**6.247.4.2 uint16 OsclUuid::data2**

**6.247.4.3 uint16 OsclUuid::data3**

**6.247.4.4 uint8 OsclUuid::data4[BYTES\_IN\_UUID\_ARRAY]**

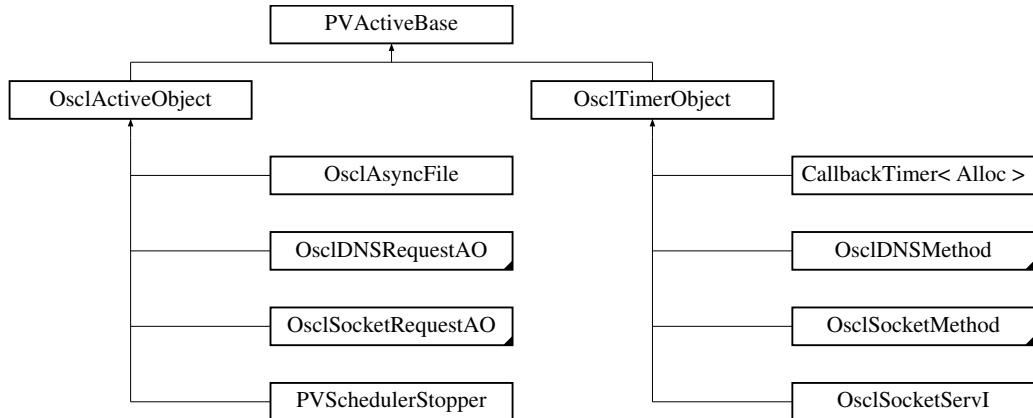
The documentation for this struct was generated from the following file:

- [oscl\\_uuid.h](#)

## 6.248 PVActiveBase Class Reference

```
#include <oscl_scheduler_aobase.h>
```

Inheritance diagram for PVActiveBase::



### Public Methods

- [PVActiveBase](#) (const char name[ ], int32 pri)
- virtual [~PVActiveBase](#) ()
- bool [IsInAnyQ](#) ()
- virtual int32 [RunError](#) (int32 aError)=0
- virtual void [Run](#) ()=0
- virtual void [DoCancel](#) ()=0
- void [AddToScheduler](#) ()
- void [RemoveFromScheduler](#) ()
- void [Destroy](#) ()
- void [Activate](#) ()
- OSCL\_IMPORT\_REF bool [IsAdded](#) () const
- void [Cancel](#) ()

### Data Fields

- uint32 [iAddedNum](#)
- [OsclNameString< PVEXECNAMELEN > iName](#)
- [PVThreadContext iThreadContext](#)
- [PVActiveStats \\* iPVAstats](#)
- [TReadyQueLink iPVReadyQLink](#)
- bool [iBusy](#)
- [OsclAOStatus iStatus](#)

### Friends

- class [PVActiveStats](#)
- class [OsclSchedulerCommonBase](#)

- class [OsclActiveObject](#)
- class [OsclTimerObject](#)
- class [OsclReadyQ](#)
- class [OsclReadyCompare](#)
- class [OsclReadySetPosition](#)
- class [OsclExecScheduler](#)

## 6.248.1 Detailed Description

PV Scheduler internal AO base class. Both [OsclActiveObject](#) and [OsclTimerObject](#) derive from this class. For Symbian, this just container has the desired additions to the basic CTimer or OsclActiveObj functionality. For non-Symbian, this class contains the entire AO implementation.

## 6.248.2 Constructor & Destructor Documentation

**6.248.2.1 PVActiveBase::PVActiveBase (const char *name*[ ], int32 *pri*)**

**6.248.2.2 virtual PVActiveBase::~PVActiveBase () [virtual]**

## 6.248.3 Member Function Documentation

**6.248.3.1 void PVActiveBase::Activate ()**

**6.248.3.2 void PVActiveBase::AddToScheduler ()**

Reimplemented in [OsclActiveObject](#), and [OsclTimerObject](#).

**6.248.3.3 void PVActiveBase::Cancel ()**

Reimplemented in [OsclActiveObject](#), and [OsclTimerObject](#).

**6.248.3.4 void PVActiveBase::Destroy ()**

**6.248.3.5 virtual void PVActiveBase::DoCancel () [pure virtual]**

Implements cancellation of an outstanding request.

This function is called as part of the active object's [Cancel\(\)](#).

It must call the appropriate cancel function offered by the active object's asynchronous service provider. The asynchronous service provider's cancel is expected to act immediately.

[DoCancel\(\)](#) must not wait for event completion; this is handled by [Cancel\(\)](#).

Implemented in [OsclDNSRequestAO](#), [OsclSocketRequestAO](#), [OsclActiveObject](#), and [OsclTimerObject](#).

**6.248.3.6 OSCL\_IMPORT\_REF bool PVActiveBase::IsAdded ()**

**6.248.3.7 bool PVActiveBase::IsInAnyQ () [inline]**

**6.248.3.8 void PVActiveBase::RemoveFromScheduler ()**

Reimplemented in [OsclActiveObject](#), and [OsclTimerObject](#).

**6.248.3.9 virtual void PVActiveBase::Run () [pure virtual]**

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's `WaitForAnyRequest()` function completes.

Before calling this active object's `Run()` function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request
2. marked this active object's request as complete (i.e. the request is no longer outstanding)

`Run()` runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls `ExecError()` to handle the leave.

Note that once the active scheduler's `Start()` function has been called, all user code is run under one of the program's active object's `Run()` or `RunError()` functions.

Implemented in [OsclDNSMethod](#), [OsclDNSRequestAO](#), [OsclSocketMethod](#), [OsclSocketRequestAO](#), and [CallbackTimer< Alloc >](#).

**6.248.3.10 virtual int32 PVActiveBase::RunError (int32 *aError*) [pure virtual]**

Virtual routine that gets called if the active object's Run routine leaves.

**Parameters:**

***aError*:** the leave code generated by the Run.

**Returns:**

:returns `OsclErrNone` if the error was handled, or returns the input `aError` value if not handled.

Implemented in [OsclActiveObject](#), and [OsclTimerObject](#).

## 6.248.4 Friends And Related Function Documentation

**6.248.4.1 friend class OsclActiveObject [friend]**

**6.248.4.2 friend class OsclExecScheduler [friend]**

**6.248.4.3 friend class OsclReadyCompare [friend]**

**6.248.4.4 friend class OsclReadyQ [friend]**

**6.248.4.5 friend class OsclReadySetPosition [friend]**

**6.248.4.6 friend class OsclSchedulerCommonBase [friend]**

**6.248.4.7 friend class OsclTimerObject [friend]**

**6.248.4.8 friend class PVActiveStats [friend]**

## 6.248.5 Field Documentation

**6.248.5.1 uint32 PVActiveBase::iAddedNum**

**6.248.5.2 bool PVActiveBase::iBusy**

**6.248.5.3 OsclNameString<PVEXECNAMELEN> PVActiveBase::iName**

**6.248.5.4 PVActiveStats\* PVActiveBase::iPVActiveStats**

**6.248.5.5 TReadyQueLink PVActiveBase::iPVReadyQLink**

**6.248.5.6 OsclAOStatus PVActiveBase::iStatus**

The request status associated with an asynchronous request.

This is passed as a parameter to all asynchronous service providers.

The active scheduler uses this to check whether the active object's request has completed.

The function can use the completion code to judge the success or otherwise of the request.

Request status contains one of the values OSCL\_REQUEST\_ERR\_NONE: request completed with no error, or request is not active. OSCL\_REQUEST\_PENDING: request is active & pending OSCL\_REQUEST\_ERR\_CANCEL: request was canceled before completion. or any user-defined value.

**6.248.5.7 PVThreadContext PVActiveBase::iThreadContext**

The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_aobase.h](#)

## 6.249 PVActiveStats Class Reference

```
#include <oscl_scheduler_aobase.h>
```

### Friends

- class [PVActiveBase](#)
- class [OsclExecScheduler](#)
- class [OsclExecSchedulerCommonBase](#)
- class [OsclActiveObject](#)
- class [OsclTimerObject](#)
- class [OsclReadyQ](#)

### 6.249.1 Detailed Description

PV AO statistics

### 6.249.2 Friends And Related Function Documentation

**6.249.2.1 friend class OsclActiveObject [friend]**

**6.249.2.2 friend class OsclExecScheduler [friend]**

**6.249.2.3 friend class OsclExecSchedulerCommonBase [friend]**

**6.249.2.4 friend class OsclReadyQ [friend]**

**6.249.2.5 friend class OsclTimerObject [friend]**

**6.249.2.6 friend class PVActiveBase [friend]**

The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_aobase.h](#)

## 6.250 PVLogger Class Reference

```
#include <pvlogger.h>
```

### Public Types

- `typedef int32 log_level_type`
- `typedef int32 message_id_type`
- `typedef int32 filter_status_type`
- `typedef _OsclBasicAllocator alloc_type`

### Public Methods

- `void SetLogLevel (log_level_type level)`
- `OSCL_IMPORT_REF void SetLogLevelAndPropagate (log_level_type level)`
- `log_level_type GetLogLevel ()`
- `void DisableAppenderInheritance ()`
- `void AddAppender (OsclSharedPtr< PVLoggerAppender > &appender)`
- `void RemoveAppender (OsclSharedPtr< PVLoggerAppender > &appender)`
- `void AddFilter (OsclSharedPtr< PVLoggerFilter > &filter)`
- `uint32 GetNumAppenders ()`
- `OSCL_IMPORT_REF bool IsActive (log_level_type level)`
- `OSCL_IMPORT_REF void LogMsgStringV (message_id_type msgID, const char *fmt, va_list arguments)`
- `OSCL_IMPORT_REF void LogMsgBuffersV (message_id_type msgID, int32 numPairs, va_list arguments)`
- `OSCL_IMPORT_REF void LogMsgString (message_id_type msgID, const char *fmt,...)`
- `OSCL_IMPORT_REF void LogMsgBuffers (message_id_type msgID, int32 numPairs,...)`
- `OSCL_IMPORT_REF PVLogger (const char *inputTag, log_level_type level, bool oAppenderInheritance)`
- `virtual ~PVLogger ()`

### Static Public Methods

- `OSCL_IMPORT_REF void Init ()`
- `OSCL_IMPORT_REF void Cleanup ()`
- `OSCL_IMPORT_REF PVLogger * GetLoggerObject (const char *inputTag)`

### Protected Methods

- `void SetParent (PVLogger *parentLogger)`
- `PVLogger * GetParent ()`

### Friends

- class `PVLoggerRegistry`

## 6.250.1 Member Typedef Documentation

**6.250.1.1** `typedef _OsclBasicAllocator PVLogger::alloc_type`

**6.250.1.2** `typedef int32 PVLogger::filter_status_type`

**6.250.1.3** `typedef int32 PVLogger::log_level_type`

**6.250.1.4** `typedef int32 PVLogger::message_id_type`

## 6.250.2 Constructor & Destructor Documentation

**6.250.2.1** `OSCL_IMPORT_REF PVLogger::PVLogger (const char * inputTag, log_level_type level, bool oAppenderInheritance)`

Logger Constructor

**Parameters:**

*tag* Logger tag, unique to a logging control point

*level* Active Log level of the logger

*oAppenderInheritance*

**Returns:**

NONE

**6.250.2.2** `virtual PVLogger::~PVLogger () [inline, virtual]`

## 6.250.3 Member Function Documentation

**6.250.3.1** `void PVLogger::AddAppender (OsclSharedPtr< PVLoggerAppender > & appender) [inline]`

This method adds an appender to the logging control point. Each logger maintains a list of appenders. Any msg to a logger if deemed active is logged to all the appenders.

**Parameters:**

*appender* pointer to the appender to add

**Returns:**

NONE

**Exceptions:**

*leaves* if out of memory

**6.250.3.2** `void PVLogger::AddFilter (OsclSharedPtr< PVLoggerFilter > & filter) [inline]`

This method adds a message filter to the logging control point. Each logger maintains a list of filters. Any msg to a logger if deemed active is passed through the msg filters prior to logging.

**Parameters:**

*msgFilter* pointer to the filter to add

**Returns:**

NONE

**Exceptions:**

*leaves* if out of memory

**6.250.3.3 OSCL\_IMPORT\_REF void PVLogger::Cleanup () [static]**

Frees the PVLogger singleton used by the current thread. This must be called before thread exit. No messages can be logged after cleanup.

**Returns:****6.250.3.4 void PVLogger::DisableAppenderInheritance () [inline]**

This method disables appender inheritance for the logging control point

**6.250.3.5 OSCL\_IMPORT\_REF PVLogger\* PVLogger::GetLoggerObject (const char \* *inputTag*) [static]**

This is a factory method to create a log control point, with a certain input tag. There is a central registry of all the loggers, with their corresponding tags, called PV Logger Registry. In case the logger with the specified tag exists in the global registry, it is returned, else a new one is created and a pointer to the same is returned.

**Parameters:**

*inputTag* logger tag, viz. "x.y.z"

*level* log level associated with the logging control point (All messages with log levels less than equal to the log level of the control point would be logged)

*oAppenderInheritance*

**Returns:**

PVLogger\* Pointer to the logging control point

**Exceptions:**

*leaves* if out of memory

**6.250.3.6 log\_level\_type PVLogger::GetLogLevel () [inline]**

This method returns the log level of a control point. This could either have been set explicitly by the user (at the time of creation or later) or could have been inherited from one of its ancestors.

**Returns:**

log level associated with the logging control point

**6.250.3.7 uint32 PVLogger::GetNumAppenders () [inline]**

This method returns the number of appenders attached to the logging control point.

**6.250.3.8 PVLogger\* PVLogger::GetParent () [inline, protected]****6.250.3.9 OSCL\_IMPORT\_REF void PVLogger::Init () [static]**

PVLogger needs to be initialized once per thread. This creates the PVLogger singleton that is used throughout the duration of the thread. Initialization must occur before the first message is logged.

**Exceptions:**

*leaves* if out of memory

**6.250.3.10 OSCL\_IMPORT\_REF bool PVLogger::IsActive ([log\\_level\\_type level](#))**

This method determines if a msg passed to the logging control point is active or not. Only messages that are deemed active are logged. Messages are considered not active if any of the following criteria are met:

- All logging is disabled at this logging control point
- If all the log levels, leading upto the root log point are uninitialized
- If the log level of the incoming message is LESS THAN that of the active log level of the logging control point.

**Returns:**

BOOL

**6.250.3.11 OSCL\_IMPORT\_REF void PVLogger::LogMsgBuffers ([message\\_id\\_type msgID](#), int32 *numPairs*, ...)**

This method logs opaque data buffers to all the appenders, after running through the message filters. After logging the message to the appenders attached to the current control point, the message is passed up to the parent node, only if appender inheritance is enabled.

**Parameters:**

*msgID* Message ID, that is unique to a message

*numPairs* Number of (ptr\_len, ptr) pairs

*arguments* Variable list of arguments

**Returns:**

NONE

**6.250.3.12 OSCL\_IMPORT\_REF void PVLogger::LogMsgBuffersV ([message\\_id\\_type msgID](#), int32 *numPairs*, va\_list *arguments*)**

This method logs opaque data buffers to all the appenders, after running through the message filters. After logging the message to the appenders attached to the current control point, the message is passed up to the parent node, only if appender inheritance is enabled.

**Parameters:**

*msgID* Message ID, that is unique to a message

*numPairs* Number of (ptr\_len, ptr) pairs

*arguments* Variable list of arguments

**Returns:**

NONE

**6.250.3.13 OSCL\_IMPORT\_REF void PVLogger::LogMsgString (*message\_id\_type msgID, const char \*fmt, ...)***

This method logs formatted text msg to all the appenders, after running through the message filters. After logging the message to the appenders attached to the current control point, the message is passed up to the parent node, only if appender inheritance is enabled.

**Parameters:**

*msgID* Message ID, that is unique to a message

*fmt* format string, similar to one taken by printf

*arguments* Variable list of arguments

**Returns:**

NONE

**6.250.3.14 OSCL\_IMPORT\_REF void PVLogger::LogMsgStringV (*message\_id\_type msgID, const char \*fmt, va\_list arguments)***

This method logs formatted text msg to all the appenders, after running through the message filters. After logging the message to the appenders attached to the current control point, the message is passed up to the parent node, only if appender inheritance is enabled.

**Parameters:**

*msgID* Message ID, that is unique to a message

*fmt* format string, similar to one taken by printf

*arguments* Variable list of arguments

**Returns:**

NONE

**6.250.3.15 void PVLogger::RemoveAppender (*OsclSharedPtr< PVLoggerAppender > & appender*) [inline]**

This method removes an appender from the logging control point. Each logger maintains a list of appenders. Any msg to a logger if deemed active is logged to all the appenders.

**Parameters:**

*appender* pointer to the appender to delete

**Returns:**

NONE

**6.250.3.16 void PVLogger::SetLogLevel (*log\_level\_type* *level*) [inline]**

This method is used to set the log level of a control point.

**Parameters:**

*level* log level associated with the logging control point

**Returns:**

NONE

**6.250.3.17 OSCL\_IMPORT\_REF void PVLogger::SetLogLevelAndPropagate (*log\_level\_type* *level*)**

This method is used to set the log level of a control point, as well as to propagate the level to all the descendants of this control point.

**Parameters:**

*level* log level associated with the logging control point

**Returns:**

NONE

**6.250.3.18 void PVLogger::SetParent (PVLogger \**parentLogger*) [inline, protected]****6.250.4 Friends And Related Function Documentation****6.250.4.1 friend class PVLoggerRegistry [friend]**

The documentation for this class was generated from the following file:

- [pvlogger.h](#)

## 6.251 PVLoggerAppender Class Reference

```
#include <pvlogger_accessories.h>
```

### Public Types

- `typedef PVLogger::message_id_type message_id_type`

### Public Methods

- `virtual ~PVLoggerAppender ()`
- `virtual void AppendString (message_id_type msgID, const char *fmt, va_list va)=0`
- `virtual void AppendBuffers (message_id_type msgID, int32 numPairs, va_list va)=0`

#### 6.251.1 Detailed Description

Base class for all message appenders. This class defines the interface to the message appenders. There are two kinds of msg appender APIs, one to append text messages, and other to append opaque message buffers.

#### 6.251.2 Member Typedef Documentation

##### 6.251.2.1 `typedef PVLogger::message_id_type PVLoggerAppender::message_id_type`

#### 6.251.3 Constructor & Destructor Documentation

##### 6.251.3.1 `virtual PVLoggerAppender::~PVLoggerAppender () [inline, virtual]`

#### 6.251.4 Member Function Documentation

##### 6.251.4.1 `virtual void PVLoggerAppender::AppendBuffers (message_id_type msgID, int32 numPairs, va_list va) [pure virtual]`

##### 6.251.4.2 `virtual void PVLoggerAppender::AppendString (message_id_type msgID, const char *fmt, va_list va) [pure virtual]`

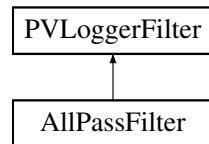
The documentation for this class was generated from the following file:

- `pvlogger_accessories.h`

## 6.252 PVLoggerFilter Class Reference

```
#include <pvlogger_accessories.h>
```

Inheritance diagram for PVLoggerFilter:::



### Public Types

- `typedef PVLogger::message_id_type message_id_type`
- `typedef PVLogger::log_level_type log_level_type`
- `typedef PVLogger::filter_status_type filter_status_type`

### Public Methods

- `virtual ~PVLoggerFilter ()`
- `virtual filter_status_type FilterString (char *tag, message_id_type msgID, log_level_type level)=0`
- `virtual filter_status_type FilterOpaqueMessge (char *tag, message_id_type msgID, log_level_type level)=0`

#### 6.252.1 Detailed Description

Base class for all message filters. This class defines the interface to the message filters. There are two kinds of msg filtering APIs, one to filter text messages, and other to filter opaque message buffers.

#### 6.252.2 Member Typedef Documentation

##### 6.252.2.1 `typedef PVLogger::filter_status_type PVLoggerFilter::filter_status_type`

Reimplemented in [AllPassFilter](#).

##### 6.252.2.2 `typedef PVLogger::log_level_type PVLoggerFilter::log_level_type`

Reimplemented in [AllPassFilter](#).

##### 6.252.2.3 `typedef PVLogger::message_id_type PVLoggerFilter::message_id_type`

Reimplemented in [AllPassFilter](#).

### 6.252.3 Constructor & Destructor Documentation

6.252.3.1 `virtual PVLoggerFilter::~PVLoggerFilter () [inline, virtual]`

### 6.252.4 Member Function Documentation

6.252.4.1 `virtual filter_status_type PVLoggerFilter::FilterOpaqueMessge (char * tag, message_id_type msgID, log_level_type level) [pure virtual]`

Implemented in [AllPassFilter](#).

6.252.4.2 `virtual filter_status_type PVLoggerFilter::FilterString (char * tag, message_id_type msgID, log_level_type level) [pure virtual]`

Implemented in [AllPassFilter](#).

The documentation for this class was generated from the following file:

- [pvlogger\\_accessories.h](#)

## 6.253 PVLoggerLayout Class Reference

```
#include <pvlogger_accessories.h>
```

### Public Types

- `typedef PVLogger::message_id_type message_id_type`

### Public Methods

- `virtual ~PVLoggerLayout ()`
- `virtual int32 FormatString (char *formatBuf, int32 formatBufSize, message_id_type msgID, const char *fmt, va_list va)=0`
- `virtual int32 FormatOpaqueMessage (char *formatBuf, int32 formatBufSize, message_id_type msgID, int32 numPairs, va_list va)=0`

#### 6.253.1 Detailed Description

Base class for all message formatters. This class defines the interface to the message formatter. There are two kinds of msg formatting APIs, one to format text messages, and other to format opaque message buffers.

#### 6.253.2 Member Typedef Documentation

##### 6.253.2.1 `typedef PVLogger::message_id_type PVLoggerLayout::message_id_type`

#### 6.253.3 Constructor & Destructor Documentation

##### 6.253.3.1 `virtual PVLoggerLayout::~PVLoggerLayout () [inline, virtual]`

#### 6.253.4 Member Function Documentation

##### 6.253.4.1 `virtual int32 PVLoggerLayout::FormatOpaqueMessage (char *formatBuf, int32 formatBufSize, message_id_type msgID, int32 numPairs, va_list va) [pure virtual]`

Formats the data and copies it to the given buffer.

#### Returns:

The length of the buffer used.

##### 6.253.4.2 `virtual int32 PVLoggerLayout::FormatString (char *formatBuf, int32 formatBufSize, message_id_type msgID, const char *fmt, va_list va) [pure virtual]`

Formats the string and copies it to the given buffer.

#### Returns:

The length of the string not including the trailing '\0'

The documentation for this class was generated from the following file:

- [pvlogger\\_accessories.h](#)

## 6.254 PVLoggerRegistry Class Reference

```
#include <pvlogger_registry.h>
```

### Public Types

- `typedef PVLogger::log_level_type log_level_type`
- `typedef PVLogger::alloc_type alloc_type`

### Public Methods

- `OSCL_IMPORT_REF PVLoggerRegistry()`
- `virtual OSCL_IMPORT_REF ~PVLoggerRegistry()`
- `OSCL_IMPORT_REF PVLogger * GetPVLoggerObject (const char *tagIn)`
- `OSCL_IMPORT_REF PVLogger * CreatePVLogger (const char *tagIn, log_level_type level, bool oAppenderInheritance)`
- `OSCL_IMPORT_REF bool SetNodeLogLevelExplicit (char *tagIn, log_level_type level)`
- `OSCL_IMPORT_REF void SetNodeLogLevelExplicit (Oscl_TagTree< PVLogger *, alloc_type >::node_type *node, log_level_type level)`

### Static Public Methods

- `OSCL_IMPORT_REF PVLoggerRegistry * GetPVLoggerRegistry()`

#### 6.254.1 Detailed Description

Class: PVLoggerRegistry

PVLoggerRegistry class, maintains a repository of all the loggers, along with their associated tags, in a tag tree. Any request for a log control point is serviced by this class.

Memory Ownership: Creates log control points for each tag, and holds these pointers in the tag tree. `PVLogger` registry is responsible for calling the destructor on each of these loggers.

#### 6.254.2 Member Typedef Documentation

##### 6.254.2.1 `typedef PVLogger::alloc_type PVLoggerRegistry::alloc_type`

##### 6.254.2.2 `typedef PVLogger::log_level_type PVLoggerRegistry::log_level_type`

#### 6.254.3 Constructor & Destructor Documentation

##### 6.254.3.1 `OSCL_IMPORT_REF PVLoggerRegistry::PVLoggerRegistry()`

PVLoggerRegistry Constructor

##### 6.254.3.2 `virtual OSCL_IMPORT_REF PVLoggerRegistry::~PVLoggerRegistry () [virtual]`

PVLoggerRegistry Destructor

## 6.254.4 Member Function Documentation

### 6.254.4.1 OSCL\_IMPORT\_REF PVLogger\* PVLoggerRegistry::CreatePVLogger (const char \* *tagIn*, log\_level\_type *level*, bool *oAppenderInheritance*)

This method creates a log control point, with specified tag, and level

**Parameters:**

*inputTag* logger tag, viz. "x.y.z"  
*level* log level associated with the logging control point  
*oAppenderInheritance*

**Returns:**

PVLogger<alloc\_type, TheLock>\* Pointer to the logging control point

### 6.254.4.2 OSCL\_IMPORT\_REF PVLogger\* PVLoggerRegistry::GetPVLoggerObject (const char \* *tagIn*)

PVLoggerRegistry method to get access to a logging control point, associated with a tag. In case the logger for this tag does not exist, it is created afresh, else pointer to the existing one is returned.

**Parameters:**

*inputTag* logger tag, viz. "x.y.z"  
*level* log level associated with the logging control point  
*oAppenderInheritance*

**Returns:**

PVLogger<Alloc, TheLock>\* Pointer to the logging control point

### 6.254.4.3 OSCL\_IMPORT\_REF PVLoggerRegistry\* PVLoggerRegistry::GetPVLoggerRegistry () [static]

Get the logger registry. There is only one logger registry instance per thread.

### 6.254.4.4 OSCL\_IMPORT\_REF void PVLoggerRegistry::SetNodeLogLevelExplicit (Oscl\_TagTree< PVLogger \*, alloc\_type >::node\_type \* *node*, log\_level\_type *level*)

This method recursively propagates the log level to all the descendants, of a node.

**Parameters:**

*node* Node ptr, associated with a logger, from the tag tree.  
*level* log level associated with the logging control point

**Returns:**

NONE

**6.254.4.5 OSCL\_IMPORT\_REF bool PVLoggerRegistry::SetNodeLogLevelExplicit (char \* *tagIn*,  
*log\_level\_type level*)**

This method propagates the log level to all the descendants of the node, with a specified tag.

**Parameters:**

*tagIn* logger tag, viz. "x.y.z"

*level* log level associated with the logging control point

**Returns:**

true on success, else false.

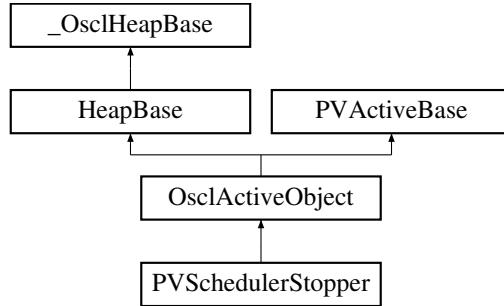
The documentation for this class was generated from the following file:

- [pvlogger\\_registry.h](#)

## 6.255 PVSchedulerStopper Class Reference

```
#include <oscl_scheduler.h>
```

Inheritance diagram for PVSchedulerStopper::



### Public Methods

- [PVSchedulerStopper \(\)](#)
- [~PVSchedulerStopper \(\)](#)

#### 6.255.1 Detailed Description

Scheduler stopper AO class, for internal use by scheduler.

#### 6.255.2 Constructor & Destructor Documentation

##### 6.255.2.1 PVSchedulerStopper::PVSchedulerStopper ()

##### 6.255.2.2 PVSchedulerStopper::~PVSchedulerStopper ()

The documentation for this class was generated from the following file:

- [oscl\\_scheduler.h](#)

## 6.256 PVSockBufRecv Class Reference

```
#include <oscl_socket_request.h>
```

### Public Methods

- [PVSockBufRecv \(\)](#)
- [PVSockBufRecv \(uint8 \\*aPtr, uint32 aLen, uint32 aMax\)](#)
- [PVSockBufRecv \(const PVSockBufRecv &a\)](#)

### Data Fields

- [uint8 \\* iPtr](#)
- [uint32 iLen](#)
- [uint32 iMaxLen](#)

#### 6.256.1 Constructor & Destructor Documentation

**6.256.1.1 PVSockBufRecv::PVSockBufRecv () [inline]**

**6.256.1.2 PVSockBufRecv::PVSockBufRecv (uint8 \* *aPtr*, uint32 *aLen*, uint32 *aMax*) [inline]**

**6.256.1.3 PVSockBufRecv::PVSockBufRecv (const PVSockBufRecv & *a*) [inline]**

#### 6.256.2 Field Documentation

**6.256.2.1 uint32 PVSockBufRecv::iLen**

**6.256.2.2 uint32 PVSockBufRecv::iMaxLen**

**6.256.2.3 uint8\* PVSockBufRecv::iPtr**

The documentation for this class was generated from the following file:

- [oscl\\_socket\\_request.h](#)

## 6.257 PVSockBufSend Class Reference

```
#include <oscl_socket_request.h>
```

### Public Methods

- [PVSockBufSend \(\)](#)
- [PVSockBufSend \(const uint8 \\*aPtr, uint32 aLen\)](#)
- [PVSockBufSend \(const PVSockBufSend &a\)](#)

### Data Fields

- [const uint8 \\* iPtr](#)
- [uint32 iLen](#)

#### 6.257.1 Constructor & Destructor Documentation

**6.257.1.1 PVSockBufSend::PVSockBufSend () [inline]**

**6.257.1.2 PVSockBufSend::PVSockBufSend (const uint8 \* aPtr, uint32 aLen) [inline]**

**6.257.1.3 PVSockBufSend::PVSockBufSend (const PVSockBufSend & a) [inline]**

#### 6.257.2 Field Documentation

**6.257.2.1 uint32 PVSockBufSend::iLen**

**6.257.2.2 const uint8\* PVSockBufSend::iPtr**

The documentation for this class was generated from the following file:

- [oscl\\_socket\\_request.h](#)

## 6.258 PVThreadContext Class Reference

```
#include <oscl_scheduler_threadcontext.h>
```

### Public Methods

- OSCL\_IMPORT\_REF PVThreadContext ()
- OSCL\_IMPORT\_REF ~PVThreadContext ()
- OSCL\_IMPORT\_REF bool IsSameThreadContext ()
- OSCL\_IMPORT\_REF void EnterThreadContext ()
- OSCL\_IMPORT\_REF void ExitThreadContext ()

### Static Public Methods

- OSCL\_IMPORT\_REF uint32 Id ()
- OSCL\_IMPORT\_REF bool ThreadHasScheduler ()

### Friends

- class PVActiveBase
- class OsclActiveObject
- class OsclTimerObject
- class OsclExecScheduler
- class OsclCoeActiveScheduler
- class OsclExecSchedulerCommonBase
- class OsclExecSchedulerBase
- class OsclCoeActiveSchedulerBase

### 6.258.1 Constructor & Destructor Documentation

#### 6.258.1.1 OSCL\_IMPORT\_REF PVThreadContext::PVThreadContext ()

#### 6.258.1.2 OSCL\_IMPORT\_REF PVThreadContext::~PVThreadContext ()

### 6.258.2 Member Function Documentation

#### 6.258.2.1 OSCL\_IMPORT\_REF void PVThreadContext::EnterThreadContext ()

enter and exit thread context.

#### 6.258.2.2 OSCL\_IMPORT\_REF void PVThreadContext::ExitThreadContext ()

#### 6.258.2.3 OSCL\_IMPORT\_REF uint32 PVThreadContext::Id () [static]

static routine to get a unique thread ID for caller's thread context.

**6.258.2.4 OSCL\_IMPORT\_REF bool PVThreadContext::IsSameThreadContext ()**

compare caller's thread context to this one.

**6.258.2.5 OSCL\_IMPORT\_REF bool PVThreadContext::ThreadHasScheduler () [static]**

a static utility to tell whether the calling thread has any scheduler— either Oscl scheduler or native scheduler.

**6.258.3 Friends And Related Function Documentation****6.258.3.1 friend class OsclActiveObject [friend]****6.258.3.2 friend class OsclCoeActiveScheduler [friend]****6.258.3.3 friend class OsclCoeActiveSchedulerBase [friend]****6.258.3.4 friend class OsclExecScheduler [friend]****6.258.3.5 friend class OsclExecSchedulerBase [friend]****6.258.3.6 friend class OsclExecSchedulerCommonBase [friend]****6.258.3.7 friend class OsclTimerObject [friend]****6.258.3.8 friend class PVActiveBase [friend]**

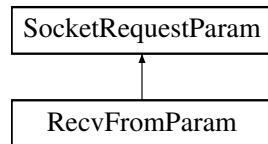
The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_threadcontext.h](#)

## 6.259 RecvFromParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for RecvFromParam::



### Public Methods

- [RecvFromParam \(uint8 \\*aPtr, uint32 aMaxLen, OsclNetworkAddress &aAddress, uint32 flags, uint32 aMultiMax, Oscl\\_Vector< uint32, OsclMemAllocator > \\*aPacketLen, Oscl\\_Vector< OsclNetworkAddress, OsclMemAllocator > \\*aPacketSource\)](#)

### Data Fields

- [PVSockBufRecv iBufRecv](#)
- [uint32 iFlags](#)
- [OsclNetworkAddress & iAddr](#)
- [uint32 iMultiMaxLen](#)
- [Oscl\\_Vector< uint32, OsclMemAllocator > \\* iPacketLen](#)
- [Oscl\\_Vector< OsclNetworkAddress, OsclMemAllocator > \\* iPacketSource](#)

#### 6.259.1 Constructor & Destructor Documentation

**6.259.1.1 RecvFromParam::RecvFromParam (uint8 \*& aPtr, uint32 aMaxLen, OsclNetworkAddress & aAddress, uint32 flags, uint32 aMultiMax, Oscl\_Vector< uint32, OsclMemAllocator > \* aPacketLen, Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator > \* aPacketSource) [inline]**

#### 6.259.2 Field Documentation

**6.259.2.1 OsclNetworkAddress& RecvFromParam::iAddr**

**6.259.2.2 PVSockBufRecv RecvFromParam::iBufRecv**

**6.259.2.3 uint32 RecvFromParam::iFlags**

**6.259.2.4 uint32 RecvFromParam::iMultiMaxLen**

**6.259.2.5 Oscl\_Vector<uint32, OsclMemAllocator>\* RecvFromParam::iPacketLen**

**6.259.2.6 Oscl\_Vector<OsclNetworkAddress, OsclMemAllocator>\***  
**RecvFromParam::iPacketSource**

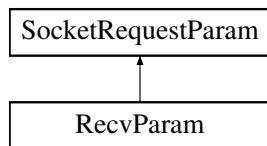
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_request.h](#)

## 6.260 RecvParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for RecvParam::



### Public Methods

- [RecvParam \(uint8 \\*&aPtr, uint32 aMaxLen, uint32 flags\)](#)

### Data Fields

- [PVSockBufRecv iBufRecv](#)
- [uint32 iFlags](#)

#### 6.260.1 Constructor & Destructor Documentation

**6.260.1.1 RecvParam::RecvParam (uint8 \*& aPtr, uint32 aMaxLen, uint32 flags) [inline]**

#### 6.260.2 Field Documentation

**6.260.2.1 PVSockBufRecv RecvParam::iBufRecv**

**6.260.2.2 uint32 RecvParam::iFlags**

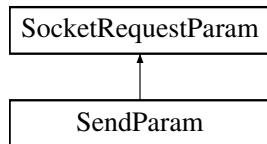
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_request.h](#)

## 6.261 SendParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for SendParam::



### Public Methods

- [SendParam \(const uint8 \\*&aPtr, uint32 aLen, uint32 aFlags\)](#)

### Data Fields

- [PVSockBufSend iBufSend](#)
- [uint32 iFlags](#)
- [uint32 iXferLen](#)

#### 6.261.1 Detailed Description

Socket method parameter sets

#### 6.261.2 Constructor & Destructor Documentation

6.261.2.1 [SendParam::SendParam \(const uint8 \\*& aPtr, uint32 aLen, uint32 aFlags\) \[inline\]](#)

#### 6.261.3 Field Documentation

6.261.3.1 [PVSockBufSend SendParam::iBufSend](#)

6.261.3.2 [uint32 SendParam::iFlags](#)

6.261.3.3 [uint32 SendParam::iXferLen](#)

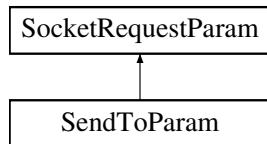
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_request.h](#)

## 6.262 SendToParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for SendToParam::



### Public Methods

- [SendToParam \(const uint8 \\*aPtr, uint32 aLen, OsclNetworkAddress &aAddr, uint32 flags\)](#)
- [~SendToParam \(\)](#)

### Data Fields

- [PVSockBufSend iBufSend](#)
- [uint32 iFlags](#)
- [OsclNetworkAddress iAddr](#)
- [uint32 iXferLen](#)

#### 6.262.1 Constructor & Destructor Documentation

**6.262.1.1 SendToParam::SendToParam (const uint8 \*& aPtr, uint32 aLen, OsclNetworkAddress & aAddr, uint32 flags) [inline]**

**6.262.1.2 SendToParam::~SendToParam () [inline]**

#### 6.262.2 Field Documentation

**6.262.2.1 OsclNetworkAddress SendToParam::iAddr**

**6.262.2.2 PVSockBufSend SendToParam::iBufSend**

**6.262.2.3 uint32 SendToParam::iFlags**

**6.262.2.4 uint32 SendToParam::iXferLen**

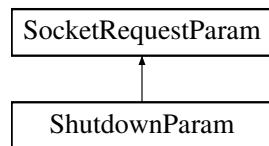
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_request.h](#)

## 6.263 ShutdownParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for ShutdownParam::



### Public Methods

- [ShutdownParam \(TPVSocketShutdown aHow\)](#)

### Data Fields

- [TPVSocketShutdown iHow](#)

#### 6.263.1 Constructor & Destructor Documentation

6.263.1.1 ShutdownParam::ShutdownParam ([TPVSocketShutdown aHow](#)) [inline]

#### 6.263.2 Field Documentation

##### 6.263.2.1 [TPVSocketShutdown ShutdownParam::iHow](#)

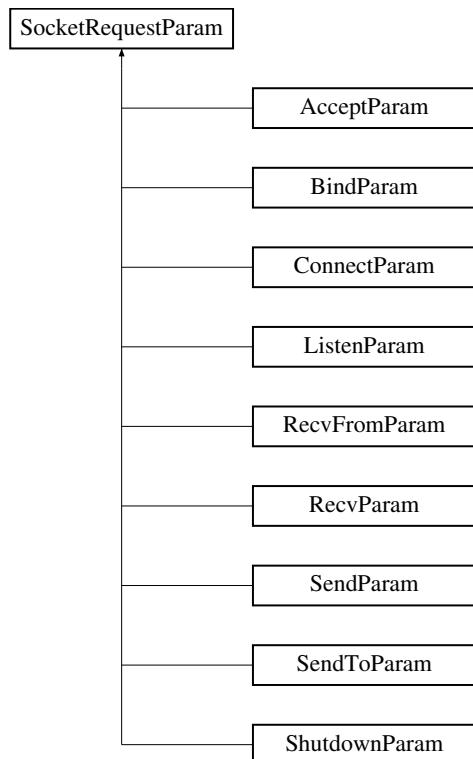
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_request.h](#)

## 6.264 SocketRequestParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for SocketRequestParam::



### Public Methods

- [SocketRequestParam \(TPVSocketFxn aFxn\)](#)

### Data Fields

- [TPVSocketFxn iFxn](#)

#### 6.264.1 Detailed Description

Base class for all socket method parameter sets

## 6.264.2 Constructor & Destructor Documentation

6.264.2.1 `SocketRequestParam::SocketRequestParam (TPVSocketFxn aFxn)` [inline]

## 6.264.3 Field Documentation

6.264.3.1 `TPVSocketFxn SocketRequestParam::iFxn`

The documentation for this class was generated from the following file:

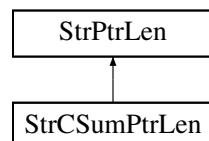
- `oscl_socket_request.h`

## 6.265 StrCSumPtrLen Struct Reference

same as [StrPtrLen](#), but includes checksum field and method to speed up querying

```
#include <oscl_str_ptr_len.h>
```

Inheritance diagram for StrCSumPtrLen::



### Public Types

- [typedef int16 CheckSumType](#)

### Public Methods

- [void setPtrLen \(const char \\*newPtr, uint32 newLen\)](#)
- [CheckSumType getCheckSum \(\) const](#)
- [OSCL\\_IMPORT\\_REF void setCheckSum \(\)](#)
- [StrCSumPtrLen \(\)](#)
- [StrCSumPtrLen \(const char \\*newPtr\)](#)
- [StrCSumPtrLen \(const char \\*newPtr, uint32 newLen\)](#)
- [StrCSumPtrLen \(const StrCSumPtrLen &rhs\)](#)
- [StrCSumPtrLen \(const StrPtrLen &rhs\)](#)
- [c\\_bool isCIEquivalentTo \(const StrCSumPtrLen &rhs\) const](#)
- [c\\_bool operator== \(const StrCSumPtrLen &rhs\) const](#)
- [c\\_bool operator!= \(const StrCSumPtrLen &rhs\) const](#)
- [StrCSumPtrLen & operator= \(const StrCSumPtrLen &rhs\)](#)
- [StrCSumPtrLen & operator= \(const StrPtrLen &rhs\)](#)
- [StrCSumPtrLen & operator= \(const char \\*rhs\)](#)

### Protected Attributes

- [CheckSumType checkSum](#)

#### 6.265.1 Detailed Description

same as [StrPtrLen](#), but includes checksum field and method to speed up querying

## 6.265.2 Member Typedef Documentation

**6.265.2.1** `typedef int16 StrCSumPtrLen::CheckSumType`

## 6.265.3 Constructor & Destructor Documentation

**6.265.3.1** `StrCSumPtrLen::StrCSumPtrLen () [inline]`

**6.265.3.2** `StrCSumPtrLen::StrCSumPtrLen (const char * newPtr) [inline]`

**6.265.3.3** `StrCSumPtrLen::StrCSumPtrLen (const char * newPtr, uint32 newLen) [inline]`

**6.265.3.4** `StrCSumPtrLen::StrCSumPtrLen (const StrCSumPtrLen & rhs) [inline]`

**6.265.3.5** `StrCSumPtrLen::StrCSumPtrLen (const StrPtrLen & rhs) [inline]`

## 6.265.4 Member Function Documentation

**6.265.4.1** `CheckSumType StrCSumPtrLen::getCheckSum () const [inline]`

**6.265.4.2** `c_bool StrCSumPtrLen::isCIEquivalentTo (const StrCSumPtrLen & rhs) const [inline]`

**6.265.4.3** `c_bool StrCSumPtrLen::operator!= (const StrCSumPtrLen & rhs) const [inline]`

**6.265.4.4** `StrCSumPtrLen& StrCSumPtrLen::operator= (const char * rhs) [inline]`

Reimplemented from [StrPtrLen](#).

**6.265.4.5** `StrCSumPtrLen& StrCSumPtrLen::operator= (const StrPtrLen & rhs) [inline]`

Reimplemented from [StrPtrLen](#).

**6.265.4.6** `StrCSumPtrLen& StrCSumPtrLen::operator= (const StrCSumPtrLen & rhs) [inline]`

**6.265.4.7** `c_bool StrCSumPtrLen::operator== (const StrCSumPtrLen & rhs) const [inline]`

**6.265.4.8** `OSCL_IMPORT_REF void StrCSumPtrLen::setCheckSum ()`

**6.265.4.9** `void StrCSumPtrLen::setPtrLen (const char * newPtr, uint32 newLen) [inline]`

Reimplemented from [StrPtrLen](#).

## 6.265.5 Field Documentation

**6.265.5.1** `CheckSumType StrCSumPtrLen::checkSum [protected]`

The documentation for this struct was generated from the following file:

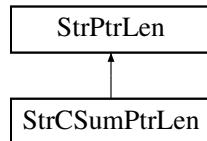
- [oscl\\_str\\_ptr\\_len.h](#)

## 6.266 StrPtrLen Struct Reference

This data structure encapsulates a set of functions used to perform.

```
#include <oscl_str_ptr_len.h>
```

Inheritance diagram for StrPtrLen::



### Public Methods

- [StrPtrLen](#) (const char \*newPtr)
- [StrPtrLen](#) (const char \*newPtr, uint32 newLen)
- [StrPtrLen](#) ()
- [StrPtrLen](#) (const StrPtrLen &rhs)
- const char \* [c\\_str](#) () const
- int32 [length](#) () const
- int32 [size](#) () const
- void [setPtrLen](#) (const char \*newPtr, uint32 newLen)
- [c\\_bool](#) [isCIEquivalentTo](#) (const StrPtrLen &rhs) const
- [c\\_bool](#) [isCIPrefixOf](#) (const StrPtrLen &rhs) const
- int32 [operator==](#) (const StrPtrLen &rhs) const
- int32 [operator!=](#) (const StrPtrLen &rhs) const
- StrPtrLen & [operator=](#) (const StrPtrLen &rhs)
- StrPtrLen & [operator=](#) (const char \*rhs)

### Protected Methods

- bool [isLetter](#) (const char c) const

### Protected Attributes

- const char \* [ptr](#)
- int32 [len](#)

#### 6.266.1 Detailed Description

This data structure encapsulates a set of functions used to perform.

standard string operations. It should be used for null-terminated constant (non-modifiable) strings of char type.

## 6.266.2 Constructor & Destructor Documentation

**6.266.2.1** `StrPtrLen::StrPtrLen (const char * newPtr) [inline]`

**6.266.2.2** `StrPtrLen::StrPtrLen (const char * newPtr, uint32 newLen) [inline]`

**6.266.2.3** `StrPtrLen::StrPtrLen () [inline]`

**6.266.2.4** `StrPtrLen::StrPtrLen (const StrPtrLen & rhs) [inline]`

## 6.266.3 Member Function Documentation

**6.266.3.1** `const char* StrPtrLen::c_str () const [inline]`

**6.266.3.2** `c_bool StrPtrLen::isCIEquivalentTo (const StrPtrLen & rhs) const [inline]`

**6.266.3.3** `c_bool StrPtrLen::isCIPrefixOf (const StrPtrLen & rhs) const [inline]`

**6.266.3.4** `bool StrPtrLen::isLetter (const char c) const [inline, protected]`

**6.266.3.5** `int32 StrPtrLen::length () const [inline]`

**6.266.3.6** `int32 StrPtrLen::operator!= (const StrPtrLen & rhs) const [inline]`

**6.266.3.7** `StrPtrLen& StrPtrLen::operator= (const char * rhs) [inline]`

Reimplemented in [StrCSumPtrLen](#).

**6.266.3.8** `StrPtrLen& StrPtrLen::operator= (const StrPtrLen & rhs) [inline]`

Reimplemented in [StrCSumPtrLen](#).

**6.266.3.9** `int32 StrPtrLen::operator== (const StrPtrLen & rhs) const [inline]`

**6.266.3.10** `void StrPtrLen::setPtrLen (const char * newPtr, uint32 newLen) [inline]`

Reimplemented in [StrCSumPtrLen](#).

**6.266.3.11** `int32 StrPtrLen::size () const [inline]`

## 6.266.4 Field Documentation

**6.266.4.1** `int32 StrPtrLen::len [protected]`

**6.266.4.2** `const char* StrPtrLen::ptr [protected]`

The documentation for this struct was generated from the following file:

- [oscl\\_str\\_ptr\\_len.h](#)

## 6.267 TimeValue Class Reference

The TimeValue class represents a time value in a format native to the system.

```
#include <oscl_time.h>
```

### Public Methods

- OSCL\_COND\_IMPORT\_REF [TimeValue](#) ()
 

*Create a TimeValue representing the current time.*
- OSCL\_COND\_IMPORT\_REF [TimeValue](#) (const [TimeValue](#) &Tv)
 

*Copy constructor.*
- OSCL\_COND\_IMPORT\_REF [TimeValue](#) (long tv, [TimeUnits](#) units)
 

*Create a TimeValue representing an interval of tv units.*
- OSCL\_COND\_IMPORT\_REF [TimeValue](#) (const [OsclBasicTimeStruct](#) &in\_tv)
 

*Create a TimeValue representing the absolute time specified by the BasicTimeStruct.*
- OSCL\_COND\_IMPORT\_REF [TimeValue](#) ([OsclBasicDateTimeStruct](#) in\_ts)
 

*Create a TimeValue representing the absolute time specified by the BasicDateTimeStruct.*
- OSCL\_COND\_IMPORT\_REF int32 [get\\_local\\_time](#) ()
 

*Get the local time after having adjusted for daylight saving.*
- OSCL\_COND\_IMPORT\_REF void [set\\_to\\_zero](#) ()
 

*Set the time value to zero (represents a zero interval).*
- OSCL\_COND\_IMPORT\_REF void [set\\_to\\_current\\_time](#) ()
 

*Set the time value to the current system time.*
- OSCL\_COND\_IMPORT\_REF void [set\\_from\\_ntp\\_time](#) (const uint32 ntp\_offset)
 

*This method covers a 32-bit NTP offset to system time.*
- OSCL\_COND\_IMPORT\_REF uint32 [get\\_sec](#) () const
 

*Get a 32 bit value representing the seconds since the (system dependent) epoch.*
- OSCL\_COND\_IMPORT\_REF int32 [to\\_msec](#) () const
 

*Get a 32 bit value representing the number of microseconds in the time value.*
- OSCL\_IMPORT\_REF char \* [get\\_str\\_ctime](#) (CtimeStrBuf ctime\_strbuf)
 

*Get a string containing a text representation of this TimeValue object.*
- OSCL\_IMPORT\_REF int [get\\_pv8601\\_str\\_time](#) (PV8601timeStrBuf time\_strbuf)
 

*Get a PV extended text representation of the Time based on the ISO 8601 format.*
- OSCL\_IMPORT\_REF char \* [get\\_rfc822\\_gmtime\\_str](#) (int max\_time\_strlen, char \*time\_str)
 

*Get a string containing a text representation of this TimeValue object.*

*Get a text representation of the time in the GMT timezone based on the RFC 822 / RFC 1123 (also described in the HTTP spec RFC 2068 and RFC 2616.*

- OSCL\_COND\_IMPORT\_REF bool `is_zero ()`  
*Determine if the time value is zero.*
- OSCL\_COND\_IMPORT\_REF TimeValue & `operator= (const TimeValue &a)`  
*Assignment operator.*
- OSCL\_COND\_IMPORT\_REF TimeValue & `operator+= (const TimeValue &a)`  
 $+ = \text{operator}$
- OSCL\_COND\_IMPORT\_REF TimeValue & `operator-= (const TimeValue &a)`  
 $- = \text{operator}$
- OSCL\_COND\_IMPORT\_REF TimeValue & `operator *= (const int scale)`  
*This operator scales the time value by a constant.*
- OSCL\_COND\_IMPORT\_REF OsclBasicTimeStruct \* `get_timeval_ptr ()`

## Friends

- class `NTPTime`
- OSCL\_COND\_IMPORT\_REF friend bool `operator== (const TimeValue &a, const TimeValue &b)`
- OSCL\_COND\_IMPORT\_REF friend bool `operator!= (const TimeValue &a, const TimeValue &b)`
- OSCL\_COND\_IMPORT\_REF friend bool `operator<= (const TimeValue &a, const TimeValue &b)`
- OSCL\_COND\_IMPORT\_REF friend bool `operator>= (const TimeValue &a, const TimeValue &b)`
- OSCL\_COND\_IMPORT\_REF friend bool `operator< (const TimeValue &a, const TimeValue &b)`
- OSCL\_COND\_IMPORT\_REF friend bool `operator> (const TimeValue &a, const TimeValue &b)`

### 6.267.1 Detailed Description

The TimeValue class represents a time value in a format native to the system.

This class provides common time functions independent of any OS. The actual representation used is native to the system that the class is compiled on to increase efficiency. Macros used in this class:

- OSCL\_HAS\_ANSI\_STRING\_SUPPORT:

Definitions to determine the type of basic time structure used to store the time

- OSCL\_HAS\_UNIX\_TIME\_FUNCS
- OSCL\_HAS\_SYMBIAN\_SUPPORT
- OSCL\_HAS\_MSWIN\_SUPPORT

### 6.267.2 Constructor & Destructor Documentation

#### 6.267.2.1 OSCL\_COND\_IMPORT\_REF TimeValue::TimeValue ()

Create a TimeValue representing the current time.

**6.267.2.2 OSCL\_COND\_IMPORT\_REF TimeValue::TimeValue (const TimeValue & *Tv*)**

Copy constructor.

**6.267.2.3 OSCL\_COND\_IMPORT\_REF TimeValue::TimeValue (long *tv*, TimeUnits *units*)**

Create a TimeValue representing an interval of tv units.

**Parameters:**

*tv* The number of units in the interval to be represented by this TimeValue.

*units* The units of the tv argument. Must be in the enumeration TimeUnits.

**6.267.2.4 OSCL\_COND\_IMPORT\_REF TimeValue::TimeValue (const OsclBasicTimeStruct & *in\_tv*)**

Create a TimeValue representing the absolute time specified by the BasicTimeStruct.

**Parameters:**

*in\_tv* OsclBasicTimeStruct as defined for each platform.

**6.267.2.5 OSCL\_COND\_IMPORT\_REF TimeValue::TimeValue (OsclBasicDateTimeStruct *in\_ts*)**

Create a TimeValue representing the absolute time specified by the BasicDateTimeStruct.

**Parameters:**

*in\_ts* OsclBasicDateTimeStruct as defined for each platform provides the date in a readable format (i.e. day, date , week etc.) Notes: Implementation incomplete (= not done) on Win32, Wince, Symbian

### 6.267.3 Member Function Documentation

**6.267.3.1 OSCL\_COND\_IMPORT\_REF int32 TimeValue::get\_local\_time ()**

Get the local time after having adjusted for daylight saving.

Notes: Implementation incomplete (= not done) on Win32, Wince, Symbian

**6.267.3.2 OSCL\_IMPORT\_REF int TimeValue::get\_pv8601\_str\_time (PV8601timeStrBuf *time\_strbuf*)**

Get a PV extended text representation of the Time based on the ISO 8601 format.

**Parameters:**

*time\_strbuf* A PV8601timeStrBuf object to which the text representation will be written,

**Returns:**

The number of characters copied to the buffer, not including the terminating null. The returned string is of the form "19850412T101530.047Z".

**6.267.3.3 OSCL\_IMPORT\_REF char\* TimeValue::get\_rfc822\_gmtime\_str (int max\_time\_strlen, char \* time\_str)**

Get a text representation of the time in the GMT timezone based on the RFC 822 / RFC 1123 (also described in the HTTP spec RFC 2068 and RFC 2616).

**Parameters:**

*max\_time\_strlen* The maximum number of characters that can be written to the buffer.

*time\_str* A pointer to the buffer to which the characters will be written.

**Returns:**

Returns a pointer to the buffer (same as *time\_str*) containing a null terminated (c-style) string of the form "Wed, 30 Jun 1993 21:49:08 GMT".

**6.267.3.4 OSCL\_COND\_IMPORT\_REF uint32 TimeValue::get\_sec ()**

Get a 32 bit value representing the seconds since the (system dependent) epoch.

**Returns:**

This call returns a 32 bit value representing the number of seconds since the epoch. On unix systems this represents the number of seconds since the unix epoch Jan 1 1970. On Win32 this represents the number of seconds since Jan 1, 1601. This is intended to be used for intervals rather than for absolute time. (On Win32 for example, a 32 bit value would be too small to represent the number of seconds from the epoch until the current time.)

**6.267.3.5 OSCL\_IMPORT\_REF char\* TimeValue::get\_str\_ctime (CtimeStrBuf ctime\_strbuf)**

Get a string containing a text representation of this TimeValue object.

**Parameters:**

*ctime\_strbuf* A CtimeStrBuf object to which the text representation will be written,

**Returns:**

A pointer to the input CtimeStrBuf is returned. This string is null terminated of the form "Wed Jun 30 21:49:08 1993".

**6.267.3.6 OSCL\_COND\_IMPORT\_REF OsclBasicTimeStruct\* TimeValue::get\_timeval\_ptr ()****6.267.3.7 OSCL\_COND\_IMPORT\_REF uint32 TimeValue::get\_usec ()**

Get a 32 bit value representing the number of microseconds in the time value.

**Returns:**

Returns a uint32 value representing the number of microseconds in the time value after subtracting off the whole seconds.

**6.267.3.8 OSCL\_COND\_IMPORT\_REF bool TimeValue::is\_zero ()**

Determine if the time value is zero.

**6.267.3.9 OSCL\_COND\_IMPORT\_REF TimeValue& TimeValue::operator \*= (const int scale)**

This operator scales the time value by a constant.

**6.267.3.10 OSCL\_COND\_IMPORT\_REF TimeValue& TimeValue::operator+= (const TimeValue & a)**

+= operator

**6.267.3.11 OSCL\_COND\_IMPORT\_REF TimeValue& TimeValue::operator-= (const TimeValue & a)**

-= operator

**6.267.3.12 OSCL\_COND\_IMPORT\_REF TimeValue& TimeValue::operator= (const TimeValue & a)**

Assignment operator.

**6.267.3.13 OSCL\_COND\_IMPORT\_REF void TimeValue::set\_from\_ntp\_time (const uint32 ntp\_offset)**

This method converts a 32-bit NTP offset to system time.

This method takes a 32-bit ntp offset which is the number of seconds from 0 h Jan 1, 1900 and converts it to the system time

**6.267.3.14 OSCL\_COND\_IMPORT\_REF void TimeValue::set\_to\_current\_time ()**

Set the time value to the current system time.

**6.267.3.15 OSCL\_COND\_IMPORT\_REF void TimeValue::set\_to\_zero ()**

Set the time value to zero (represents a zero interval).

**6.267.3.16 OSCL\_COND\_IMPORT\_REF int32 TimeValue::to\_msec ()**

## 6.267.4 Friends And Related Function Documentation

**6.267.4.1 friend class NTPTime [friend]**

**6.267.4.2 OSCL\_COND\_IMPORT\_REF friend bool operator!= (const TimeValue & a, const TimeValue & b) [friend]**

**6.267.4.3 OSCL\_COND\_IMPORT\_REF friend bool operator< (const TimeValue & a, const TimeValue & b) [friend]**

**6.267.4.4 OSCL\_COND\_IMPORT\_REF friend bool operator<= (const TimeValue & a, const TimeValue & b) [friend]**

**6.267.4.5 OSCL\_COND\_IMPORT\_REF friend bool operator== (const TimeValue & a, const TimeValue & b) [friend]**

**6.267.4.6 OSCL\_COND\_IMPORT\_REF friend bool operator> (const TimeValue & a, const TimeValue & b) [friend]**

**6.267.4.7 OSCL\_COND\_IMPORT\_REF friend bool operator>= (const TimeValue & a, const TimeValue & b) [friend]**

The documentation for this class was generated from the following file:

- [oscl\\_time.h](#)

## 6.268 TLSStorageOps Class Reference

```
#include <oscl_tls.h>
```

### Static Public Methods

- OSCL\_IMPORT\_REF void [save\\_registry \(TOsclTlsKey \\*key, OsclAny \\*ptr, int32 &\)](#)
- OSCL\_IMPORT\_REF [OsclAny \\* get\\_registry \(TOsclTlsKey \\*key\)](#)

#### 6.268.1 Member Function Documentation

**6.268.1.1 OSCL\_IMPORT\_REF OsclAny\* TLSStorageOps::get\_registry (TOsclTlsKey \* *key*)  
[static]**

**6.268.1.2 OSCL\_IMPORT\_REF void TLSStorageOps::save\_registry (TOsclTlsKey \* *key*,  
OsclAny \* *ptr*, int32 &) [static]**

The documentation for this class was generated from the following file:

- [oscl\\_tls.h](#)

## 6.269 TReadyQueLink Class Reference

```
#include <oscl_scheduler_readyq.h>
```

### Public Methods

- [TReadyQueLink \(\)](#)

### Data Fields

- int32 [iAOPriority](#)
- uint32 [iTimeToRunTicks](#)
- uint32 [iTimeQueuedTicks](#)
- uint32 [iSeqNum](#)
- OsclAny \* [iIsIn](#)

#### 6.269.1 Detailed Description

This class defines the queue link, which is common to both ready Q and timer Q. Each AO contains its own queue link object.

#### 6.269.2 Constructor & Destructor Documentation

##### 6.269.2.1 [TReadyQueLink::TReadyQueLink \(\) \[inline\]](#)

#### 6.269.3 Field Documentation

##### 6.269.3.1 [int32 TReadyQueLink::iAOPriority](#)

##### 6.269.3.2 [OsclAny\\* TReadyQueLink::iIsIn](#)

##### 6.269.3.3 [uint32 TReadyQueLink::iSeqNum](#)

##### 6.269.3.4 [uint32 TReadyQueLink::iTimeQueuedTicks](#)

##### 6.269.3.5 [uint32 TReadyQueLink::iTimeToRunTicks](#)

The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_readyq.h](#)

## 6.270 WStrPtrLen Struct Reference

This data structure encapsulates a set of functions used to perform.

```
#include <oscl_str_ptr_len.h>
```

### Public Methods

- [WStrPtrLen \(const oscl\\_wchar \\*newPtr\)](#)
- [WStrPtrLen \(const oscl\\_wchar \\*newPtr, uint32 newLen\)](#)
- [WStrPtrLen \(\)](#)
- [WStrPtrLen \(const WStrPtrLen &rhs\)](#)
- [const oscl\\_wchar \\* c\\_str \(\) const](#)
- [int32 length \(\) const](#)
- [int32 size \(\) const](#)
- [void setPtrLen \(const oscl\\_wchar \\*newPtr, uint32 newLen\)](#)
- [c\\_bool isCIEquivalentTo \(const WStrPtrLen &rhs\) const](#)
- [int32 operator== \(const WStrPtrLen &rhs\) const](#)
- [int32 operator!= \(const WStrPtrLen &rhs\) const](#)
- [WStrPtrLen & operator= \(const WStrPtrLen &rhs\)](#)
- [WStrPtrLen & operator= \(const oscl\\_wchar \\*rhs\)](#)

### Protected Attributes

- [const oscl\\_wchar \\* ptr](#)
- [int32 len](#)

#### 6.270.1 Detailed Description

This data structure encapsulates a set of functions used to perform.

standard string operations. It should be used for null-terminated constant strings (non-modifiable) of wchar type.

## 6.270.2 Constructor & Destructor Documentation

- 6.270.2.1 `WStrPtrLen::WStrPtrLen (const oscl_wchar * newPtr) [inline]`
- 6.270.2.2 `WStrPtrLen::WStrPtrLen (const oscl_wchar * newPtr, uint32 newLen) [inline]`
- 6.270.2.3 `WStrPtrLen::WStrPtrLen () [inline]`
- 6.270.2.4 `WStrPtrLen::WStrPtrLen (const WStrPtrLen & rhs) [inline]`

## 6.270.3 Member Function Documentation

- 6.270.3.1 `const oscl_wchar* WStrPtrLen::c_str () const [inline]`
- 6.270.3.2 `c_bool WStrPtrLen::isCIEquivalentTo (const WStrPtrLen & rhs) const [inline]`
- 6.270.3.3 `int32 WStrPtrLen::length () const [inline]`
- 6.270.3.4 `int32 WStrPtrLen::operator!= (const WStrPtrLen & rhs) const [inline]`
- 6.270.3.5 `WStrPtrLen& WStrPtrLen::operator= (const oscl_wchar * rhs) [inline]`
- 6.270.3.6 `WStrPtrLen& WStrPtrLen::operator= (const WStrPtrLen & rhs) [inline]`
- 6.270.3.7 `int32 WStrPtrLen::operator== (const WStrPtrLen & rhs) const [inline]`
- 6.270.3.8 `void WStrPtrLen::setPtrLen (const oscl_wchar * newPtr, uint32 newLen) [inline]`
- 6.270.3.9 `int32 WStrPtrLen::size () const [inline]`

## 6.270.4 Field Documentation

- 6.270.4.1 `int32 WStrPtrLen::len [protected]`
- 6.270.4.2 `const oscl_wchar* WStrPtrLen::ptr [protected]`

The documentation for this struct was generated from the following file:

- `oscl_str_ptr_len.h`

# Chapter 7

## oscl File Documentation

### 7.1 oscl\_aostatus.h File Reference

Some basic types used with active objects.

```
#include "osclconfig_proc.h"
#include "oscl_base.h"
#include "oscl_aostatus.inl"
```

#### Data Structures

- class [OsclAOStatus](#)

#### Variables

- const int32 [OSCL\\_REQUEST\\_ERR\\_NONE](#) = 0
- const int32 [OSCL\\_REQUEST\\_PENDING](#) = (-0x7fffffff)
- const int32 [OSCL\\_REQUEST\\_ERR\\_CANCEL](#) = (-1)
- const int32 [OSCL\\_REQUEST\\_ERR\\_GENERAL](#) = (-2)

#### 7.1.1 Detailed Description

Some basic types used with active objects.

## 7.2 oscl\_assert.h File Reference

The file [oscl\\_assert.h](#) provides an OSCL\_ASSERT macro to document assumptions and test them during development.

```
#include "oscl_base.h"  
#include "oscl_assert.inl"
```

### Defines

- #define [OSCL\\_ASSERT](#)(\_expr) ((\_expr)?((void)0):OSCL Assert(# \_expr,\_\_FILE\_\_,\_\_LINE\_\_))

### Functions

- OSCL\_COND\_IMPORT\_REF void [\\_OSCL\\_Abort](#) ()  
*This function terminates the current process abnormally.*
- OSCL\_IMPORT\_REF void [OSCL Assert](#) (const char \*expr, const char \*filename, int line\_number)  
*OSCL\_ASSERT macro evaluates an expression and when the result is false, prints a diagnostic message and aborts the program.*

### 7.2.1 Detailed Description

The file [oscl\\_assert.h](#) provides an OSCL\_ASSERT macro to document assumptions and test them during development.

## 7.3 oscl\_base.h File Reference

The file [oscl\\_base.h](#) is the public header that should be included to pick up the platform configuration, basic type definitions, and common macros.

```
#include "osclconfig.h"
#include "oscl_base_macros.h"
#include "oscl_types.h"
#include "osclconfig_check.h"
#include "pv_config.h"
```

### Defines

- `#define OSCL_HAS_SINGLETON_SUPPORT 1`

### Functions

- `void PVOsclBase_Init ()`
- `void PVOsclBase_Cleanup ()`

#### 7.3.1 Detailed Description

The file [oscl\\_base.h](#) is the public header that should be included to pick up the platform configuration, basic type definitions, and common macros.

## **7.4 oscl\_base\_alloc.h File Reference**

A basic allocator that does not rely on other modules.

```
#include "osclconfig.h"  
#include "oscl_defalloc.h"  
#include "osclconfig_memory.h"
```

### **Data Structures**

- class [\\_OsclBasicAllocator](#)

#### **7.4.1 Detailed Description**

A basic allocator that does not rely on other modules.

## 7.5 oscl\_base\_macros.h File Reference

This file defines common macros and constants for basic compilation support.

```
#include "osclconfig.h"
```

### Defines

- #define **NULL\_TERM\_CHAR** '\0'  
*The NULL\_TERM\_CHAR is used to terminate c-style strings.*
- #define **NULL** (0)  
*if the NULL macro isn't already defined, then define it as zero.*
- #define **OSCL\_INLINE** inline
- #define **OSCL\_COND\_EXPORT\_REF**
- #define **OSCL\_COND\_IMPORT\_REF**
- #define **OSCL\_CONST\_CAST**(type, exp) ((type)(exp))  
*Type casting macros.*
- #define **OSCL\_STATIC\_CAST**(type, exp) ((type)(exp))
- #define **OSCL\_REINTERPRET\_CAST**(type, exp) ((type)(exp))
- #define **OSCL\_DYNAMIC\_CAST**(type, exp) ((type)(exp))
- #define **OSCL\_UNUSED\_ARG**(vbl) (void)(vbl)
- #define **OSCL\_UNUSED\_RETURN**(value) return value
- #define **OSCL\_MIN**(a, b) ((a) < (b) ? (a) : (b))
- #define **OSCL\_MAX**(a, b) ((a) > (b) ? (a) : (b))
- #define **OSCL\_ABS**(a) ((a) > (0) ? (a) : -(a))
- #define **OSCL\_TEMPLATED\_DESTRUCTOR\_CALL**(type, simple\_type) type :: ~simple\_type ()
- #define **OSCL\_UNSIGNED\_CONST**(x) x
- #define **OSCL\_PACKED\_VAR** "error"
- #define **OSCL\_BEGIN\_PACKED** "error"
- #define **OSCL\_END\_PACKED** "error"

### 7.5.1 Detailed Description

This file defines common macros and constants for basic compilation support.

## 7.6 oscl\_bin\_stream.h File Reference

Defines a set of binary stream classes which handle portable input / output of binary data regardless of the native byte order.

```
#include "oscl_base.h"
#include "oscl_bin_stream.inl"
```

### Data Structures

- class [OsclBinIStream](#)
- class [OsclBinIStreamBigEndian](#)
- class [OsclBinIStreamLittleEndian](#)
- class [OsclBinOStream](#)

*Class OsclBinOStream implements the basic stream functions for an output stream.*

- class [OsclBinOStreamBigEndian](#)

*Class OsclBinOStreamBigEndian implements a binary output stream using big endian byte ordering.*

- class [OsclBinOStreamLittleEndian](#)

*Class OsclBinOStreamLittleEndian implements a binary output stream using little endian byte ordering.*

- class [OsclBinStream](#)

### 7.6.1 Detailed Description

Defines a set of binary stream classes which handle portable input / output of binary data regardless of the native byte order.

## 7.7 oscl\_byte\_order.h File Reference

This file defines functions providing byte ordering utility (e.g., switching between big and little endian orders).

```
#include "oscl_base.h"  
#include "oscl_byte_order.inl"
```

### Functions

- void [little\\_endian\\_to\\_host](#) (char \*data, uint32 size)  
*Convert little endian to host format.*
- void [host\\_to\\_little\\_endian](#) (char \*data, unsigned int size)  
*Convert host to little endian format.*
- void [big\\_endian\\_to\\_host](#) (char \*data, unsigned int size)  
*Convert big endian to host format.*
- void [host\\_to\\_big\\_endian](#) (char \*data, unsigned int size)  
*Convert host to big endian format.*

### 7.7.1 Detailed Description

This file defines functions providing byte ordering utility (e.g., switching between big and little endian orders).

## 7.8 oscl\_defalloc.h File Reference

The file defines simple default memory allocator classes. These allocators are used by the [Oscl\\_Vector](#) and [Oscl\\_Map](#) class, etc.

```
#include "oscl_base.h"  
#include "osclconfig_compiler_warnings.h"  
#include "oscl_mem_inst.h"
```

### Data Structures

- class [Oscl\\_Alloc](#)
- class [Oscl\\_Dealloc](#)
- class [Oscl\\_DefAlloc](#)
- class [Oscl\\_TAlloc](#)
- class [OsclAllocDestructDealloc](#)
- class [OsclDestructDealloc](#)
- struct [rebind](#)

### Defines

- #define [OSCL\\_DISABLE\\_WARNING\\_TRUNCATE\\_DEBUG\\_MESSAGE](#)
- #define [ALLOCATE\(n\)](#) [allocate\\_fl\(n,\\_\\_FILE\\_\\_,\\_\\_LINE\\_\\_\)](#)
- #define [ALLOC\\_AND\\_CONSTRUCT\(n\)](#) [alloc\\_and\\_construct\\_fl\(n,\\_\\_FILE\\_\\_,\\_\\_LINE\\_\\_\)](#)

#### 7.8.1 Detailed Description

The file defines simple default memory allocator classes. These allocators are used by the [Oscl\\_Vector](#) and [Oscl\\_Map](#) class, etc.

## 7.9 oscl\_dll.h File Reference

Defines a DLL entry point.

```
#include "osclconfig.h"
```

### Defines

- #define **OSCL\_DLL\_ENTRY\_POINT()** void oscl\_dll\_entry\_point() {}
- #define **OSCL\_DLL\_ENTRY\_POINT\_DEFAULT()**

#### 7.9.1 Detailed Description

Defines a DLL entry point.

## 7.10 oscl\_dns.h File Reference

The file [oscl\\_socket.h](#) defines the OSCL DNS APIs.

```
#include "osclconfig_io.h"
#include "oscl_socket_types.h"
#include "oscl_defalloc.h"
#include "oscl_socket.h"
```

### Data Structures

- class [OsclIDNS](#)
- class [OsclDNSObserver](#)

### Enumerations

- enum [TPVDNSFxn](#) { [EPVDNSGetHostByName](#) }
- enum [TPVDNSEvent](#) { [EPVDNSSuccess](#), [EPVDNSPending](#), [EPVDNSTimeout](#), [EPVDNSFailure](#), [EPVDNSCancel](#) }

#### 7.10.1 Detailed Description

The file [oscl\\_socket.h](#) defines the OSCL DNS APIs.

## 7.11 oscl\_dns\_gethostbyname.h File Reference

```
#include "oscl_dns_method.h"
#include "oscl_dns.h"
#include "osclconfig_io.h"
```

### Data Structures

- class [OsclGetHostByNameMethod](#)
- class [OsclGetHostByNameRequest](#)

## **7.12 oscl\_dns\_imp.h File Reference**

```
#include "oscl_dns_tuneables.h"  
#include "oscl_dns_imp_pv.h"
```

## 7.13 oscl\_dns\_imp\_base.h File Reference

```
#include "oscl_socket_imp.h"
#include "oscl_dns_request.h"
#include "oscl_dns.h"
```

### Data Structures

- class [OsclDNSIBase](#)

## 7.14 oscl\_dns\_imp\_pv.h File Reference

```
#include "oscl_socket_imp_base.h"
#include "oscl_dns_request.h"
#include "oscl_dns_imp_base.h"
```

### Data Structures

- class [OsclDNSI](#)

## 7.15 oscl\_dns\_method.h File Reference

```
#include "osclconfig_io.h"
#include "oscl_socket_types.h"
#include "oscl_scheduler_ao.h"
#include "oscl_dns.h"
#include "pvlogger.h"
```

### Data Structures

- class [OsclIDNSMethod](#)
- class [OsclDNSRequestAO](#)

## 7.16 oscl\_dns\_param.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_dns_tuneables.h"
#include "oscl_namestring.h"
#include "oscl_dns.h"
#include "oscl_mutex.h"
#include "oscl_semaphore.h"
```

### Data Structures

- class [DNSRequestParam](#)
- class [GetHostByNameParam](#)

### Typedefs

- typedef [OsclMemAllocator TDNSRequestParamAllocator](#)

#### 7.16.1 Typedef Documentation

##### 7.16.1.1 typedef [OsclMemAllocator TDNSRequestParamAllocator](#)

## 7.17 oscl\_dns\_request.h File Reference

```
#include "oscl_dns_tuneables.h"
#include "oscl_namestring.h"
#include "oscl_dns.h"
#include "oscl_socket_types.h"
```

### Data Structures

- class [OsclDNSRequest](#)

## 7.18 oscl\_dns\_tuneables.h File Reference

```
#include "osclconfig_io.h"
#include "osclconfig_proc.h"
```

### Defines

- #define PV\_DNS\_SERVER 1
- #define PV\_DNS\_IS\_THREAD OSCL\_HAS\_THREAD\_SUPPORT

#### 7.18.1 Define Documentation

##### 7.18.1.1 #define PV\_DNS\_IS\_THREAD OSCL\_HAS\_THREAD\_SUPPORT

PV\_DNS\_IS\_THREAD chooses either the threaded or AO-based implementation of the PV DNS request.  
Note: AO-based option is not good here, since some DNS requests will block the caller until completion.

##### 7.18.1.2 #define PV\_DNS\_SERVER 1

Enable/disable the PV DNS server here.

## 7.19 oscl\_double\_list.h File Reference

Internal use types for scheduler.

```
#include "osclconfig_proc.h"
#include "oscl_base.h"
#include "oscl_assert.h"
#include "oscl_double_list.inl"
```

### Data Structures

- class [OsclDoubleLink](#)
- class [OsclDoubleList](#)
- class [OsclDoubleListBase](#)
- class [OsclDoubleRunner](#)
- class [OsclPriorityLink](#)
- class [OsclPriorityList](#)

### Defines

- #define [QUE\\_ITER\\_BEGIN](#)(\_type, \_qname)
- #define [QUE\\_ITER\\_END](#)(\_qname)

### Functions

- template<class T, class S> T \* [OsclPtrAdd](#) (T \*aPtr, S aVal)
- template<class T, class S> T \* [OsclPtrSub](#) (T \*aPtr, S aVal)

#### 7.19.1 Detailed Description

Internal use types for scheduler.

## 7.20 oscl\_errno.h File Reference

Defines functions to access additional information on errors where supported through an errno or similar service.

```
#include "oscl_base.h"
#include "osclconfig_error.h"
#include "oscl_errno.inl"
```

### Functions

- OSCL\_IMPORT\_REF bool [OSCL\\_IsErrnoSupported \(\)](#)  
*This function determines if a particular system saves the error number that occurs on a system call.*
- OSCL\_IMPORT\_REF int [OSCL\\_GetLastError \(\)](#)  
*This function returns the value of the system's global error number variable.*
- OSCL\_IMPORT\_REF bool [OSCL\\_SetLastError \(int newVal\)](#)  
*This function sets the last error code for the system.*
- OSCL\_IMPORT\_REF char \* [OSCL\\_StrError \(int errnum\)](#)  
*This function maps an error number to an error-message string.*

### 7.20.1 Detailed Description

Defines functions to access additional information on errors where supported through an errno or similar service.

## 7.21 oscl\_error.h File Reference

OSCL Error trap and cleanup include file.

```
#include "oscl_heapbase.h"
#include "oscl_defalloc.h"
#include "oscl_error_codes.h"
#include "oscl_singleton.h"
#include "oscl_assert.h"
#include "oscl_tls.h"
```

### Data Structures

- class [OsclError](#)
- class [OsclErrorTrap](#)
- class [OsclTLSEx](#)
- class [OsclTLSRegistryEx](#)

### Defines

- #define [OSCL\\_TRAPSTACK\\_PUSH\(a\)](#) OsclError::PushL(a)
- #define [OSCL\\_TRAPSTACK\\_POP\(\)](#) OsclError::Pop()
- #define [OSCL\\_TRAPSTACK\\_POPDEALLOC\(\)](#) OsclError::PopDealloc()

#### 7.21.1 Detailed Description

OSCL Error trap and cleanup include file.

## 7.22 oscl\_error\_allocator.h File Reference

Defines a memory allocation class used by the oscl error layer.

```
#include "oscl_base.h"  
#include "oscl_base_macros.h"  
#include "osclconfig_error.h"  
#include "oscl_assert.h"  
#include "oscl_defalloc.h"
```

### Data Structures

- class [OsclErrorAllocator](#)

*This class provides static methods to invoke the user defined memory allocation routines.*

#### 7.22.1 Detailed Description

Defines a memory allocation class used by the oscl error layer.

## 7.23 oscl\_error\_codes.h File Reference

Defines basic error and leave codes.

### Defines

- #define `OsclErrNone` 0
- #define `OsclErrGeneral` 100
- #define `OsclErrNoMemory` 101
- #define `OsclErrCancelled` 102
- #define `OsclErrNotSupported` 103
- #define `OsclErrArgument` 104
- #define `OsclErrBadHandle` 105
- #define `OsclErrAlreadyExists` 106
- #define `OsclErrBusy` 107
- #define `OsclErrNotReady` 108
- #define `OsclErrCorrupt` 109
- #define `OsclErrTimeout` 110
- #define `OsclErrOverflow` 111
- #define `OsclErrUnderflow` 112
- #define `OsclErrInvalidState` 113
- #define `OsclErrNoResources` 114
- #define `OsclErrNotInstalled` 115
- #define `OsclErrAlreadyInstalled` 116
- #define `OsclErrSystemCallFailed` 117
- #define `OsclErrNoHandler` 118
- #define `OsclErrThreadContextIncorrect` 119
- #define `OSCL_ERR_NONE` `OsclErrNone`
- #define `OSCL_BAD_ALLOC_EXCEPTION_CODE` `OsclErrNoMemory`
- #define `OsclSuccess` 0
- #define `OsclPending` 1
- #define `OsclFailure` -1

### Typedefs

- typedef int32 `OsclLeaveCode`
- typedef int32 `OsclReturnCode`

### 7.23.1 Detailed Description

Defines basic error and leave codes.

## 7.24 oscl\_error\_imp.h File Reference

Internal error implementation support.

```
#include "osclconfig_error.h"  
#include "oscl_error_imp_jumps.h"
```

### Defines

- #define PVERROR\_IMP\_JUMPS

#### 7.24.1 Detailed Description

Internal error implementation support.

## 7.25 oscl\_error\_imp\_cppexceptions.h File Reference

Implementation File for Leave using C++ exceptions.

```
#include "oscl_error_trapcleanup.h"
```

### Data Structures

- class [internalLeave](#)

### Defines

- #define [PVError\\_DoLeave\(\)](#) [internalLeave](#) \_\_ilv;\_\_ilv.a=0;throw(\_\_ilv)
- #define [\\_PV\\_TRAP](#)(\_\_r, \_\_s)
- #define [\\_PV\\_TRAP\\_NO\\_TLS](#)(\_\_trapimp, \_\_r, \_\_s)

### 7.25.1 Detailed Description

Implementation File for Leave using C++ exceptions.

## 7.26 oscl\_error\_imp\_fatalerror.h File Reference

Implementation File for Leave using system fatal error.

```
#include "oscl_assert.h"
```

### Defines

- #define PVError\_DoLeave() \_OSCL\_Abort()
- #define \_PV\_TRAP(\_\_r, \_\_s)
- #define \_PV\_TRAP\_NO\_TLS(\_\_tr, \_\_r, \_\_s)

#### 7.26.1 Detailed Description

Implementation File for Leave using system fatal error.

#### 7.26.2 Define Documentation

##### 7.26.2.1 #define \_PV\_TRAP(\_\_r, \_\_s)

**Value:**

```
__r=OsclErrNone; \
{__s;}
```

##### 7.26.2.2 #define \_PV\_TRAP\_NO\_TLS(\_\_tr, \_\_r, \_\_s)

**Value:**

```
__r=OsclErrNone; \
{__s;}
```

##### 7.26.2.3 #define PVError\_DoLeave() \_OSCL\_Abort()

## 7.27 oscl\_error\_imp\_jumps.h File Reference

Implementation of using Setjmp / Longjmp.

```
#include "oscl_error_trapcleanup.h"
#include "oscl_assert.h"
#include "osclconfig_error.h"
#include "oscl_defalloc.h"
#include "oscl_error.h"
```

### Data Structures

- class [OsclJump](#)

### Defines

- #define OSCL\_JUMP\_MAX\_JUMP\_MARKS OSCL\_MAX\_TRAP\_LEVELS
- #define internalLeave (-1)
- #define PVError\_DoLeave() OsclJump::StaticJump(internalLeave)
- #define \_PV\_TRAP(\_\_r, \_\_s)
- #define \_PV\_TRAP\_NO\_TLS(\_\_trapimp, \_\_r, \_\_s)

#### 7.27.1 Detailed Description

Implementation of using Setjmp / Longjmp.

#### 7.27.2 Define Documentation

##### 7.27.2.1 #define \_PV\_TRAP(\_\_r, \_\_s)

**Value:**

```
__r=OsclErrNone; \
{ \
    OsclErrorTrapImp* __trap=OsclErrorTrapImp::Trap(); \
    if(!__trap){__s;}else{ \
        int __tr=setjmp(*(__trap->iJumpData->Top())); \
        if (__tr==0)\ 
            {__s;} \
        else if (__tr==internalLeave)\ 
            {__r=__trap->iLeave;} \
            __trap->UnTrap();} \
}
```

##### 7.27.2.2 #define \_PV\_TRAP\_NO\_TLS(\_\_trapimp, \_\_r, \_\_s)

**Value:**

```
__r=OsclErrNone; \
{ \
    OsclErrorTrapImp* __trap=OsclErrorTrapImp::TrapNoTls(__trapimp); \
    if(!__trap){__s;}else{ \
        int __tr=setjmp(*((__trap->iJumpData->Top()))); \
        if (__tr==0)\ 
            {__s;} \
        else if (__tr==internalLeave)\ 
            {__r=__trap->iLeave;} \
        __trap->UnTrap();} \
}
```

#### **7.27.2.3 #define PVError\_DoLeave() OsclJump::StaticJump(internalLeave)**

## 7.28 oscl\_error\_trapcleanup.h File Reference

OSCL Error trap and cleanup implementation include file.

```
#include "osclconfig_error.h"
#include "oscl_heapbase.h"
#include "oscl_defalloc.h"
#include "oscl_assert.h"
#include "oscl_error.h"
#include "oscl_base_alloc.h"
#include "oscl_tls.h"
#include "oscl_singleton.h"
#include "oscl_error_imp.h"
```

### Data Structures

- class [OsclErrorTrapImp](#)
- class [OsclTrapStack](#)
- class [OsclTrapStackItem](#)

### Defines

- #define [OSCL\\_MAX\\_TRAP\\_LEVELS](#) 20
- #define [PVERRORTRAP\\_REGISTRY\\_ID](#) [OSCL\\_TLS\\_ID\\_PVERRORTRAP](#)
- #define [PVERRORTRAP\\_REGISTRY](#) [OsclTLSRegistry](#)

#### 7.28.1 Detailed Description

OSCL Error trap and cleanup implementation include file.

## 7.29 oscl\_exception.h File Reference

contains all the exception handling macros and classes

```
#include "oscl_error.h"
#include "oscl_error_imp.h"
```

### Data Structures

- class **OsclException**

*oscl\_exception.h contains all the exception handling macros and classes This template class provides the base exception class that all exceptions derive from*

### Defines

- #define **OSCL\_LEAVE(\_leave\_status)** OsclError::Leave(\_leave\_status)  
*Use this macro to cause a Leave. It terminates the execution of the current active function.*
- #define **OSCL\_TRY(\_leave\_status, \_statements)** \_PV\_TRAP(\_leave\_status,\_statements)  
*This macro will be used to set up a try block.*
- #define **OSCL\_TRY\_NO\_TLS(\_trapimp, \_leave\_status, \_statements)** \_PV\_TRAP\_NO\_TLS(\_-  
 $_trapimp, _leave_status, _statements)$   
• #define **OSCL\_FIRST\_CATCH\_ANY(\_leave\_status, \_statements)** if (\_leave\_status!=OsclErrNone){ \_statements; }  
*This section defines the macros to be used in the catch block following the try block Use this macro to call a function that handles all exception types thrown in the preceding try block.*
- #define **OSCL\_FIRST\_CATCH(\_leave\_status, \_catch\_value, \_statements)** if (\_leave\_status!=OsclErrNone && \_leave\_status == \_catch\_value){\_statements;}  
*Use this macro to define a block of code that catches the first exception type thrown in the preceding try block.*
- #define **OSCL\_CATCH(\_leave\_status, \_catch\_value, \_statements)** else if (\_leave\_status!=OsclErrNone && \_leave\_status == \_catch\_value){\_statements;}  
*Use this macro to define a block of code for catching additional exception types.*
- #define **OSCL\_CATCH\_ANY(\_leave\_status, \_statements)** else if (\_leave\_status!=OsclErrNone){ \_-  
 $_statements;$   
*Use this macro to call a function that will catch all remaining exception types.*
- #define **OSCL\_LAST\_CATCH(\_leave\_status)** else if (\_leave\_status!=OsclErrNone){OSCL\_-  
 $LEAVE(_leave_status);$   
*Use this macro if OSCL\_CATCH\_ANY has not been used. It will mark the end of the catch block.*

### 7.29.1 Detailed Description

contains all the exception handling macros and classes

## 7.30 oscl\_exclusive\_ptr.h File Reference

This file defines the [OsclExclusivePtr](#) template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error.

```
#include "oscl_defalloc.h"
```

### Data Structures

- class [OsclExclusiveArrayPtr](#)

*The OsclExclusiveArrayPtr class is a template class that defines an array pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the OsclExclusiveArrayPtr expires, its destructor uses delete to free the memory.*

- class [OsclExclusivePtr](#)

*The OsclExclusivePtr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the OsclExclusivePtr expires, its destructor uses delete to free the memory.*

- class [OsclExclusivePtrA](#)

*The OsclExclusivePtrA class is a template class that defines any pointer like object intended to be assigned an address obtained (directly or or indirectly) through Alloc. When the OsclExclusivePtrA expires, Alloc is used to free the memory.*

### 7.30.1 Detailed Description

This file defines the [OsclExclusivePtr](#) template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error.

## 7.31 oscl\_file\_async\_read.h File Reference

```
#include "oscl_base.h"
#include "osclconfig_io.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_scheduler_ao.h"
#include "oscl_file_io.h"
#include "oscl_semaphore.h"
```

### Data Structures

- class [OsclAsyncFile](#)
- class [OsclAsyncFileBuffer](#)
- class [OsclBuf](#)
- class [OsclPtr](#)
- class [OsclPtrC](#)

## 7.32 oscl\_file\_cache.h File Reference

The file [oscl\\_file\\_cache.h](#) defines the class [OsclFileCache](#).

```
#include "osclconfig_io.h"  
#include "oscl_base.h"  
#include "oscl_file_io.h"
```

### Data Structures

- class [OsclFileCache](#)
- class [OsclFileCacheBuffer](#)

#### 7.32.1 Detailed Description

The file [oscl\\_file\\_cache.h](#) defines the class [OsclFileCache](#).

## 7.33 oscl\_file\_dir\_utils.h File Reference

The file `oscl_file_dir_utils.h` defines some unix-style directory ops.

```
#include "osclconfig_io.h"
#include "oscl_base.h"
```

### Data Structures

- struct `oscl_fsstat`
- struct `oscl_stat_buf`

### Typedefs

- typedef `oscl_fsstat` `OSCL_FSSTAT`
- typedef `oscl_stat_buf` `OSCL_STAT_BUF`

### Enumerations

- enum `OSCL_FILEMGMT_PERMS` { `OSCL_FILEMGMT_PERMS_READ` = 0x1, `OSCL_FILEMGMT_PERMS_WRITE` = 0x2, `OSCL_FILEMGMT_PERMS_EXECUTE` = 0x4 }
- enum `OSCL_FILEMGMT_MODES` { `OSCL_FILEMGMT_MODE_DIR` = 0x1 }
- enum `OSCL_FILEMGMT_ERR_TYPE` { `OSCL_FILEMGMT_E_OK` = 0, `OSCL_FILEMGMT_E_PATH_TOO_LONG`, `OSCL_FILEMGMT_E_PATH_NOT_FOUND`, `OSCL_FILEMGMT_E_ALREADY_EXISTS`, `OSCL_FILEMGMT_E_NOT_EMPTY`, `OSCL_FILEMGMT_E_PERMISSION_DENIED`, `OSCL_FILEMGMT_E_NO_MATCH`, `OSCL_FILEMGMT_E_UNKNOWN`, `OSCL_FILEMGMT_E_SYS_SPECIFIC`, `OSCL_FILEMGMT_E_NOT_IMPLEMENTED` }

### Functions

- `OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_getcwd (oscl_wchar *path, uint32 size)`
- `OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_getcwd (char *path, uint32 size)`
- `OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_stat (const oscl_wchar *path, OSCL_STAT_BUF *statbuf)`
- `OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_stat (const char *path, OSCL_STAT_BUF *statbuf)`
- `OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_mkdir (const oscl_wchar *path)`
- `OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_mkdir (const char *path)`
- `OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_rmdir (const oscl_wchar *path)`
- `OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_rmdir (const char *path)`
- `OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_chdir (const oscl_wchar *path)`
- `OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_chdir (const char *path)`
- `OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_rename (const oscl_wchar *oldpath, const oscl_wchar *newpath)`
- `OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_rename (const char *oldpath, const char *newpath)`

- OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_statfs (OSCL\_FSSTAT \*stats, const char \*path)
- OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_statfs (OSCL\_FSSTAT \*stats, const oscl\_wchar \*path)

### 7.33.1 Detailed Description

The file [oscl\\_file\\_dir\\_utils.h](#) defines some unix-style directory ops.

## 7.34 oscl\_file\_find.h File Reference

The file [oscl\\_file\\_find.h](#) defines the class [Oscl\\_FileFind](#).

```
#include "osclconfig_io.h"  
#include "oscl_base.h"  
#include "oscl_mem.h"  
#include "oscl_vector.h"  
#include "oscl_string_containers.h"  
#include "oscl_file_types.h"
```

### Data Structures

- class [Oscl\\_FileFind](#)

#### 7.34.1 Detailed Description

The file [oscl\\_file\\_find.h](#) defines the class [Oscl\\_FileFind](#).

## 7.35 oscl\_file\_handle.h File Reference

The file [oscl\\_file\\_handle.h](#) defines the class [OsclFileHandle](#).

```
#include "osclconfig_io.h"  
#include "oscl_base.h"
```

### Data Structures

- class [OsclFileHandle](#)

### TypeDefs

- [typedef FILE \\* TOsclFileHandle](#)

#### 7.35.1 Detailed Description

The file [oscl\\_file\\_handle.h](#) defines the class [OsclFileHandle](#).

## 7.36 oscl\_file\_io.h File Reference

The file [oscl\\_file\\_io.h](#) defines the class [Oscl\\_File](#). This is the public API to the basic file I/O operations.

```
#include "osclconfig_io.h"  
#include "oscl_base.h"  
#include "oscl_mem.h"  
#include "oscl_vector.h"  
#include "oscl_file_server.h"  
#include "oscl_file_find.h"  
#include "oscl_file_dir_utils.h"  
#include "oscl_file_handle.h"
```

### Data Structures

- class [Oscl\\_File](#)
- class [OsclFixedCacheParam](#)
- class [OsclCacheObserver](#)

### Defines

- #define [TOsclFileOffsetInt32](#) int32

#### 7.36.1 Detailed Description

The file [oscl\\_file\\_io.h](#) defines the class [Oscl\\_File](#). This is the public API to the basic file I/O operations.

## 7.37 oscl\_file\_native.h File Reference

The file [oscl\\_file\\_native.h](#) defines the class [OsclNativeFile](#). This is the porting layer for basic file I/O operations.

```
#include "osclconfig_io.h"
#include "oscl_base.h"
#include "oscl_aostatus.h"
#include "oscl_file_io.h"
```

### Data Structures

- class [OsclNativeFile](#)

#### 7.37.1 Detailed Description

The file [oscl\\_file\\_native.h](#) defines the class [OsclNativeFile](#). This is the porting layer for basic file I/O operations.

## 7.38 oscl\_file\_server.h File Reference

The file [oscl\\_file\\_server.h](#) defines the class [Oscl\\_FileServer](#). This is the porting layer for file server implementations.

```
#include "osclconfig_io.h"  
#include "oscl_base.h"
```

### Data Structures

- class [Oscl\\_FileServer](#)

#### 7.38.1 Detailed Description

The file [oscl\\_file\\_server.h](#) defines the class [Oscl\\_FileServer](#). This is the porting layer for file server implementations.

## 7.39 oscl\_file\_stats.h File Reference

File stats class.

```
#include "oscl_base.h"
#include "osclconfig_io.h"
```

### Data Structures

- class [OsclFileStats](#)
- class [OsclFileStatsItem](#)

### Defines

- #define [OSCL\\_FILE\\_STATS\\_LOGGER\\_NODE](#) "OsclFileStats"

### Enumerations

- enum [TOsclFileOp](#) { [EOsclFileOp\\_Open](#), [EOsclFileOp\\_Close](#), [EOsclFileOp\\_Read](#), [EOsclFileOp\\_Write](#), [EOsclFileOp\\_Seek](#), [EOsclFileOp\\_Tell](#), [EOsclFileOp\\_Size](#), [EOsclFileOp\\_Flush](#), [EOsclFileOp\\_EndOfFile](#), [EOsclFileOp\\_NativeOpen](#), [EOsclFileOp\\_NativeClose](#), [EOsclFileOp\\_NativeRead](#), [EOsclFileOp\\_NativeWrite](#), [EOsclFileOp\\_NativeSeek](#), [EOsclFileOp\\_NativeTell](#), [EOsclFileOp\\_NativeSize](#), [EOsclFileOp\\_NativeFlush](#), [EOsclFileOp\\_NativeEndOfFile](#), [EOsclFileOp\\_Last](#) }

### 7.39.1 Detailed Description

File stats class.

## 7.40 oscl\_file\_types.h File Reference

The file [oscl\\_file\\_types.h](#) defines some constants and types for file I/O implementations. Anything that needs to be shared across implementation modules can go here.

### Data Structures

- class [OsclNativeFileParams](#)

### Defines

- #define [OSCL\\_IO\\_FILENAME\\_MAXLEN](#) 512
- #define [OSCL\\_IO\\_EXTENSION\\_MAXLEN](#) 512
- #define [OSCL\\_FILE\\_WCHAR\\_PATH\\_DELIMITER](#) \_STRLIT("/")
- #define [OSCL\\_FILE\\_CHAR\\_PATH\\_DELIMITER](#) \_STRLIT\_CHAR("/")

### 7.40.1 Detailed Description

The file [oscl\\_file\\_types.h](#) defines some constants and types for file I/O implementations. Anything that needs to be shared across implementation modules can go here.

## 7.41 oscl\_heapbase.h File Reference

OSCL Heap Base include file.

```
#include "osclconfig_error.h"
#include "oscl_base.h"
#include "oscl_heapbase.inl"
```

### Data Structures

- class [\\_OsclHeapBase](#)
- class [OsclTrapItem](#)

### Typedefs

- [typedef void\(\\* OsclTrapOperation \)\(OsclAny \\*\)](#)

#### 7.41.1 Detailed Description

OSCL Heap Base include file.

## 7.42 oscl\_init.h File Reference

Global oscl initialization.

```
#include "oscl_base.h"  
#include <stdio.h>
```

### Data Structures

- class [OsclInit](#)
- class [OsclSelect](#)

#### 7.42.1 Detailed Description

Global oscl initialization.

## 7.43 oscl\_int64\_utils.h File Reference

```
#include "oscl_base.h"
```

### Data Structures

- class [Oscl\\_Int64\\_Utils](#)  
*The Oscl\_Int64\_Utils class provides a wrapper for commonly used int64/uint64 operations.*
- struct [OsclInteger64Transport](#)

### Typedefs

- typedef [OsclInteger64Transport \\_OsclInteger64Transport](#)

#### 7.43.1 Typedef Documentation

##### 7.43.1.1 typedef struct [OsclInteger64Transport \\_OsclInteger64Transport](#)

###### [OsclInteger64Transport](#) Structure

Structure to only transport 64-bit integer values uint64 and int64 could be classes so needed for cases where having a class will not work.

## 7.44 oscl\_ip\_socket.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
```

### Data Structures

- class [OsclIPSocketI](#)

## 7.45 oscl\_linked\_list.h File Reference

The file [oscl\\_linked\\_list.h](#) defines the template class [Oscl\\_Linked\\_List](#) which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

```
#include "oscl_base.h"
#include "oscl_defalloc.h"
#include "oscl_opaque_type.h"
#include "oscl_assert.h"
```

### Data Structures

- class [LinkedListElement](#)
- class [Oscl\\_Linked\\_List](#)
- class [Oscl\\_Linked\\_List\\_Base](#)
- class [Oscl\\_MTLinked\\_List](#)

#### 7.45.1 Detailed Description

The file [oscl\\_linked\\_list.h](#) defines the template class [Oscl\\_Linked\\_List](#) which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

## 7.46 oscl\_lock\_base.h File Reference

This file defines an abstract lock class, [OsclLockBase](#), that is used for APIs potentially requiring multi-thread safety. A null-lock implementation, [OsclNullLock](#), is also provided for single-thread configurations (basically a noop for lock/unlock). Also provides the [OsclScopedLock](#) class which is template class takes care of freeing the lock when the class goes out of scope.

### Data Structures

- class [OsclLockBase](#)
- class [OsclNullLock](#)
- class [OsclScopedLock](#)

*The OsclScopedLock class is a template class that handles unlocking an abstract class on destruction. This is very useful for ensuring that the lock is released when the OsclScopedLock goes out of scope.*

#### 7.46.1 Detailed Description

This file defines an abstract lock class, [OsclLockBase](#), that is used for APIs potentially requiring multi-thread safety. A null-lock implementation, [OsclNullLock](#), is also provided for single-thread configurations (basically a noop for lock/unlock). Also provides the [OsclScopedLock](#) class which is template class takes care of freeing the lock when the class goes out of scope.

## 7.47 oscl\_map.h File Reference

The file `oscl_map.h` defines the template class `Oscl_Map` which has a very similar API as the STL Map class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

```
#include "oscl_base.h"
#include "oscl_tree.h"
#include "osclconfig_compiler_warnings.h"
```

### Data Structures

- struct `Oscl_Less`
- class `Oscl_Map`
- struct `Oscl_Select1st`
- class `value_compare`

### Defines

- `#define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE`

#### 7.47.1 Detailed Description

The file `oscl_map.h` defines the template class `Oscl_Map` which has a very similar API as the STL Map class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

#### 7.47.2 Define Documentation

##### 7.47.2.1 `#define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE`

## 7.48 oscl\_math.h File Reference

Provides math functions.

```
#include "osclconfig_util.h"
#include "oscl_base.h"
#include "oscl_math.inl"
```

### Functions

- OSCL\_COND\_IMPORT\_REF double `oscl_log` (double value)
- OSCL\_COND\_IMPORT\_REF double `oscl_log10` (double value)
- OSCL\_COND\_IMPORT\_REF double `oscl_sqrt` (double value)
- OSCL\_COND\_IMPORT\_REF double `oscl_pow` (double x, double y)
- OSCL\_COND\_IMPORT\_REF double `oscl_exp` (double value)
- OSCL\_COND\_IMPORT\_REF double `oscl_sin` (double value)
- OSCL\_COND\_IMPORT\_REF double `oscl_cos` (double value)
- OSCL\_COND\_IMPORT\_REF double `oscl_tan` (double value)
- OSCL\_COND\_IMPORT\_REF double `oscl_asin` (double value)
- OSCL\_COND\_IMPORT\_REF double `oscl_atan` (double value)
- OSCL\_COND\_IMPORT\_REF double `oscl_floor` (double value)

### 7.48.1 Detailed Description

Provides math functions.

## 7.49 oscl\_media\_data.h File Reference

Defines a container class for media data made up of a collection of memory fragments.

```
#include "oscl_base.h"  
#include "oscl_mem_basic_functions.h"  
#include "oscl_media_status.h"
```

### Data Structures

- class [BufferFragment](#)
- class [BufferMgr](#)
- class [BufferState](#)
- class [BuffFragGroup](#)
- class [MediaData](#)
- class [MemAllocator](#)

### Typedefs

- typedef void(\* [BufferFreeFuncPtr](#) )(void \*)
- typedef uint32 [MediaTimestamp](#)

#### 7.49.1 Detailed Description

Defines a container class for media data made up of a collection of memory fragments.

## 7.50 oscl\_media\_status.h File Reference

Defines a status values for the [MediaData](#) containers.

### Data Structures

- class [BufFragStatusClass](#)
- class [MediaStatusClass](#)

### Variables

- const int32 [APPEND\\_MEDIA\\_AT\\_END](#) = -1

#### 7.50.1 Detailed Description

Defines a status values for the [MediaData](#) containers.

## 7.51 oscl\_mem.h File Reference

This file contains basic memory definitions for common use across platforms.

```
#include "osclconfig_memory.h"
#include "oscl_base.h"
#include "oscl_types.h"
#include "oscl_assert.h"
#include "oscl_mem_basic_functions.h"
#include "oscl_lock_base.h"
#include "osclconfig_compiler_warnings.h"
#include "oscl_mem_inst.h"
#include "oscl_heapbase.h"
#include "oscl_defalloc.h"
#include "oscl_refcounter.h"
#include "oscl_error.h"
#include "oscl_exception.h"
#include "oscl_mem.inl"
```

### Data Structures

- class [HeapBase](#)
- class [OsclAuditCB](#)
- class [OsclMem](#)
- class [OsclMemAllocator](#)
- class [OsclMemAllocDestructDealloc](#)
- class [OsclMemBasicAllocator](#)
- class [OsclMemBasicAllocDestructDealloc](#)
- class [OsclMemGlobalAuditObject](#)

### Defines

- #define [OSCL\\_DISABLE\\_WARNING\\_TRUNCATE\\_DEBUG\\_MESSAGE](#)
- #define [OSCL\\_HAS\\_GLOBAL\\_NEW\\_DELETE](#) 1
- #define [OSCL\\_CLEANUP\\_BASE\\_CLASS\(T\)](#) \_OSCL\_CLEANUP\_BASE\_CLASS(T)
- #define [OSCL\\_ALLOC\\_NEW\(T\\_allocator, T, params\)](#) new(T\_allocator.allocate(1)) T params
- #define [OSCL\\_TRAP\\_ALLOC\\_NEW\(T\\_ptr, T\\_allocator, T, params\)](#) \_OSCL\_TRAP\_NEW(T\_allocator.allocate(1),T\_allocator.deallocate,T\_ptr,T,params)
- #define [OSCL\\_ALLOC\\_DELETE\(ptr, T\\_allocator, T\)](#)
- #define [OSCL\\_MALLOC\(count\)](#) \_oscl\_default\_audit\_malloc(count)
- #define [oscl\\_malloc\(a\)](#) OSCL\_MALLOC(a)
- #define [OSCL\\_DEFAULT\\_MALLOC\(x\)](#) OSCL\_MALLOC(x)
- #define [OSCL\\_AUDIT\\_MALLOC\(auditCB, count\)](#) \_oscl\_audit\_malloc(count, auditCB)
- #define [OSCL\\_CALLOC\(num, size\)](#) \_oscl\_default\_audit\_calloc(num,size)
- #define [oscl\\_calloc\(a, b\)](#) OSCL\_CALLOC(a,b)

- #define **OSCL\_AUDIT\_CALLOC**(auditCB, num, size) \_oscl\_audit\_calloc(num,size, auditCB)
- #define **OSCL\_REALLOC**(ptr, new\_size) \_oscl\_default\_audit\_realloc(ptr,new\_size)
- #define **oscl\_realloc**(a, b) OSCL\_REALLOC(a,b)
- #define **OSCL\_AUDIT\_REALLOC**(auditCB, ptr, new\_size) \_oscl\_audit\_realloc(ptr,new\_size, auditCB)
- #define **OSCL\_FREE**(ptr) \_oscl\_audit\_free(ptr)
- #define **oscl\_free**(x) OSCL\_FREE(x)
- #define **OSCL\_DEFAULT\_FREE**(x) OSCL\_FREE(x)
- #define **OSCL\_NEW**(T, params) new T params
- #define **OSCL\_PLACEMENT\_NEW**(ptr, constructor) new(ptr) constructor
- #define **OSCL\_TRAP\_NEW**(T\_ptr, T, params) \_OSCL\_TRAP\_NEW(\_oscl\_default\_audit\_new(sizeof(T)),\_oscl\_audit\_free,T\_ptr,T,params)
- #define **OSCL\_AUDIT\_NEW**(auditCB, T, params) new(\_oscl\_audit\_new(sizeof(T),auditCB)) T params
- #define **OSCL\_TRAP\_AUDIT\_NEW**(T\_ptr, auditCB, T, params) \_OSCL\_TRAP\_NEW(\_oscl\_audit\_new(sizeof(T),auditCB),\_oscl\_audit\_free,T\_ptr,T,params)
- #define **OSCL\_DELETE**(ptr)
- #define **OSCL\_AUDIT\_ARRAY\_NEW**(auditCB, T, count) new(\_oscl\_audit\_new(sizeof(T)\*(count),auditCB)) T
- #define **OSCL\_ARRAY\_NEW**(T, count) new T[count]
- #define **OSCL\_ARRAY\_DELETE**(ptr) delete [] ptr
- #define **OSCL\_DISABLE\_WARNING\_TRUNCATE\_DEBUG\_MESSAGE**
- #define **\_OSCL\_TRAP\_NEW**(exp, freeFunc, T\_ptr, T, params)
- #define **\_OSCL\_CLEANUP\_BASE\_CLASS**(T) this → T::~T()

## Functions

- **OSCL\_COND\_IMPORT\_REF** **uint** **oscl\_mem\_aligned\_size** (**uint** **size**)
- **OSCL\_IMPORT\_REF** **void** **OsclMemInit** (**OsclAuditCB** &auditCB)
- **OSCL\_IMPORT\_REF** **void \*** **\_oscl\_audit\_malloc** (**size\_t**, **OsclAuditCB** &, **const char** \*f=NULL, **const int** l=0)
- **OSCL\_IMPORT\_REF** **void \*** **\_oscl\_audit\_calloc** (**size\_t**, **size\_t**, **OsclAuditCB** &, **const char** \*f=NULL, **const int** l=0)
- **OSCL\_IMPORT\_REF** **void \*** **\_oscl\_audit\_realloc** (**void** \*, **size\_t**, **OsclAuditCB** &, **const char** \*f=NULL, **const int** l=0)
- **OSCL\_IMPORT\_REF** **void \*** **\_oscl\_audit\_new** (**size\_t**, **OsclAuditCB** &, **const char** \*f=NULL, **const int** l=0)
- **OSCL\_IMPORT\_REF** **void \*** **\_oscl\_default\_audit\_malloc** (**size\_t**, **const char** \*f=NULL, **const int** l=0)
- **OSCL\_IMPORT\_REF** **void \*** **\_oscl\_default\_audit\_calloc** (**size\_t**, **size\_t**, **const char** \*f=NULL, **const int** l=0)
- **OSCL\_IMPORT\_REF** **void \*** **\_oscl\_default\_audit\_realloc** (**void** \*, **size\_t**, **const char** \*f=NULL, **const int** l=0)
- **OSCL\_IMPORT\_REF** **void \*** **\_oscl\_default\_audit\_new** (**size\_t**, **const char** \*f=NULL, **const int** l=0)
- **OSCL\_IMPORT\_REF** **void** **\_oscl\_audit\_free** (**void** \*)
- **void \*** **operator new** (**size\_t** aSize, **const char** \*aFile, **int** aLine)
- **void \*** **operator new** (**size\_t** aSize)
- **void operator delete** (**void** \*aPtr)
- **void \*** **operator new[]** (**size\_t** aSize, **const char** \*aFile, **int** aLine)
- **void \*** **operator new[]** (**size\_t** aSize)
- **void operator delete[]** (**void** \*aPtr)

### 7.51.1 Detailed Description

This file contains basic memory definitions for common use across platforms.

This is the main entry point header file for the OSCL memory library. It should be the only one users directly include. Basic memory copy, compare, and move functions are defined here as well as the allocation functions.

### 7.51.2 Define Documentation

#### 7.51.2.1 #define OSCL\_DISABLE\_WARNING\_TRUNCATE\_DEBUG\_MESSAGE

Previously this was in oscl\_mem\_imp.h

### 7.51.3 Function Documentation

#### 7.51.3.1 void operator delete (void \* *aPtr*) [inline]

#### 7.51.3.2 void\* operator new (size\_t *aSize*) [inline]

## **7.52 oscl\_mem\_align.h File Reference**

## 7.53 oscl\_mem\_audit.h File Reference

This file contains the definition and partial implementation of MM\_Audit class.

```
#include "oscl_lock_base.h"
#include "oscl_base_alloc.h"
#include "oscl_tagtree.h"
#include "oscl_mem.h"
#include "oscl_mem_auto_ptr.h"
#include "osclconfig_compiler_warnings.h"
```

### Data Structures

- struct [MM\\_AllocInfo](#)
- struct [MM\\_AllocNode](#)
- struct [MM\\_AllocQueryInfo](#)
- class [MM\\_Audit\\_Imp](#)
- struct [MM\\_AuditOverheadStats](#)
- struct [MM\\_FailInsertParam](#)
- struct [MM\\_Stats\\_CB](#)
- struct [MM\\_Stats\\_t](#)
- class [OsclMemAudit](#)
- class [OsclMemStatsNode](#)

### Defines

- #define [OSCL\\_DISABLE\\_WARNING\\_TRUNCATE\\_DEBUG\\_MESSAGE](#)
- #define [MM\\_ALLOC\\_MAX\\_QUERY\\_FILENAME\\_LEN](#) 128
- #define [MM\\_ALLOC\\_MAX\\_QUERY\\_TAG\\_LEN](#) 64
- #define [MM\\_AUDIT\\_VALIDATE\\_BLOCK](#) 1
- #define [MM\\_AUDIT\\_PREFILL\\_FLAG](#) 0x1
- #define [MM\\_AUDIT\\_POSTFILL\\_FLAG](#) 0x2
- #define [MM\\_AUDIT\\_VALIDATE\\_ALL\\_HEAP\\_FLAG](#) 0x4
- #define [MM\\_AUDIT\\_VALIDATE\\_ON\\_FREE\\_FLAG](#) 0x8
- #define [MM\\_AUDIT\\_ALLOC\\_NODE\\_ENABLE\\_FLAG](#) 0x10
- #define [MM\\_AUDIT\\_SUPPRESS\\_FILENAME\\_FLAG](#) 0x20
- #define [DEFAULT\\_MM\\_AUDIT\\_MODE](#) 0

### Typedefs

- typedef [OSCLMemAutoPtr< char, Oscl\\_TAlloc< char, OsclMemBasicAllocator > >](#) [MMAudit\\_CharAutoPtr](#)
- typedef [OSCLMemAutoPtr< uint8, Oscl\\_TAlloc< uint8, \\_OsclBasicAllocator > >](#) [MMAudit\\_Uint8AutoPtr](#)
- typedef [OSCLMemAutoPtr< MM\\_AllocNode, Oscl\\_TAlloc< MM\\_AllocNode, OsclMemBasicAllocator > >](#) [MM\\_AllocNodeAutoPtr](#)
- typedef [OSCLMemAutoPtr< OsclMemStatsNode, Oscl\\_TAlloc< OsclMemStatsNode, OsclMemBasicAllocator > >](#) [MM\\_StatsNodeTagTreeType](#)

- `typedef OSCLMemAutoPtr< OsclMemStatsNode, Oscl_TAlloc< OsclMemStatsNode, OsclMemBasicAllocator > > OsclMemStatsNodeAutoPtr`
- `typedef Oscl_TAlloc< MM_StatsNodeTagTreeType, OsclMemBasicAllocator > TagTree_Allocator`
- `typedef Oscl_TagTree< MM_StatsNodeTagTreeType, TagTree_Allocator > OsclTagTreeType`

### 7.53.1 Detailed Description

This file contains the definition and partial implementation of MM\_Audit class.

### 7.53.2 Define Documentation

#### 7.53.2.1 #define OSCL\_DISABLE\_WARNING\_TRUNCATE\_DEBUG\_MESSAGE

## 7.54 oscl\_mem\_audit\_internals.h File Reference

This file contains the internal definitions for the mem audit library.

```
#include "oscl_base.h"
#include "oscl_mem_audit.h"
#include "oscl_mem_inst.h"
#include "osclconfig_compiler_warnings.h"
```

### Data Structures

- struct [MM\\_AllocBlockFence](#)
- struct [MM\\_AllocBlockHdr](#)

### Defines

- #define [OSCL\\_DISABLE\\_WARNING\\_TRUNCATE\\_DEBUG\\_MESSAGE](#)
- #define [MM\\_AUDIT\\_ALLOC\\_NODE\\_SUPPORT](#) 1
- #define [MM\\_AUDIT\\_FENCE\\_SUPPORT](#) 0
- #define [MM\\_AUDIT\\_INCLUDE\\_ALL\\_HEAP\\_VALIDATION](#) 1
- #define [MM\\_AUDIT\\_FILL\\_SUPPORT](#) 0
- #define [MM\\_AUDIT\\_FAILURE\\_SIMULATION\\_SUPPORT](#) 1
- #define [FENCE\\_PATTERN](#) 0xAA
- #define [MIN\\_FENCE\\_SIZE](#) 4
- #define [MEM\\_ALIGN\\_SIZE](#) 8
- #define [COMPUTE\\_MEM\\_ALIGN\\_SIZE](#)(x, y, z) (y+(((x+y)%z) ? (z - (x+y)%z) : 0))
- #define [DEFAULT\\_PREFILL\\_PATTERN](#) 0x96
- #define [DEFAULT\\_POSTFILL\\_PATTERN](#) 0x5A

### 7.54.1 Detailed Description

This file contains the internal definitions for the mem audit library.

### 7.54.2 Define Documentation

#### 7.54.2.1 #define OSCL\_DISABLE\_WARNING\_TRUNCATE\_DEBUG\_MESSAGE

## 7.55 oscl\_mem\_auto\_ptr.h File Reference

This file defines the oscl\_mem\_auto\_ptr template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error.

```
#include "osclconfig_memory.h"  
#include "osclconfig_compiler_warnings.h"  
#include "oscl_mem.h"
```

### Data Structures

- class [OSCLMemAutoPtr](#)

*The oscl\_auto\_ptr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the oscl\_auto\_ptr expires, its destructor uses delete to free the memory.*

### Defines

- #define [OSCL\\_DISABLE\\_WARNING\\_TRUNCATE\\_DEBUG\\_MESSAGE](#)
- #define [OSCL\\_DISABLE\\_WARNING\\_RETURN\\_TYPE\\_NOT\\_UDT](#)

#### 7.55.1 Detailed Description

This file defines the oscl\_mem\_auto\_ptr template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error.

#### 7.55.2 Define Documentation

##### 7.55.2.1 #define OSCL\_DISABLE\_WARNING\_TRUNCATE\_DEBUG\_MESSAGE

## 7.56 oscl\_mem\_basic\_functions.h File Reference

This file contains prototypes for the basic memory functions.

```
#include "oscl_base.h"  
#include "oscl_mem_basic_functions.inl"
```

### Functions

- OSCL\_COND\_IMPORT\_REF void \* [\\_oscl\\_malloc](#) (int32 count)
- OSCL\_COND\_IMPORT\_REF void \* [\\_oscl\\_calloc](#) (int32 nelems, int32 size)
- OSCL\_COND\_IMPORT\_REF void \* [\\_oscl\\_realloc](#) (void \*src, int32 count)
- OSCL\_COND\_IMPORT\_REF void [\\_oscl\\_free](#) (void \*src)
- OSCL\_COND\_IMPORT\_REF void \* [oscl\\_memcpy](#) (void \*dest, const void \*src, uint32 count)
- OSCL\_COND\_IMPORT\_REF void \* [oscl\\_memmove](#) (void \*dest, const void \*src, uint32 count)
- OSCL\_COND\_IMPORT\_REF void \* [oscl\\_memmove32](#) (void \*dest, const void \*src, uint32 count)
- OSCL\_COND\_IMPORT\_REF void \* [oscl\\_memset](#) (void \*dest, uint8 val, uint32 count)
- OSCL\_COND\_IMPORT\_REF int [oscl\\_memcmp](#) (const void \*buf1, const void \*buf2, uint32 count)

### 7.56.1 Detailed Description

This file contains prototypes for the basic memory functions.

## 7.57 oscl\_mem\_inst.h File Reference

The file defines default memory instrumentation level.

```
#include "osclconfig_memory.h"
```

### Defines

- #define **PVMEM\_INST\_LEVEL** 1

#### 7.57.1 Detailed Description

The file defines default memory instrumentation level.

## 7.58 oscl\_mem\_mempool.h File Reference

This file contains the definition of memory pool allocators.

```
#include "oscl_mem.h"  
#include "oscl_defalloc.h"  
#include "oscl_vector.h"
```

### Data Structures

- struct [MemPoolBlockInfo](#)
- struct [MemPoolBufferInfo](#)
- class [OsclMemPoolFixedChunkAllocator](#)
- class [OsclMemPoolFixedChunkAllocatorObserver](#)
- class [OsclMemPoolResizableAllocator](#)
- class [OsclMemPoolResizableAllocatorMemoryObserver](#)
- class [OsclMemPoolResizableAllocatorObserver](#)

#### 7.58.1 Detailed Description

This file contains the definition of memory pool allocators.

## 7.59 oscl\_mempool\_allocator.h File Reference

This file contains the definition of memory pool allocator for leave/trap.

```
#include "oscl_defalloc.h"
```

### Data Structures

- class [OsclMemPoolAllocator](#)

#### 7.59.1 Detailed Description

This file contains the definition of memory pool allocator for leave/trap.

## 7.60 oscl\_mutex.h File Reference

This file provides implementation of mutex.

```
#include "osclconfig_proc.h"  
#include "oscl_types.h"  
#include "oscl_base.h"  
#include "oscl_thread.h"  
#include "oscl_lock_base.h"
```

### Data Structures

- class [OsclMutex](#)
- class [OsclThreadLock](#)

### Typedefs

- typedef [OsclMutex OsclNoYieldMutex](#)

#### 7.60.1 Detailed Description

This file provides implementation of mutex.

#### 7.60.2 Typedef Documentation

##### 7.60.2.1 typedef [OsclMutex OsclNoYieldMutex](#)

Class [OsclNoYieldMutex](#) can be used in use cases where there will be no CPU-yielding operation done while the Mutex is locked.

CPU-yielding operations include [OsclMutex::Lock](#), [OsclSemaphore::Wait](#), [OsclThread::Sleep](#), and [OsclBrewThreadUtil::BThreadYield](#).

The behavior of [OsclNoYieldMutex](#) depends on whether the threading model is pre-emptive or not. When threading is pre-emptive, it is identical to [OsclMutex](#). When threading is non-pre-emptive, it is a NO-OP.

An example of this type of use case is for simple data protection.

## 7.61 oscl\_namestring.h File Reference

Name string class include file.

```
#include "oscl_base.h"
```

### Data Structures

- class [OsclNameString](#)

#### 7.61.1 Detailed Description

Name string class include file.

## 7.62 oscl\_opaque\_type.h File Reference

The file [oscl\\_opaque\\_type.h](#) defines pure virtual classes for working with opaque types.

```
#include "oscl_base.h"
```

### Data Structures

- class [Oscl\\_Opaque\\_Type\\_Alloc](#)
- class [Oscl\\_Opaque\\_Type\\_Alloc\\_LL](#)
- class [Oscl\\_Opaque\\_Type\\_Compare](#)

#### 7.62.1 Detailed Description

The file [oscl\\_opaque\\_type.h](#) defines pure virtual classes for working with opaque types.

## 7.63 oscl\_priqueue.h File Reference

Implements a priority queue data structure similar to STL.

```
#include "oscl_base.h"  
#include "oscl_vector.h"
```

### Data Structures

- class [OsclCompareLess](#)
- class [OsclPriorityQueue](#)
- class [OsclPriorityQueueBase](#)

#### 7.63.1 Detailed Description

Implements a priority queue data structure similar to STL.

Implements a priority queue data structure similar to the STL class. The properties of the class include O(Log<sub>2</sub>(N)) insertion and deletion complexity and O(1) complexity to access the top priority item.

## **7.64 oscl\_procstatus.h File Reference**

### **Data Structures**

- class [OsclProcStatus](#)

## 7.65 oscl\_queue.h File Reference

The file [oscl\\_queue.h](#) defines the template class [Oscl\\_Queue](#). It is similar to the STL::queue class, with some differences: - less complete - based on array rather than a deque - some interfaces modeled on oscl\_vector, for ease of transition Memory allocation is abstracted through the use of an allocator template parameter.

```
#include "oscl_base.h"  
#include "oscl_mem_basic_functions.h"  
#include "oscl_assert.h"  
#include "oscl_opaque_type.h"
```

### Data Structures

- class [Oscl\\_Queue](#)
- class [Oscl\\_Queue\\_Base](#)

#### 7.65.1 Detailed Description

The file [oscl\\_queue.h](#) defines the template class [Oscl\\_Queue](#). It is similar to the STL::queue class, with some differences: - less complete - based on array rather than a deque - some interfaces modeled on oscl\_vector, for ease of transition Memory allocation is abstracted through the use of an allocator template parameter.

## 7.66 oscl\_rand.h File Reference

Provides pseudo-random number generation.

```
#include "osclconfig_util.h"  
#include "oscl_base.h"  
#include "oscl_mem_basic_functions.h"  
#include "oscl_rand.inl"
```

### Data Structures

- class [OsclRand](#)

#### 7.66.1 Detailed Description

Provides pseudo-random number generation.

## 7.67 oscl\_refcounter.h File Reference

A general purpose reference counter to object lifetimes.

```
#include "oscl_assert.h"  
#include "oscl_defalloc.h"
```

### Data Structures

- class [Oscl\\_DefAllocWithRefCounter](#)
- class [OsclRefCounter](#)
- class [OsclRefCounterDA](#)
- class [OsclRefCounterMTDA](#)
- class [OsclRefCounterMTSA](#)
- class [OsclRefCounterSA](#)

#### 7.67.1 Detailed Description

A general purpose reference counter to object lifetimes.

## 7.68 oscl\_refcounter\_memfrag.h File Reference

This file provides the definition of reference counted memory fragment, which provides access to a buffer and helps manage its lifetime through the refcount.

```
#include "oscl_base.h"  
#include "oscl_refcounter.h"
```

### Data Structures

- class [OsclRefCounterMemFrag](#)

#### 7.68.1 Detailed Description

This file provides the definition of reference counted memory fragment, which provides access to a buffer and helps manage its lifetime through the refcount.

## 7.69 oscl\_registry\_access\_client.h File Reference

Client-side implementation Registry Access implementation.

```
#include "oscl_registry_types.h"
#include "oscl_string_containers.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
```

### Data Structures

- class [OsclRegistryAccessClient](#)

#### 7.69.1 Detailed Description

Client-side implementation Registry Access implementation.

## 7.70 oscl\_registry\_client.h File Reference

Client-side implementation of OsclRegistry.

```
#include "oscl_registry_types.h"
#include "oscl_mem.h"
#include "oscl_string.h"
```

### Data Structures

- class [OsclRegistryClient](#)

#### 7.70.1 Detailed Description

Client-side implementation of OsclRegistry.

## 7.71 oscl\_registry\_client\_impl.h File Reference

Client-side implementation of OsclRegistryInterface.

```
#include "oscl_base.h"
#include "osclconfig_proc.h"
#include "oscl_vector.h"
#include "oscl_string.h"
#include "oscl_registry_types.h"
#include "oscl_registry_serv_impl_tls.h"
```

### Data Structures

- class [OsclRegistryAccessClientImpl](#)
- class [OsclRegistryAccessClientTlsImpl](#)
- class [OsclRegistryClientImpl](#)
- class [OsclRegistryClientTlsImpl](#)

#### 7.71.1 Detailed Description

Client-side implementation of OsclRegistryInterface.

## 7.72 oscl\_registry\_serv\_impl.h File Reference

Server-side implementation of OsclRegistry interfaces.

```
#include "oscl_base.h"
#include "osclconfig_proc.h"
#include "oscl_registry_types.h"
#include "oscl_string.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_mutex.h"
```

### Data Structures

- class [OsclComponentRegistry](#)
- class [OsclComponentRegistryData](#)
- class [OsclComponentRegistryElement](#)

#### 7.72.1 Detailed Description

Server-side implementation of OsclRegistry interfaces.

## 7.73 oscl\_registry\_serv\_impl\_global.h File Reference

```
#include "osclconfig_proc.h"
#include "oscl_base.h"
```

## 7.74 oscl\_registry\_serv\_impl\_tls.h File Reference

```
#include "osclconfig_proc.h"
#include "oscl_registry_serv_impl.h"
#include "oscl_registry_types.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
```

### Data Structures

- class [OsclRegistryServTlsImpl](#)

## 7.75 oscl\_registry\_types.h File Reference

Common types used in Oscl registry interfaces.

```
#include "oscl_types.h"  
#include "oscl_string_containers.h"
```

### Data Structures

- class [OsclRegistryAccessElement](#)

### TypeDefs

- typedef [OsclAny](#) \* [OsclComponentFactory](#)

#### 7.75.1 Detailed Description

Common types used in Oscl registry interfaces.

## 7.76 oscl\_scheduler.h File Reference

```
#include "oscl_scheduler_types.h"
#include "oscl_scheduler_ao.h"
#include "oscl_scheduler_threadcontext.h"
#include "oscl_defalloc.h"
#include "oscl_mem.h"
```

### Data Structures

- class [OsclExecScheduler](#)
- class [OsclExecSchedulerCommonBase](#)
- class [OsclScheduler](#)
- class [OsclSchedulerObserver](#)
- class [PVSchedulerStopper](#)

### Defines

- #define [PVSCHEDNAMELEN](#) 30

## 7.77 oscl\_scheduler\_ao.h File Reference

Oscl Scheduler user execution object classes.

```
#include "oscl_scheduler_aobase.h"
#include "oscl_mem.h"
#include "oscl_scheduler_types.h"
```

### Data Structures

- class [OsclActiveObject](#)
- class [OsclTimerObject](#)

#### 7.77.1 Detailed Description

Oscl Scheduler user execution object classes.

## 7.78 oscl\_scheduler\_aobase.h File Reference

Oscl Scheduler internal active object classes.

```
#include "oscl_namestring.h"
#include "oscl_scheduler_threadcontext.h"
#include "oscl_scheduler_readyq.h"
#include "oscl_string_containers.h"
#include "oscl_scheduler_types.h"
```

### Data Structures

- class [PVActiveBase](#)
- class [PVActiveStats](#)

### Defines

- #define [OSCL\\_ZEROIZE](#)(ptr, size) oscl\_memset(ptr, 0, size)
- #define [PVEEXECNAMELEN](#) 30

#### 7.78.1 Detailed Description

Oscl Scheduler internal active object classes.

## 7.79 oscl\_scheduler\_readyq.h File Reference

ready q types for oscl scheduler

```
#include "oscl_scheduler_tuneables.h"
#include "oscl_priqueue.h"
#include "oscl_base_alloc.h"
#include "oscl_semaphore.h"
#include "oscl_mem.h"
#include "oscl_string_containers.h"
#include "oscl_scheduler_types.h"
#include "oscl_mutex.h"
```

### Data Structures

- class [OsclReadyAlloc](#)
- class [OsclReadyCompare](#)
- class [OsclReadyQ](#)
- class [OsclTimerCompare](#)
- class [OsclTimerQ](#)
- class [TReadyQueLink](#)

### Typedefs

- typedef [PVActiveBase](#) \* TOsclReady

#### 7.79.1 Detailed Description

ready q types for oscl scheduler

## 7.80 oscl\_scheduler\_threadcontext.h File Reference

Thread context functions needed by oscl scheduler.

```
#include "oscl_double_list.h"
#include "oscl_mutex.h"
#include "oscl_aostatus.h"
```

### Data Structures

- class [PVThreadContext](#)

### Enumerations

- enum [TPVThreadContext](#) { [EPVThreadContext\\_InThread](#), [EPVThreadContext\\_OsclThread](#), [EPVThreadContext\\_NonOsclThread](#), [EPVThreadContext\\_Undetermined](#) }

### 7.80.1 Detailed Description

Thread context functions needed by oscl scheduler.

## 7.81 oscl\_scheduler\_tuneables.h File Reference

Tuneable settings for Oscl Scheduler.

```
#include "osclconfig_proc.h"
```

### Defines

- #define PV\_SCHED\_ENABLE\_AO\_STATS 1
- #define PV\_SCHED\_ENABLE\_LOOP\_STATS 0
- #define PV\_SCHED\_ENABLE\_PERF\_LOGGING 1
- #define PV\_SCHED\_ENABLE\_THREAD\_CONTEXT\_CHECKS 1
- #define PV\_SCHED\_LOG\_Q 0
- #define PV\_SCHED\_CHECK\_Q 0
- #define PV\_SCHED\_FAIR\_SCHEDULING 1
- #define OSCL\_PERF\_SUMMARY\_LOGGING 0

### 7.81.1 Detailed Description

Tuneable settings for Oscl Scheduler.

## 7.82 oscl\_scheduler\_types.h File Reference

Scheduler common types include file.

```
#include "osclconfig_proc.h"  
#include "oscl_aostatus.h"  
#include "oscl_heapbase.h"
```

### Data Structures

- class [OsclExecSchedulerBase](#)

#### 7.82.1 Detailed Description

Scheduler common types include file.

## 7.83 oscl\_semaphore.h File Reference

This file provides implementation of mutex.

```
#include "osclconfig_proc.h"  
#include "oscl_thread.h"
```

### Data Structures

- class [OsclSemaphore](#)

#### 7.83.1 Detailed Description

This file provides implementation of mutex.

## 7.84 oscl\_shared\_ptr.h File Reference

This file defines a template class [OsclSharedPtr](#) which is a "smart pointer" to the parameterized type.

```
#include "oscl_base.h"
#include "oscl_refcounter.h"
#include "osclconfig_compiler_warnings.h"
```

### Data Structures

- class [OsclSharedPtr](#)  
*A parameterized smart pointer class.*

### Defines

- #define [OSCL\\_DISABLE\\_WARNING\\_RETURN\\_TYPE\\_NOT\\_UDT](#)

#### 7.84.1 Detailed Description

This file defines a template class [OsclSharedPtr](#) which is a "smart pointer" to the parameterized type.

## 7.85 oscl\_singleton.h File Reference

This file defines the [OsclSingleton](#) class. This class provides a container which used to give access to a set of process-level singleton objects. Each object is indexed by an integer ID, listed below. There can only be one instance of each object per process at a given time.

```
#include "oscl_base.h"
#include "oscl_defalloc.h"
```

### Data Structures

- class [OsclSingleton](#)
- class [OsclSingletonRegistry](#)
- class [SingletonTable](#)

### Variables

- const uint32 [OSCL\\_SINGLETON\\_ID\\_TEST](#) = 0
- const uint32 [OSCL\\_SINGLETON\\_ID\\_OSCLMEM](#) = 1
- const uint32 [OSCL\\_SINGLETON\\_ID\\_PVLOGGER](#) = 2
- const uint32 [OSCL\\_SINGLETON\\_ID\\_PVSCHEDULER](#) = 3
- const uint32 [OSCL\\_SINGLETON\\_ID\\_PVERRORTRAP](#) = 4
- const uint32 [OSCL\\_SINGLETON\\_ID\\_SDPMEDIAPARSER](#) = 5
- const uint32 [OSCL\\_SINGLETON\\_ID\\_PAYLOADPARSER](#) = 6
- const uint32 [OSCL\\_SINGLETON\\_ID\\_CPM\\_PLUGIN](#) = 7
- const uint32 [OSCL\\_SINGLETON\\_ID\\_PVMFRECOGNIZER](#) = 8
- const uint32 [OSCL\\_SINGLETON\\_ID\\_OSCLREGISTRY](#) = 9
- const uint32 [OSCL\\_SINGLETON\\_ID\\_OMX](#) = 10
- const uint32 [OSCL\\_SINGLETON\\_ID\\_OMXMASTERCORE](#) = 11
- const uint32 [OSCL\\_SINGLETON\\_ID\\_TICKCOUNT](#) = 12
- const uint32 [OSCL\\_SINGLETON\\_ID\\_LAST](#) = 13

### 7.85.1 Detailed Description

This file defines the [OsclSingleton](#) class. This class provides a container which used to give access to a set of process-level singleton objects. Each object is indexed by an integer ID, listed below. There can only be one instance of each object per process at a given time.

[OsclSingleton](#) is initialized in [OsclBase::Init](#).

## 7.85.2 Variable Documentation

- 7.85.2.1 `const uint32 OSCL_SINGLETON_ID_CPM_PLUGIN = 7`
- 7.85.2.2 `const uint32 OSCL_SINGLETON_ID_LAST = 13`
- 7.85.2.3 `const uint32 OSCL_SINGLETON_ID_OMX = 10`
- 7.85.2.4 `const uint32 OSCL_SINGLETON_ID_OMXMASTERCORE = 11`
- 7.85.2.5 `const uint32 OSCL_SINGLETON_ID_OSCLMEM = 1`
- 7.85.2.6 `const uint32 OSCL_SINGLETON_ID_OSCLREGISTRY = 9`
- 7.85.2.7 `const uint32 OSCL_SINGLETON_ID_PAYLOADPARSER = 6`
- 7.85.2.8 `const uint32 OSCL_SINGLETON_ID_PVERRORTRAP = 4`
- 7.85.2.9 `const uint32 OSCL_SINGLETON_ID_PVLOGGER = 2`
- 7.85.2.10 `const uint32 OSCL_SINGLETON_ID_PVMFRECOGNIZER = 8`
- 7.85.2.11 `const uint32 OSCL_SINGLETON_ID_PVSCHEDULER = 3`
- 7.85.2.12 `const uint32 OSCL_SINGLETON_ID_SDPMEDIAPARSER = 5`
- 7.85.2.13 `const uint32 OSCL_SINGLETON_ID_TEST = 0`
- 7.85.2.14 `const uint32 OSCL_SINGLETON_ID_TICKCOUNT = 12`

## 7.86 oscl\_snprintf.h File Reference

Provides a portable implementation of snprintf.

```
#include "oscl_base.h"  
#include "osclconfig_util.h"
```

### Functions

- OSCL\_IMPORT\_REF int32 [oscl\\_snprintf](#) (char \*str, uint32 count, const char \*fmt,...)
- OSCL\_IMPORT\_REF int32 [oscl\\_snprintf](#) ([oscl\\_wchar](#) \*str, uint32 count, const [oscl\\_wchar](#) \*fmt,...)
- OSCL\_IMPORT\_REF int32 [oscl\\_vsnprintf](#) (char \*str, uint32 count, const char \*fmt, va\_list args)
- OSCL\_IMPORT\_REF int32 [oscl\\_vsnprintf](#) ([oscl\\_wchar](#) \*str, uint32 count, const [oscl\\_wchar](#) \*fmt, va\_list args)

### 7.86.1 Detailed Description

Provides a portable implementation of snprintf.

## 7.87 oscl\_socket.h File Reference

The file [oscl\\_socket.h](#) defines the OSCL Socket APIs.

```
#include "osclconfig_io.h"  
#include "oscl_heapbase.h"  
#include "oscl_defalloc.h"  
#include "oscl_vector.h"  
#include "oscl_mem.h"  
#include "oscl_socket_types.h"
```

### Data Structures

- class [OsclSocketServ](#)
- class [OsclTCPSocket](#)
- class [OsclUDPSocket](#)

#### 7.87.1 Detailed Description

The file [oscl\\_socket.h](#) defines the OSCL Socket APIs.

## 7.88 oscl\_socket\_accept.h File Reference

```
#include "oscl_socket_imp.h"
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_method.h"
```

### Data Structures

- class [OsclAcceptMethod](#)
- class [OsclAcceptRequest](#)

## 7.89 oscl\_socket\_bind.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

### Data Structures

- class [OsclBindMethod](#)
- class [OsclBindRequest](#)

## 7.90 oscl\_socket\_connect.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

### Data Structures

- class [OsclConnectMethod](#)
- class [OsclConnectRequest](#)

## **7.91 oscl\_socket\_imp.h File Reference**

```
#include "oscl_socket_tuneables.h"  
#include "oscl_socket_imp_pv.h"
```

## 7.92 oscl\_socket\_imp\_base.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_request.h"
#include "oscl_defalloc.h"
#include "oscl_mutex.h"
#include "oscl_socket_stats.h"
#include "oscl_base.h"
```

### Data Structures

- class [OsclSocketIBase](#)

## 7.93 oscl\_socket\_imp\_pv.h File Reference

```
#include "oscl_socket_imp_base.h"
```

### Data Structures

- class [OsclSocketI](#)

### Defines

- #define [PVSOCK\\_ERR\\_BAD\\_PARAM](#) (-1)
- #define [PVSOCK\\_ERR SOCK\\_NOT\\_OPEN](#) (-2)
- #define [PVSOCK\\_ERR SOCK\\_NO\\_SERV](#) (-3)
- #define [PVSOCK\\_ERR SERV\\_NOT\\_CONNECTED](#) (-4)
- #define [PVSOCK\\_ERR SOCK\\_NOT\\_CONNECTED](#) (-5)
- #define [PVSOCK\\_ERR NOT\\_IMPLEMENTED](#) (-6)

### 7.93.1 Define Documentation

#### 7.93.1.1 #define PVSOCK\_ERR\_BAD\_PARAM (-1)

some error codes for request completion these are negative so they won't conflict with errors from the OS socket layer.

#### 7.93.1.2 #define PVSOCK\_ERR\_NOT\_IMPLEMENTED (-6)

#### 7.93.1.3 #define PVSOCK\_ERR\_SERV\_NOT\_CONNECTED (-4)

#### 7.93.1.4 #define PVSOCK\_ERR SOCK\_NO\_SERV (-3)

#### 7.93.1.5 #define PVSOCK\_ERR SOCK\_NOT\_CONNECTED (-5)

#### 7.93.1.6 #define PVSOCK\_ERR SOCK\_NOT\_OPEN (-2)

## 7.94 oscl\_socket\_listen.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

### Data Structures

- class [OsclListenMethod](#)
- class [OsclListenRequest](#)

### Defines

- #define [OSCL\\_SOCKET\\_LISTEN\\_H\\_INCLUDEDd](#)

#### 7.94.1 Define Documentation

##### 7.94.1.1 #define OSCL\_SOCKET\_LISTEN\_H\_INCLUDEDd

## 7.95 oscl\_socket\_method.h File Reference

```
#include "osclconfig_io.h"
#include "oscl_socket_types.h"
#include "oscl_scheduler_ao.h"
#include "oscl_socket_request.h"
#include "pvlogger.h"
#include "oscl_socket_tuneables.h"
#include "oscl_ip_socket.h"
#include "oscl_socket_imp.h"
```

### Data Structures

- class [OsclSocketMethod](#)
- class [OsclSocketRequestAO](#)

### Defines

- #define [MSEC\\_TO\\_MICROSEC](#) 1000

#### 7.95.1 Define Documentation

##### 7.95.1.1 #define MSEC\_TO\_MICROSEC 1000

## 7.96 oscl\_socket\_recv.h File Reference

```
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

### Data Structures

- class [OsclRecvMethod](#)
- class [OsclRecvRequest](#)

## 7.97 oscl\_socket\_recv\_from.h File Reference

```
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

### Data Structures

- class [OsclRecvFromMethod](#)
- class [OsclRecvFromRequest](#)

## 7.98 oscl\_socket\_request.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_socket_tuneables.h"
```

### Data Structures

- class [AcceptParam](#)
- class [BindParam](#)
- class [ConnectParam](#)
- class [ListenParam](#)
- class [OsclSocketRequest](#)
- class [PVSockBufRecv](#)
- class [PVSockBufSend](#)
- class [RecvFromParam](#)
- class [RecvParam](#)
- class [SendParam](#)
- class [SendToParam](#)
- class [ShutdownParam](#)
- class [SocketRequestParam](#)

## 7.99 oscl\_socket\_send.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_method.h"
```

### Data Structures

- class [OsclSendMethod](#)
- class [OsclSendRequest](#)

## 7.100 oscl\_socket\_send\_to.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

### Data Structures

- class [OsclSendToMethod](#)
- class [OsclSendToRequest](#)

## 7.101 oscl\_socket\_serv\_imp.h File Reference

```
#include "osclconfig_io.h"
#include "oscl_socket_tuneables.h"
#include "oscl_socket_serv_imp_pv.h"
```

## 7.102 oscl\_socket\_serv\_imp\_base.h File Reference

```
#include "oscl_base.h"
#include "oscl_socket_stats.h"
```

### Data Structures

- class [OsclSocketServIBase](#)

## 7.103 oscl\_socket\_serv\_imp\_pv.h File Reference

```
#include "oscl_socket_serv_imp_base.h"
#include "oscl_socket_serv_imp_reqlist.h"
#include "oscl_socket_tuneables.h"
#include "oscl_scheduler_ao.h"
```

### Data Structures

- class [OsclSocketServI](#)

### Defines

- #define [OSCL\\_READSET\\_FLAG](#) 0x04
- #define [OSCL\\_WRITESET\\_FLAG](#) 0x02
- #define [OSCL\\_EXCEPTSET\\_FLAG](#) 0x01

#### 7.103.1 Define Documentation

##### 7.103.1.1 #define OSCL\_EXCEPTSET\_FLAG 0x01

##### 7.103.1.2 #define OSCL\_READSET\_FLAG 0x04

A bitmask for socket select operations

##### 7.103.1.3 #define OSCL\_WRITESET\_FLAG 0x02

## 7.104 oscl\_socket\_serv\_imp\_reqlist.h File Reference

```
#include "oscl_socket_tuneables.h"
#include "oscl_defalloc.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
```

### Data Structures

- class [OsclSocketServRequestList](#)
- class [OsclSocketServRequestQElem](#)

## 7.105 oscl\_socket\_shutdown.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_method.h"
```

### Data Structures

- class [OsclShutdownMethod](#)
- class [OsclShutdownRequest](#)

## 7.106 oscl\_socket\_stats.h File Reference

```
#include "oscl_base.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_mutex.h"
#include "oscl_socket_tuneables.h"
```

### Enumerations

- enum TOsclSocketStatEvent { EOscSocket\_RequestAO\_Success, EOscSocket\_RequestAO\_Canceled, EOscSocket\_RequestAO\_Error, EOscSocket\_RequestAO\_Timeout, EOscSocket\_ServRequestIssued, EOscSocket\_ServPoll, EOscSocket\_OS, EOscSocket\_Readable, EOscSocket\_Writable, EOscSocket\_Except, EOscSocket\_DataRecv, EOscSocket\_DataSent, EOscSocket\_ServRequestComplete, EOscSocket\_ServRequestCancelIssued, EOscSocketServ\_LoopsockOk, EOscSocketServ\_LoopsockError }
- enum TOsclSocketServStatEvent { EOscSocketServ\_SelectNoActivity = 0, EOscSocketServ\_SelectActivity, EOscSocketServ\_SelectRescheduleAsap, EOscSocketServ\_SelectReschedulePoll, EOscSocketServ\_LastEvent }

#### 7.106.1 Enumeration Type Documentation

##### 7.106.1.1 enum TOsclSocketServStatEvent

###### Enumeration values:

**EOscSocketServ\_SelectNoActivity**  
**EOscSocketServ\_SelectActivity**  
**EOscSocketServ\_SelectRescheduleAsap**  
**EOscSocketServ\_SelectReschedulePoll**  
**EOscSocketServ\_LastEvent**

##### 7.106.1.2 enum TOsclSocketStatEvent

Socket diagnostics.

###### Enumeration values:

**EOscSocket\_RequestAO\_Success**  
**EOscSocket\_RequestAO\_Canceled**  
**EOscSocket\_RequestAO\_Error**  
**EOscSocket\_RequestAO\_Timeout**  
**EOscSocket\_ServRequestIssued**  
**EOscSocket\_ServPoll**  
**EOscSocket\_OS**  
**EOscSocket\_Readable**  
**EOscSocket\_Writable**

---

**EOselSocket\_Except**  
**EOselSocket\_DataRecv**  
**EOselSocket\_DataSent**  
**EOselSocket\_ServRequestComplete**  
**EOselSocket\_ServRequestCancelIssued**  
**EOselSocketServ\_LoopsockOk**  
**EOselSocketServ\_LoopsockError**

## 7.107 oscl\_socket\_tuneables.h File Reference

```
#include "osclconfig_io.h"
#include "osclconfig_proc.h"
```

### Defines

- #define PV\_SOCKET\_REQUEST\_AO\_PRIORITY OsclActiveObject::EPriorityNominal
- #define PV\_OSCL\_SOCKET\_STATS\_LOGGING 0
- #define PV\_SOCKET\_SERVER 1
- #define PV\_SOCKET\_SERVER\_IS\_THREAD OSCL\_HAS\_THREAD\_SUPPORT
- #define PV\_SOCKET\_SERVER\_SELECT 0
- #define PV\_SOCKET\_SERVER\_THREAD\_PRIORITY ThreadPriorityAboveNormal
- #define PV\_SOCKET\_SERVER\_SELECT\_TIMEOUT\_MSEC (-1)
- #define PV\_SOCKET\_SERVER\_SELECT\_LOOPBACK\_SOCKET 0
- #define PV\_SOCKET\_SERVER\_AO\_PRIORITY (OsclActiveObject::EPriorityNominal)
- #define PV\_SOCKET\_SERVER\_AO\_INTERVAL\_MSEC 5
- #define PV\_OSCL\_SOCKET\_SERVER\_LOGGER\_OUTPUT 0
- #define PV\_OSCL\_SOCKET\_1MB\_RECV\_BUF 0
- #define PV\_SOCKET\_SERVI\_STATS 0

### 7.107.1 Define Documentation

#### 7.107.1.1 #define PV\_OSCL\_SOCKET\_1MB\_RECV\_BUF 0

Set this to 0 or 1 to enable/disable setting the socket receive buffer size to 1 MB in the Bind call. This setting only affects PV socket server implementations.

When set to 1, the code will use the OsclSetRecvBufferSize macro to set the buffer size in the Bind call.

This setting was found to improve streaming performance on WinMobile devices, but should not generally be used.

#### 7.107.1.2 #define PV\_OSCL\_SOCKET\_SERVER\_LOGGER\_OUTPUT 0

Set this to 0 or 1 to enable/disable [PVLogger](#) output from PV socket server. Note that socket server logging will appear in a different file when running threaded mode of socket server. This is quite a bit of logging, so it should generally be disabled.

#### 7.107.1.3 #define PV\_OSCL\_SOCKET\_STATS\_LOGGING 0

Set this to 0 or 1 to enable/disable socket stats logging with "OsclSocketStats" node. This feature is fairly costly so should be off in production code.

#### 7.107.1.4 #define PV\_SOCKET\_REQUEST\_AO\_PRIORITY OsclActiveObject::EPriority-Nominal

PV\_SOCKET\_REQUEST\_AO\_PRIORITY sets the priority of the socket request completion AOs.

**7.107.1.5 #define PV\_SOCKET\_SERVER 1**

Enable/disable the PV socket server here.

**7.107.1.6 #define PV\_SOCKET\_SERVER\_AO\_INTERVAL\_MSEC 5**

PV\_SOCKET\_SERVER\_AO\_INTERVAL\_MSEC sets the AO scheduling interval of the PV socket server AO for non-threaded implementations.

**7.107.1.7 #define PV\_SOCKET\_SERVER\_AO\_PRIORITY (OsclActiveObject::EPriority-Nominal)**

PV\_SOCKET\_SERVER\_AO\_PRIORITY sets priority of the PV socket server AO for non-threaded implementations.

**7.107.1.8 #define PV\_SOCKET\_SERVER\_IS\_THREAD OSCL\_HAS\_THREAD\_SUPPORT**

PV\_SOCKET\_SERVER\_IS\_THREAD chooses either the threaded or AO-based implementation of the PV socket server

**7.107.1.9 #define PV\_SOCKET\_SERVER\_SELECT 0**

PV\_SOCKET\_SERVER\_SELECT chooses whether to use "select" call or not. In threaded mode, select call is required and is forced to "1". In AO mode, "select" call is an option that defaults to "0". Avoiding any "select" call was found to greatly reduce CPU usage on WinMobile devices.

**7.107.1.10 #define PV\_SOCKET\_SERVER\_SELECT\_LOOPBACK\_SOCKET 0**

PV\_SOCKET\_SERVER\_SELECT\_LOOPBACK\_SOCKET enables the feature to wakeup the select call by writing to a loopback socket each time a new request comes in. This option is required to support the blocking select option of threaded server mode. This option is forced to "0" in AO mode.

**7.107.1.11 #define PV\_SOCKET\_SERVER\_SELECT\_TIMEOUT\_MSEC (-1)**

PV\_SOCKET\_SERVER\_SELECT\_TIMEOUT\_MSEC sets duration of the select call in the PV socket server thread for the polling select loop implementation. When the timeout is -1, the select call will block forever waiting on a new request and will use a loopback socket to signal a new request. Note: if infinite wait is selected, but loopback socket is not available, the implementation will poll at 10 msec intervals.

**7.107.1.12 #define PV\_SOCKET\_SERVER\_THREAD\_PRIORITY ThreadPriorityAboveNormal**

PV\_SOCKET\_SERVER\_THREAD\_PRIORITY sets the priority of the PV socket server thread.

**7.107.1.13 #define PV\_SOCKET\_SERVI\_STATS 0**

For detailed performance breakdown of time spend in [OsclSocketServI](#) AO. Output is logged under "OsclSchedulerPerfStats" node. Should be off in production code. This option is forced to "0" in threaded mode.

## 7.108 oscl\_socket\_types.h File Reference

```
#include "osclconfig_io.h"
#include "oscl_types.h"
#include "oscl_scheduler_types.h"
#include "oscl_namestring.h"
#include "oscl_stdstring.h"
```

### Data Structures

- class [OsclNetworkAddress](#)
- class [OsclSocketObserver](#)

### Defines

- #define [PVNETWORKADDRESS\\_LEN](#) 50

### Enumerations

- enum [TPVSocketFxn](#) { [EPVSocketSend](#) = 0, [EPVSocketSendTo](#), [EPVSocketRecv](#), [EPVSocketRecvFrom](#), [EPVSocketConnect](#), [EPVSocketAccept](#), [EPVSocketShutdown](#), [EPVSocketBind](#), [EPVSocketListen](#), [EPVSocket\\_Last](#) }
- enum [TPVSocketEvent](#) { [EPVSocketSuccess](#), [EPVSocketPending](#), [EPVSocketTimeout](#), [EPVSocketFailure](#), [EPVSocketCancel](#) }
- enum [TPVSocketShutdown](#) { [EPVSocketSendShutdown](#), [EPVSocketRecvShutdown](#), [EPVSocketBothShutdown](#) }

#### 7.108.1 Define Documentation

##### 7.108.1.1 #define PVNETWORKADDRESS\_LEN 50

#### 7.108.2 Enumeration Type Documentation

##### 7.108.2.1 enum TPVSocketEvent

Return codes for asynchronous APIs

Enumeration values:

**EPVSocketSuccess**  
**EPVSocketPending**  
**EPVSocketTimeout**  
**EPVSocketFailure**  
**EPVSocketCancel**

**7.108.2.2 enum TPVSocketFxn**

Enumeration values:

- EPVSocketSend**
- EPVSocketSendTo**
- EPVSocketRecv**
- EPVSocketRecvFrom**
- EPVSocketConnect**
- EPVSocketAccept**
- EPVSocketShutdown**
- EPVSocketBind**
- EPVSocketListen**
- EPVSocket\_Last**

**7.108.2.3 enum TPVSocketShutdown**

Enumeration values:

- EPVSocketSendShutdown**
- EPVSocketRecvShutdown**
- EPVSocketBothShutdown**

## 7.109 oscl\_stdstring.h File Reference

This file provides standard string operations such as strlen, strncpy, etc. ANSI defines undefined behavior when the destination pointer is null for operations such as strncpy, strncat, etc. But, we chose to define one. In such cases, we return the destination as null.

```
#include "oscl_base.h"
```

### Functions

- OSCL\_IMPORT\_REF uint32 `oscl_strlen` (const char \*str)
- OSCL\_IMPORT\_REF uint32 `oscl_strlen` (const `oscl_wchar` \*str)
- OSCL\_IMPORT\_REF char \* `oscl_strncpy` (char \*dest, const char \*src, uint32 count)
- OSCL\_IMPORT\_REF `oscl_wchar` \* `oscl_strncpy` (`oscl_wchar` \*dest, const `oscl_wchar` \*src, uint32 count)
- OSCL\_IMPORT\_REF int32 `oscl_strcmp` (const char \*str1, const char \*str2)
- OSCL\_IMPORT\_REF int32 `oscl_stremp` (const `oscl_wchar` \*str1, const `oscl_wchar` \*str2)
- OSCL\_IMPORT\_REF int32 `oscl_strncmp` (const char \*str1, const char \*str2, uint32 count)
- OSCL\_IMPORT\_REF int32 `oscl_strncmp` (const `oscl_wchar` \*str1, const `oscl_wchar` \*str2, uint32 count)
- OSCL\_IMPORT\_REF char \* `oscl_strncat` (char \*dest, const char \*src, uint32 count)
- OSCL\_IMPORT\_REF `oscl_wchar` \* `oscl_strncat` (`oscl_wchar` \*dest, const `oscl_wchar` \*src, uint32 count)
- OSCL\_IMPORT\_REF const char \* `oscl_strchr` (const char \*str, int32 c)
- OSCL\_IMPORT\_REF char \* `oscl_strchr` (char \*str, int32 c)
- OSCL\_IMPORT\_REF const `oscl_wchar` \* `oscl_strchr` (const `oscl_wchar` \*str, int32 c)
- OSCL\_IMPORT\_REF `oscl_wchar` \* `oscl_strchr` (`oscl_wchar` \*str, int32 c)
- OSCL\_IMPORT\_REF const char \* `oscl strrchr` (const char \*str, int32 c)
- OSCL\_IMPORT\_REF char \* `oscl strrchr` (char \*str, int32 c)
- OSCL\_IMPORT\_REF const `oscl_wchar` \* `oscl strrchr` (const `oscl_wchar` \*str, int32 c)
- OSCL\_IMPORT\_REF `oscl_wchar` \* `oscl strrchr` (`oscl_wchar` \*str, int32 c)
- OSCL\_IMPORT\_REF char \* `oscl_strset` (char \*dest, char val, uint32 count)
- OSCL\_IMPORT\_REF `oscl_wchar` \* `oscl_strset` (`oscl_wchar` \*dest, `oscl_wchar` val, uint32 count)
- OSCL\_IMPORT\_REF int32 `oscl_CIstrcmp` (const char \*str1, const char \*str2)
- OSCL\_IMPORT\_REF int32 `oscl_CIstrcmp` (const `oscl_wchar` \*str1, const `oscl_wchar` \*str2)
- OSCL\_IMPORT\_REF int32 `oscl_CIstrncmp` (const char \*str1, const char \*str2, uint32 count)
- OSCL\_IMPORT\_REF int32 `oscl_CIstrncmp` (const `oscl_wchar` \*str1, const `oscl_wchar` \*str2, uint32 count)
- OSCL\_IMPORT\_REF char `oscl_tolower` (const char car)
- OSCL\_IMPORT\_REF `oscl_wchar` `oscl_tolower` (const `oscl_wchar` car)
- OSCL\_IMPORT\_REF bool `oscl_isLetter` (const char car)
- OSCL\_IMPORT\_REF const char \* `oscl_strstr` (const char \*str1, const char \*str2)
- OSCL\_IMPORT\_REF char \* `oscl_strstr` (char \*str1, const char \*str2)
- OSCL\_IMPORT\_REF const `oscl_wchar` \* `oscl_strstr` (const `oscl_wchar` \*str1, const `oscl_wchar` \*str2)
- OSCL\_IMPORT\_REF `oscl_wchar` \* `oscl_strstr` (`oscl_wchar` \*str1, const `oscl_wchar` \*str2)
- OSCL\_IMPORT\_REF char \* `oscl_streat` (char \*dest, const char \*src)
- OSCL\_IMPORT\_REF `oscl_wchar` \* `oscl_streat` (`oscl_wchar` \*dest, const `oscl_wchar` \*src)

### 7.109.1 Detailed Description

This file provides standard string operations such as `strlen`, `strncpy`, etc. ANSI defines undefined behavior when the destination pointer is null for operations such as `strncpy`, `strncat`, etc. But, we chose to define one. In such cases, we return the destination as null.

## 7.110 oscl\_str\_ptr\_len.h File Reference

Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.

```
#include "oscl_base.h"
#include "oscl_stdstring.h"
```

### Data Structures

- struct [StrCSumPtrLen](#)  
*same as [StrPtrLen](#), but includes checksum field and method to speed up querying*
- struct [StrPtrLen](#)  
*This data structure encapsulates a set of functions used to perform.*
- struct [WStrPtrLen](#)  
*This data structure encapsulates a set of functions used to perform.*

### Typedefs

- typedef StrPtrLen [StrPtrLen](#)  
*This data structure encapsulates a set of functions used to perform.*
- typedef WStrPtrLen [WStrPtrLen](#)  
*This data structure encapsulates a set of functions used to perform.*
- typedef StrCSumPtrLen [StrCSumPtrLen](#)  
*same as [StrPtrLen](#), but includes checksum field and method to speed up querying*
- typedef [WStrPtrLen](#) [OSCL\\_TStrPtrLen](#)

### Variables

- const uint8 [OSCL\\_ASCII\\_CASE\\_MAGIC\\_BIT](#) = 0x20

#### 7.110.1 Detailed Description

Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.

## 7.111 oscl\_string.h File Reference

Provides a standardized set of string containers that can be used in place of character arrays.

```
#include "oscl_base.h"  
#include "oscl_mem.h"
```

### Data Structures

- class [OSCL\\_String](#)
- class [OSCL\\_wString](#)

#### 7.111.1 Detailed Description

Provides a standardized set of string containers that can be used in place of character arrays.

## 7.112 oscl\_string\_containers.h File Reference

Provides a standardized set of string containers that can be used in place of character arrays.

```
#include "oscl_string.h"
#include "oscl_defalloc.h"
#include "oscl_refcounter.h"
#include "oscl_error.h"
#include "oscl_string_rep.h"
#include "oscl_stdstring.h"
#include "oscl_mem.h"
```

### Data Structures

- class [OSCL\\_FastString](#)
- class [OSCL\\_HeapString](#)
- class [OSCL\\_HeapStringA](#)
- class [OSCL\\_StackString](#)
- class [OSCL\\_wFastString](#)
- class [OSCL\\_wHeapString](#)
- class [OSCL\\_wHeapStringA](#)
- class [OSCL\\_wStackString](#)

#### 7.112.1 Detailed Description

Provides a standardized set of string containers that can be used in place of character arrays.

## 7.113 oscl\_string\_rep.h File Reference

Contains some internal implementation for string containers.

```
#include "oscl_defalloc.h"
```

### Data Structures

- class [CFastRep](#)
- class [CHheapRep](#)
- class [CStackRep](#)

#### 7.113.1 Detailed Description

Contains some internal implementation for string containers.

## 7.114 oscl\_string\_uri.h File Reference

Utilities to unescape URIs.

```
#include "oscl_base.h"  
#include "oscl_string.h"
```

### Functions

- OSCL\_IMPORT\_REF bool [oscl\\_str\\_unescape\\_uri](#) (const char \*str\_buf\_in, char \*str\_buf\_out, uint32 max\_out\_buf\_bytes, uint32 max\_bytes, uint32 &out\_buf\_len)  
*unescape any of the special escape sequence in the uri string*
- OSCL\_IMPORT\_REF bool [oscl\\_str\\_unescape\\_uri](#) (const [OSCL\\_String](#) &oscl\_str\_in, [OSCL\\_String](#) &oscl\_str\_out, uint32 &out\_buf\_len)  
*unescape any of the special escape sequence in the uri string*

### 7.114.1 Detailed Description

Utilities to unescape URIs.

## 7.115 oscl\_string\_utf8.h File Reference

Utilities to validate and truncate UTF-8 encoded strings.

```
#include "oscl_base.h"
```

### Functions

- OSCL\_IMPORT\_REF bool `oscl_str_is_valid_utf8` (const uint8 \*str\_buf, uint32 &num\_valid\_characters, uint32 max\_bytes=0, uint32 max\_char\_2\_valid=0, uint32 \*num\_byte\_4\_char=NULL)

*Check if the input string contains any illegal UTF-8 character. The function scans the string and validate that each character is a valid utf-8. It stops at the first NULL character, invalid character or the max\_byte value. The string is valid if and only if every character is a valid utf-8 character and the scanning stopped on a character boundary.*

- OSCL\_IMPORT\_REF int32 `oscl_str_truncate_utf8` (uint8 \*str\_buf, uint32 max\_char, uint32 max\_bytes=0)

*Truncates the UTF-8 string upto the required size.*

### 7.115.1 Detailed Description

Utilities to validate and truncate UTF-8 encoded strings.

## 7.116 oscl\_string\_utils.h File Reference

Utilities to parse and convert strings.

```
#include "oscl_base.h"
```

### Defines

- #define `oscl_isdigit(c)` ((c) >= '0' && (c) <= '9')

### Functions

- OSCL\_IMPORT\_REF const char \* `skip_whitespace` (const char \*ptr)
- OSCL\_IMPORT\_REF char \* `skip_whitespace` (char \*ptr)
- OSCL\_IMPORT\_REF const char \* `skip_whitespace` (const char \*start, const char \*end)
- OSCL\_IMPORT\_REF const char \* `skip_to_whitespace` (const char \*start, const char \*end)
- OSCL\_IMPORT\_REF const char \* `skip_to_line_term` (const char \*start\_ptr, const char \*end\_ptr)
- OSCL\_IMPORT\_REF const char \* `skip_whitespace_and_line_term` (const char \*start, const char \*end)
- OSCL\_IMPORT\_REF int `extract_string` (const char \*in\_ptr, char \*outstring, int maxsize)
- OSCL\_IMPORT\_REF int `extract_string` (const char \*start, const char \*end, char \*outstring, int maxsize)
- OSCL\_IMPORT\_REF bool `PV_atoi` (const char \*buf, const char new\_format, uint32 &value)
- OSCL\_IMPORT\_REF bool `PV_atoi` (const char \*buf, const char new\_format, int length, uint32 &value)
- OSCL\_IMPORT\_REF bool `PV_atoi` (const char \*buf, const char new\_format, int length, `uint64` &value)
- OSCL\_IMPORT\_REF bool `PV_atof` (const char \*buf, `OsclFloat` &value)
- OSCL\_IMPORT\_REF bool `PV_atof` (const char \*buf, int length, `OsclFloat` &value)
- OSCL\_IMPORT\_REF int `oscl_abs` (int aVal)

### 7.116.1 Detailed Description

Utilities to parse and convert strings.

## 7.117 oscl\_string\_xml.h File Reference

Utilities to escape special characters in XML strings.

```
#include "oscl_base.h"
```

### Functions

- OSCL\_IMPORT\_REF bool [oscl\\_str\\_need\\_escape\\_xml](#) (const char \*str\_buf, uint32 &num\_escape\_bytes, uint32 max\_bytes=0)

*Check if the input string contains any special ASCII character like &, <, >, ', ". The function scans the string and check if each character is a special character. It stops at the first NULL character (if max\_bytes = 0), or the max\_byte value.*

- OSCL\_IMPORT\_REF int32 [oscl\\_str\\_escape\\_xml](#) (const char \*str\_buf\_in, char \*str\_buf\_out, uint32 max\_out\_buf\_bytes, uint32 max\_bytes=0, uint32 \*num\_bytes\_written=NULL)

*Escape any of the following special characters in the string Special ASCII characters: &, <, >, ', ".*

### 7.117.1 Detailed Description

Utilities to escape special characters in XML strings.

## 7.118 oscl\_tagtree.h File Reference

The file [oscl\\_tagtree.h](#) ...

```
#include "oscl_base.h"
#include "oscl_map.h"
#include "oscl_vector.h"
#include "oscl_stdstring.h"
#include "osclconfig_compiler_warnings.h"
```

### Data Structures

- struct [const\\_iterator](#)
- struct [iterator](#)
- struct [Node](#)
- struct [Oscl\\_Tag](#)
- struct [Oscl\\_Tag\\_Base](#)
- class [Oscl\\_TagTree](#)

### Defines

- #define [OSCL\\_DISABLE\\_WARNING\\_TRUNCATE\\_DEBUG\\_MESSAGE](#)

#### 7.118.1 Detailed Description

The file [oscl\\_tagtree.h](#) ...

#### 7.118.2 Define Documentation

##### 7.118.2.1 #define OSCL\_DISABLE\_WARNING\_TRUNCATE\_DEBUG\_MESSAGE

## 7.119 oscl\_tcp\_socket.h File Reference

```
#include "oscl_ip_socket.h"
#include "oscl_defalloc.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_socket_listen.h"
#include "oscl_socket_recv.h"
#include "oscl_socket_send.h"
#include "oscl_socket_accept.h"
#include "oscl_socket_shutdown.h"
#include "oscl_socket_connect.h"
#include "oscl_socket_bind.h"
```

### Data Structures

- class [OsclTCPSocketI](#)

## 7.120 oscl\_thread.h File Reference

```
#include "osclconfig_proc.h"
#include "oscl_procstatus.h"
#include "oscl_base.h"
```

### Data Structures

- class [OsclThread](#)

### TypeDefs

- typedef [TOsclThreadFuncRet\(OSCL\\_THREAD\\_DECL \\* TOsclThreadFuncPtr \)\(TOsclThreadFuncArg\)](#)

### Enumerations

- enum [OsclThread\\_State](#) { [Start\\_on\\_creation](#), [Suspend\\_on\\_creation](#) }
- enum [OsclThreadPriority](#) { [ThreadPriorityLowest](#), [ThreadPriorityLow](#), [ThreadPriorityBelowNormal](#), [ThreadPriorityNormal](#), [ThreadPriorityAboveNormal](#), [ThreadPriorityHighest](#), [ThreadPriorityTimeCritical](#) }

### 7.120.1 Detailed Description

. This file provides THREAD implementation that can be ported  
to three OS LINUX, SYMBIAN, WIN32

### 7.120.2 TypeDef Documentation

#### 7.120.2.1 [typedef TOsclThreadFuncRet\(OSCL\\_THREAD\\_DECL \\* TOsclThreadFuncPtr\)\(TOsclThreadFuncArg\)](#)

### 7.120.3 Enumeration Type Documentation

#### 7.120.3.1 enum [OsclThread\\_State](#)

Enumeration values:

[Start\\_on\\_creation](#)

[Suspend\\_on\\_creation](#)

#### 7.120.3.2 enum [OsclThreadPriority](#)

Enumeration values:

[ThreadPriorityLowest](#)

[ThreadPriorityLow](#)

[ThreadPriorityBelowNormal](#)

**ThreadPriorityNormal**

**ThreadPriorityAboveNormal**

**ThreadPriorityHighest**

**ThreadPriorityTimeCritical**

## 7.121 oscl\_tickcount.h File Reference

Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.

```
#include "oscl_base.h"
#include "oscl_tickcount.inl"
```

### Data Structures

- class [OsclTickCount](#)

### Defines

- #define [OSCLTICKCOUNT\\_MAX\\_TICKS](#) 0xffffffff

### 7.121.1 Detailed Description

Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.

## 7.122 oscl\_time.h File Reference

The file `oscl_time.h` defines two classes `NTPTime` and `TimeValue` for getting, manipulating, and formatting time values. The `TimeValue` class is based on the native system time format while `NTPTime` is used for the standard Network Time Protocol format.

```
#include "oscl_base.h"
#include "osclconfig_time.h"
#include "oscl_int64_utils.h"
#include "oscl_time.inl"
```

### Data Structures

- class `NTPTime`

*The NTPTime class represents a time value as the number of seconds since 0h (UTC) Jan. 1, 1900.*

- class `TimeValue`

*The TimeValue class represents a time value in a format native to the system.*

### Typedefs

- typedef char `CtimeStrBuf [CTIME_BUFFER_SIZE]`
- typedef char `PV8601timeStrBuf [PV8601TIME_BUFFER_SIZE]`

### Enumerations

- enum `TimeUnits` { `SECONDS` = 0, `MILLISECONDS` = 1, `MICROSECONDS` = 2 }

*The TimeUnits enum can be used when constructing a `TimeValue` class.*

### Functions

- OSCL\_IMPORT\_REF void `PV8601ToRFC822` (`PV8601timeStrBuf` `pv8601_buffer`, `CtimeStrBuf` `ctime_buffer`)
- OSCL\_IMPORT\_REF void `RFC822ToPV8601` (`CtimeStrBuf` `ctime_buffer`, `PV8601timeStrBuf`)
- OSCL\_COND\_IMPORT\_REF `TimeValue operator-` (`const TimeValue &a`, `const TimeValue &b`)

### Variables

- const int `CTIME_BUFFER_SIZE` = 26
- const int `PV8601TIME_BUFFER_SIZE` = 21
- const long `USEC_PER_SEC` = 1000000
- const long `MSEC_PER_SEC` = 1000
- const uint32 `unix_ntp_offset` = 2208988800U

### 7.122.1 Detailed Description

The file `oscl_time.h` defines two classes `NTPTime` and `TimeValue` for getting, manipulating, and formatting time values. The `TimeValue` class is based on the native system time format while `NTPTime` is used for the standard Network Time Protocol format.

## 7.123 oscl\_timer.h File Reference

```
#include "oscl_base.h"
#include "osclconfig_util.h"
#include "oscl_vector.h"
#include "oscl_tickcount.h"
#include "oscl_rand.h"
#include "oscl_scheduler_ao.h"
```

### Data Structures

- struct [\\_TimerEntry](#)
- class [CallbackTimer](#)
- class [CallbackTimerObserver](#)
- class [OsclTimer](#)
- class [OsclTimerObserver](#)

## 7.124 oscl\_tls.h File Reference

```
#include "oscl_base.h"
#include "oscl_defalloc.h"
```

### Data Structures

- class [OsclTLS](#)
- class [OsclTLSRegistry](#)
- class [TLSStorageOps](#)

### Defines

- #define [OSCL\\_TLS\\_BASE\\_SLOTS](#) OSCL\_TLS\_ID\_BASE\_LAST +1
- #define [OSCL\\_TLS\\_EXTERNAL\\_SLOTS](#) 0
- #define [OSCL\\_TLS\\_MAX\\_SLOTS](#) ( OSCL\_TLS\_BASE\_SLOTS + OSCL\_TLS\_EXTERNAL\_SLOTS)

### Typedefs

- typedef [OsclAny](#) TOsclTlsKey

### Variables

- const uint32 [OSCL\\_TLS\\_ID\\_MAGICNUM](#) = 0
- const uint32 [OSCL\\_TLS\\_ID\\_ERRORHOOK](#) = 1
- const uint32 [OSCL\\_TLS\\_ID\\_PVLOGGER](#) = 2
- const uint32 [OSCL\\_TLS\\_ID\\_TEST](#) = 3
- const uint32 [OSCL\\_TLS\\_ID\\_PVSCHEDULER](#) = 4
- const uint32 [OSCL\\_TLS\\_ID\\_PVERRORTRAP](#) = 5
- const uint32 [OSCL\\_TLS\\_ID\\_SDPMEDIAPARSER](#) = 6
- const uint32 [OSCL\\_TLS\\_ID\\_PAYLOADPARSER](#) = 7
- const uint32 [OSCL\\_TLS\\_ID\\_PVMFRECOGNIZER](#) = 8
- const uint32 [OSCL\\_TLS\\_ID\\_WMDRM](#) = 9
- const uint32 [OSCL\\_TLS\\_ID\\_OSCLREGISTRY](#) = 10
- const uint32 [OSCL\\_TLS\\_ID\\_SQLITE3](#) = 11
- const uint32 [OSCL\\_TLS\\_ID\\_BASE\\_LAST](#) = 11

## 7.125 oscl\_tree.h File Reference

The file [oscl\\_tree.h](#) defines the template class [Oscl\\_Rb\\_Tree](#) which has a very similar API as the STL Tree class. It is an implementation of a Red-Black Tree for use by the [Oscl\\_Map](#) class. Memory allocation is abstracted through the use of an allocator template parameter.

```
#include "oscl_defalloc.h"  
#include "osclconfig_compiler_warnings.h"
```

### Data Structures

- struct [Oscl\\_Pair](#)
- class [Oscl\\_Rb\\_Tree](#)
- class [Oscl\\_Rb\\_Tree\\_Base](#)
- struct [Oscl\\_Rb\\_Tree\\_Const\\_Iterator](#)
- struct [Oscl\\_Rb\\_Tree\\_Iterator](#)
- struct [Oscl\\_Rb\\_Tree\\_Node](#)
- struct [Oscl\\_Rb\\_Tree\\_Node\\_Base](#)

### Defines

- #define [OSCL\\_DISABLE\\_WARNING\\_TRUNCATE\\_DEBUG\\_MESSAGE](#)

#### 7.125.1 Detailed Description

The file [oscl\\_tree.h](#) defines the template class [Oscl\\_Rb\\_Tree](#) which has a very similar API as the STL Tree class. It is an implementation of a Red-Black Tree for use by the [Oscl\\_Map](#) class. Memory allocation is abstracted through the use of an allocator template parameter.

#### 7.125.2 Define Documentation

##### 7.125.2.1 #define OSCL\_DISABLE\_WARNING\_TRUNCATE\_DEBUG\_MESSAGE

## 7.126 oscl\_types.h File Reference

This file contains basic type definitions for common use across platforms.

```
#include "osclconfig.h"
```

### Data Structures

- struct [OsclMemoryFragment](#)

### Typedefs

- [typedef int c\\_bool](#)  
*The c\_bool type is mapped to an integer to provide a bool type for C interfaces.*
- [typedef void OsclAny](#)  
*The OsclAny is meant to be used the context of a generic pointer (i.e., no specific type).*
- [typedef char mbchar](#)  
*mbchar is multi-byte char (e.g., UTF-8) with null termination.*
- [typedef unsigned int uint](#)  
*The uint type is a convenient abbreviation for unsigned int.*
- [typedef uint8 octet](#)  
*The octet type is meant to be used for referring to a byte or collection bytes without suggesting anything about the underlying meaning of the bytes.*
- [typedef float OsclFloat](#)  
*The Float type defined as OsclFloat.*
- [typedef OSCL\\_NATIVE\\_INT64\\_TYPE int64](#)
- [typedef OSCL\\_NATIVE\\_UINT64\\_TYPE uint64](#)
- [typedef OSCL\\_NATIVE\\_WCHAR\\_TYPE oscl\\_wchar](#)
- [typedef oscl\\_wchar OSCL\\_TCHAR](#)  
*define OSCL\_TCHAR*

### 7.126.1 Detailed Description

This file contains basic type definitions for common use across platforms.

## 7.127 oscl\_udp\_socket.h File Reference

```
#include "oscl_ip_socket.h"
#include "oscl_defalloc.h"
#include "oscl_socket_recv_from.h"
#include "oscl_socket_send_to.h"
#include "oscl_socket_bind.h"
```

### Data Structures

- class [OsclUDPSocketI](#)

## 7.128 oscl\_utf8conv.h File Reference

Utilities to convert unicode to utf8 and vice versa.

```
#include "oscl_base.h"
```

### Functions

- OSCL\_IMPORT\_REF int32 [oscl\\_UTF8ToUnicode](#) (const char \*input, int32 inLength, oscl\_wchar \*output, int32 outLength)

*Convert UTF8 byte sequence to Unicode string.*

- OSCL\_IMPORT\_REF int32 [oscl\\_UncodeToUTF8](#) (const oscl\_wchar \*input, int32 inLength, char \*output, int32 outLength)

*Convert Unicode string to UTF8 byte sequence.*

### 7.128.1 Detailed Description

Utilities to convert unicode to utf8 and vice versa.

## 7.129 oscl\_uuid.h File Reference

This file defines the OSCL UUID structure used for unique identifiers as well as the short (32-bit) identifiers OsclUid32.

```
#include "oscl_base.h"
#include "oscl_mem_basic_functions.h"
#include "oscl_string_utils.h"
#include "oscl_stdstring.h"
```

### Data Structures

- struct [OsclUuid](#)

### Defines

- #define [BYTES\\_IN\\_UUID\\_ARRAY](#) 8

### Typedefs

- typedef uint32 [OsclUid32](#)

### Variables

- const char [PV\\_CHAR\\_CLOSE\\_BRACKET](#) = ')
- const char [PV\\_CHAR\\_COMMA](#) = ','

#### 7.129.1 Detailed Description

This file defines the OSCL UUID structure used for unique identifiers as well as the short (32-bit) identifiers OsclUid32.

#### 7.129.2 Define Documentation

##### 7.129.2.1 #define BYTES\_IN\_UUID\_ARRAY 8

#### 7.129.3 Typedef Documentation

##### 7.129.3.1 typedef uint32 OsclUid32

#### 7.129.4 Variable Documentation

##### 7.129.4.1 const char PV\_CHAR\_CLOSE\_BRACKET = ')

##### 7.129.4.2 const char PV\_CHAR\_COMMA = ','

## 7.130 oscl\_vector.h File Reference

The file `oscl_vector.h` defines the template class `Oscl_Vector` which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

```
#include "oscl_mem_basic_functions.h"
#include "oscl_assert.h"
#include "oscl_opaque_type.h"
#include "oscl_defalloc.h"
#include "oscl_base.h"
```

### Data Structures

- class `Oscl_Vector`
- class `Oscl_Vector_Base`

#### 7.130.1 Detailed Description

The file `oscl_vector.h` defines the template class `Oscl_Vector` which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

## 7.131 osclconfig.h File Reference

This file contains configuration information for the linux platform.

```
#include <dirent.h>
#include <dlfcn.h>
#include "osclconfig_limits_typedefs.h"
#include "osclconfig_unix_android.h"
#include "osclconfig_ix86.h"
#include "osclconfig_check.h"
```

### Defines

- #define OSCL\_HAS\_ANDROID\_SUPPORT 1
- #define OSCL\_HAS\_ANDROID\_FILE\_IO\_SUPPORT 1
- #define OSCL\_RELEASE\_BUILD 0
- #define OSCL\_UNSIGNED\_CONST(x) x##u
- #define OSCL\_NATIVE\_UINT64\_TYPE u\_int64\_t
- #define OSCL\_TEMPLATED\_DESTRUCTOR\_CALL(type, simple\_type) ~type()
- #define \_\_TFS\_\_ <>
- #define OSCL\_BEGIN\_PACKED
- #define OSCL\_PACKED\_VAR(x) x \_\_attribute\_\_((packed))
- #define OSCL\_PACKED\_STRUCT\_BEGIN
- #define OSCL\_PACKED\_STRUCT\_END \_\_attribute\_\_((packed))
- #define OSCL\_END\_PACKED
- #define OSCL\_ASSERT\_ALWAYS 0

### 7.131.1 Detailed Description

This file contains configuration information for the linux platform.

## 7.131.2 Define Documentation

7.131.2.1 `#define __TFS__ <>`

7.131.2.2 `#define OSCL_BEGIN_PACKED`

7.131.2.3 `#define OSCL_END_PACKED`

7.131.2.4 `#define OSCL_HAS_ANDROID_FILE_IO_SUPPORT 1`

7.131.2.5 `#define OSCL_HAS_ANDROID_SUPPORT 1`

7.131.2.6 `#define OSCL_NATIVE_UINT64_TYPE u_int64_t`

7.131.2.7 `#define OSCL_PACKED_STRUCT_BEGIN`

7.131.2.8 `#define OSCL_PACKED_STRUCT_END __attribute__((packed))`

7.131.2.9 `#define OSCL_PACKED_VAR(x) x __attribute__((packed))`

7.131.2.10 `#define OSCL_RELEASE_BUILD 0`

7.131.2.11 `#define OSCL_TEMPLATED_DESTRUCTOR_CALL(type, simple_type) ~type()`

7.131.2.12 `#define OSCL_UNSIGNED_CONST(x) x##u`

## 7.132 osclconfig\_ansi\_memory.h File Reference

This file contains common typedefs based on the ANSI C limits.h header.

```
#include <memory.h>
```

### Defines

- #define OSCL\_HAS\_ANSI\_MEMORY\_FUNCS 1

### Typedefs

- typedef size\_t oscl\_memsize\_t

#### 7.132.1 Detailed Description

This file contains common typedefs based on the ANSI C limits.h header.

This header file should work for any ANSI C compiler to determine the proper native C types to use for OSCL integer types.

#### 7.132.2 Define Documentation

7.132.2.1 #define OSCL\_HAS\_ANSI\_MEMORY\_FUNCS 1

#### 7.132.3 Typedef Documentation

7.132.3.1 typedef size\_t oscl\_memsize\_t

## 7.133 osclconfig\_check.h File Reference

### Typedefs

- `typedef int8 __int8__check__`
- `typedef uint8 __uint8__check__`
- `typedef int16 __int16__check__`
- `typedef uint16 __uint16__check__`
- `typedef int32 __int32__check__`
- `typedef uint32 __uint32__check__`

## 7.134 osclconfig\_compiler\_warnings.h File Reference

This file contains the ability to turn off/on compiler warnings.

### Defines

- #define OSCL\_FUNCTION\_PTR(x) (&x)

#### 7.134.1 Detailed Description

This file contains the ability to turn off/on compiler warnings.

#### 7.134.2 Define Documentation

##### 7.134.2.1 #define OSCL\_FUNCTION\_PTR(x) (&x)

## 7.135 osclconfig\_error.h File Reference

This file contains the common typedefs and header files needed to compile osclerror.

```
#include "osclconfig.h"  
#include <setjmp.h>  
#include <errno.h>  
#include "osclconfig_error_check.h"
```

### Defines

- #define OSCL\_HAS\_EXCEPTIONS 1
- #define OSCL\_HAS\_ERRNO\_H 1
- #define OSCL\_HAS\_SYMBIAN\_ERRORTRAP 0
- #define OSCL\_HAS\_SETJMP\_H 1

### 7.135.1 Detailed Description

This file contains the common typedefs and header files needed to compile osclerror.

### 7.135.2 Define Documentation

- 7.135.2.1 #define OSCL\_HAS\_ERRNO\_H 1
- 7.135.2.2 #define OSCL\_HAS\_EXCEPTIONS 1
- 7.135.2.3 #define OSCL\_HAS\_SETJMP\_H 1
- 7.135.2.4 #define OSCL\_HAS\_SYMBIAN\_ERRORTRAP 0

## **7.136 osclconfig\_error\_check.h File Reference**

## **7.137 osclconfig\_global\_new\_delete.h File Reference**

### **Functions**

- void \* **operator new** (size\_t)
- void **operator delete** (void \*)

## 7.138 osclconfig\_global\_placement\_new.h File Reference

### Functions

- void \* [operator new](#) (size\_t, void \*ptr)

#### 7.138.1 Function Documentation

##### 7.138.1.1 void\* operator new (size\_t, void \*ptr) [inline]

## 7.139 osclconfig\_io.h File Reference

This file contains common typedefs based on the ANSI C limits.h header.

```
#include "osclconfig.h"
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <fcntl.h>
#include <signal.h>
#include <netdb.h>
#include <sys/mman.h>
#include <sys/types.h>
#include <errno.h>
#include <sys/vfs.h>
#include <dirent.h>
#include <sys/stat.h>
#include "osclconfig_io_check.h"
```

### Defines

- #define OSCL\_HAS\_GLOB 0
- #define OSCL\_HAS\_ANSI\_FILE\_IO\_SUPPORT 1
- #define OSCL\_HAS\_ANSI\_64BIT\_FILE\_IO\_SUPPORT 0
- #define OSCL\_HAS\_MSWIN\_FILE\_IO\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_COMPATIBLE\_IO\_FUNCTION 0
- #define OSCL\_HAS\_NATIVE\_FILE\_CACHE\_ENABLE 1
- #define OSCL\_FILE\_BUFFER\_MAX\_SIZE 32768
- #define OSCL\_HAS\_PV\_FILE\_CACHE 0
- #define OSCL\_HAS\_LARGE\_FILE\_SUPPORT 1
- #define OSCL\_HAS\_SYMBIAN\_SOCKET\_SERVER 0
- #define OSCL\_HAS\_SYMBIAN\_DNS\_SERVER 0
- #define OSCL\_HAS\_BERKELEY\_SOCKETS 1
- #define OSCL\_HAS\_SOCKET\_SUPPORT 1
- #define OsclValidInetAddr(addr) (inet\_addr(addr)!=INADDR\_NONE)
- #define OsclMakeSockAddr(sockaddr, port, addrstr, ok)
- #define OsclUnMakeSockAddr(sockaddr, addrstr) addrstr=inet\_ntoa(sockaddr.sin\_addr);
- #define OsclSetRecvBufferSize(s, val, ok, err)
- #define OsclBind(s, addr, ok, err)
- #define OsclJoin(s, addr, ok, err)
- #define OsclListen(s, size, ok, err)

- #define `OsclAccept`(s, accept\_s, ok, err, wouldblock)
- #define `OsclSetNonBlocking`(s, ok, err)
- #define `OsclShutdown`(s, how, ok, err)
- #define `OsclSocket`(s, fam, type, prot, ok, err)
- #define `OsclSendTo`(s, buf, len, addr, ok, err, nbytes, wouldblock)
- #define `OsclSend`(s, buf, len, ok, err, nbytes, wouldblock)
- #define `OsclCloseSocket`(s, ok, err)
- #define `OsclConnect`(s, addr, ok, err, wouldblock)
- #define `OsclGetAsyncSockErr`(s, ok, err)
- #define `OsclConnectComplete`(s, wset, eset, success, fail, ok, err)
- #define `OsclRecv`(s, buf, len, ok, err, nbytes, wouldblock)
- #define `OsclRecvFrom`(s, buf, len, paddr, paddrlen, ok, err, nbytes, wouldblock)
- #define `OsclSocketSelect`(nfds, rd, wr, ex, timeout, ok, err, nhandles)
- #define `OsclSocketStartup`(ok)
- #define `OsclSocketCleanup`(ok)
- #define `OsclGethostbyname`(name, hostent, ok, err)
- #define `OsclGetDottedAddr`(hostent, dottedaddr, ok)
- #define `OSCL_SD_RECEIVE` SHUT\_RD
- #define `OSCL_SD_SEND` SHUT\_WR
- #define `OSCL_SD_BOTH` SHUT\_RDWR
- #define `OSCL_AF_INET` AF\_INET
- #define `OSCL_SOCK_STREAM` SOCK\_STREAM
- #define `OSCL_SOCK_DGRAM` SOCK\_DGRAM
- #define `OSCL IPPROTO_TCP` IPPROTO\_TCP
- #define `OSCL IPPROTO_UDP` IPPROTO\_UDP
- #define `_FILE_OFFSET_BITS` 64

## Typedefs

- typedef int `TOsclSocket`
- typedef sockaddr\_in `TOsclSockAddr`
- typedef socklen\_t `TOsclSockAddrLen`
- typedef hostent `TOsclHostent`
- typedef off\_t `TOsclFileOffset`

### 7.139.1 Detailed Description

This file contains common typedefs based on the ANSI C limits.h header.

This header file should work for any ANSI C compiler to determine the proper native C types to use for OSCL integer types.

## 7.139.2 Define Documentation

- 7.139.2.1 `#define _FILE_OFFSET_BITS 64`
- 7.139.2.2 `#define OSCL_AF_INET AF_INET`
- 7.139.2.3 `#define OSCL_FILE_BUFFER_MAX_SIZE 32768`
- 7.139.2.4 `#define OSCL_HAS_ANSI_64BIT_FILE_IO_SUPPORT 0`
- 7.139.2.5 `#define OSCL_HAS_ANSI_FILE_IO_SUPPORT 1`
- 7.139.2.6 `#define OSCL_HAS_BERKELEY_SOCKETS 1`
- 7.139.2.7 `#define OSCL_HAS_GLOB 0`
- 7.139.2.8 `#define OSCL_HAS_LARGE_FILE_SUPPORT 1`
- 7.139.2.9 `#define OSCL_HAS_MSWIN_FILE_IO_SUPPORT 0`
- 7.139.2.10 `#define OSCL_HAS_NATIVE_FILE_CACHE_ENABLE 1`
- 7.139.2.11 `#define OSCL_HAS_PV_FILE_CACHE 0`
- 7.139.2.12 `#define OSCL_HAS_SOCKET_SUPPORT 1`
- 7.139.2.13 `#define OSCL_HAS_SYMBIAN_COMPATIBLE_IO_FUNCTION 0`
- 7.139.2.14 `#define OSCL_HAS_SYMBIAN_DNS_SERVER 0`
- 7.139.2.15 `#define OSCL_HAS_SYMBIAN_SOCKET_SERVER 0`
- 7.139.2.16 `#define OSCL IPPROTO_TCP IPPROTO_TCP`
- 7.139.2.17 `#define OSCL IPPROTO_UDP IPPROTO_UDP`
- 7.139.2.18 `#define OSCL_SD_BOTH SHUT_RDWR`
- 7.139.2.19 `#define OSCL_SD_RECEIVE SHUT_RD`
- 7.139.2.20 `#define OSCL_SD_SEND SHUT_WR`
- 7.139.2.21 `#define OSCL SOCK_DGRAM SOCK_DGRAM`
- 7.139.2.22 `#define OSCL SOCK_STREAM SOCK_STREAM`
- 7.139.2.23 `#define OsclAccept(s, accept_s, ok, err, wouldblock)`

**Value:**

```
accept_s=accept(s,NULL,NULL); \
ok=(accept_s!=(-1)); \
if (!ok){err=errno;wouldblock=(err==EAGAIN||err==EWOULDBLOCK);} 
```

**7.139.2.24 #define OsclBind(s, addr, ok, err)**

**Value:**

```
TOsclSockAddr* tmpadr = &addr;\n    sockaddr* sadr = OSCL_STATIC_CAST(sockaddr*, tmpadr);\n    ok=(bind(s,sadr,sizeof(addr))!=(-1));\n    if (!ok)err=errno
```

**7.139.2.25 #define OsclCloseSocket(s, ok, err)**

**Value:**

```
ok=(close(s)!=(-1));\n    if (!ok)err=errno
```

**7.139.2.26 #define OsclConnect(s, addr, ok, err, wouldblock)**

**Value:**

```
TOsclSockAddr* tmpadr = &addr;\n    sockaddr* sadr = OSCL_STATIC_CAST(sockaddr*, tmpadr);\n    ok=(connect(s,sadr,sizeof(addr))!=(-1));\n    if (!ok){err=errno;wouldblock=(err==EINPROGRESS);}
```

**7.139.2.27 #define OsclConnectComplete(s, wset, eset, success, fail, ok, err)**

**Value:**

```
success=fail=false;\n    if (FD_ISSET(s,&eset))\\\n        {fail=true;OsclGetAsyncSockErr(s,ok,err);}\n    else if (FD_ISSET(s,&wset))\\\n        {OsclGetAsyncSockErr(s,ok,err);if (ok && err==0)success=true;else fail=true;}
```

**7.139.2.28 #define OsclGetAsyncSockErr(s, ok, err)**

**Value:**

```
int opterr;socklen_t optlen(sizeof(opterr));\n    ok=(getsockopt(s,SOL_SOCKET,SO_ERROR,(void *)&opterr,&optlen)!=(-1));\n    if(ok)err=opterr;else err=errno;
```

**7.139.2.29 #define OsclGetDottedAddr(hostent, dottedaddr, ok)**

**Value:**

```
long *_hostaddr=(long*)hostent->h_addr_list[0];\n    struct in_addr _inaddr;\n    _inaddr.s_addr=* _hostaddr;\n    dottedaddr/inet_ntoa(_inaddr);\n    ok=(dottedaddr!=NULL);
```

**7.139.2.30 #define OsclGethostbyname(name, hostent, ok, err)**

**Value:**

```
hostent=gethostbyname((const char*)name); \
ok=(hostent!=NULL); \
if (!ok)err=errno;
```

**7.139.2.31 #define OsclJoin(s, addr, ok, err)**

**Value:**

```
{ \
    struct ip_mreq mreq; \
    void* p = &addr; \
ok=(bind(s,(sockaddr*)p,sizeof(addr))!=(-1)); \
mreq.imr_multiaddr.s_addr = addr.sin_addr.s_addr ; \
mreq.imr_interface.s_addr = htonl(INADDR_ANY); \
ok=(setsockopt(s, IPPROTO_IP, IP_ADD_MEMBERSHIP, &mreq, sizeof(struct ip_mreq))!=(-1)); \
if (!ok)err=errno; \
}
```

**7.139.2.32 #define OsclListen(s, size, ok, err)**

**Value:**

```
ok=(listen(iSocket,qSize)!=(-1)); \
if (!ok)err=errno
```

**7.139.2.33 #define OsclMakeSockAddr(sockaddr, port, addrstr, ok)**

**Value:**

```
sockaddr.sin_family=OSCL_AF_INET; \
sockaddr.sin_port=htons(port); \
int32 result/inet_aton((const char*)addrstr,&sockaddr.sin_addr); \
ok=(result!=0);
```

**7.139.2.34 #define OsclRecv(s, buf, len, ok, err, nbytes, wouldblock)**

**Value:**

```
nbytes=recv(s,(void *)buf),(size_t)(len),0); \
ok=(nbytes!=(-1)); \
if (!ok){err=errno;wouldblock=(err==EAGAIN);}
```

**7.139.2.35 #define OsclRecvFrom(s, buf, len, paddr, paddrlen, ok, err, nbytes, wouldblock)**

**Value:**

```
\ 
void* p=paddr;\n
nbytes=recvfrom(s,(void*)(buf),(size_t)(len),0,(struct sockaddr*)p,paddrlen);\n
ok=(nbytes!=(-1));\n
if (!ok){err=errno;wouldblock=(err==EAGAIN);}\n
}
```

**7.139.2.36 #define OsclSend(s, buf, len, ok, err, nbytes, wouldblock)**

**Value:**

```
nbytes=send(s,(const void*)(buf),(size_t)(len),0);\n
ok=(nbytes!=(-1));\n
if (!ok){err=errno;wouldblock=(err==EAGAIN||err==EWOULDBLOCK);}
```

**7.139.2.37 #define OsclSendTo(s, buf, len, addr, ok, err, nbytes, wouldblock)**

**Value:**

```
TOsclSockAddr* tmpadr = &addr;\n
sockaddr* sadr = OSCL_STATIC_CAST(sockaddr*, tmpadr);\n
nbytes=sendto(s,(const void*)(buf),(size_t)(len),0,sadr,(socklen_t)sizeof(addr));\n
ok=(nbytes!=(-1));\n
if (!ok){err=errno;wouldblock=(err==EAGAIN||err==EWOULDBLOCK);}
```

**7.139.2.38 #define OsclSetNonBlocking(s, ok, err)**

**Value:**

```
ok=(fcntl(s,F_SETFL,O_NONBLOCK)!=(-1));\n
if (!ok)err=errno
```

**7.139.2.39 #define OsclSetRecvBufferSize(s, val, ok, err)**

**Value:**

```
ok=(setsockopt(s,SOL_SOCKET,SO_RCVBUF,(char*)&val, sizeof(int)) !=-1);\n
if (!ok)err=errno
```

**7.139.2.40 #define OsclShutdown(s, how, ok, err)**

**Value:**

```
ok=(shutdown(iSocket,how)!=(-1));\n
if (!ok)err=errno
```

**7.139.2.41 #define OsclSocket(s, fam, type, prot, ok, err)****Value:**

```
s=socket(fam,type,prot); \
ok=(s!=(-1)); \
if (!ok)err=errno
```

**7.139.2.42 #define OsclSocketCleanup(ok)****Value:**

```
signal(SIGPIPE,SIG_DFL); \
ok=true
```

**7.139.2.43 #define OsclSocketSelect(nfds, rd, wr, ex, timeout, ok, err, nhandles)****Value:**

```
nhandles=select(nfds,&rd,&wr,&ex,&timeout); \
ok=(nhandles!=(-1)); \
if (!ok)err=errno
```

**7.139.2.44 #define OsclSocketStartup(ok)****Value:**

```
signal(SIGPIPE,SIG_IGN); \
ok=true
```

**7.139.2.45 #define OsclUnMakeSockAddr(sockaddr, addrstr) addrstr=inet\_ntoa(sockaddr.sin\_addr);****7.139.2.46 #define OsclValidInetAddr(addr) (inet\_addr(addr)!=INADDR\_NONE)**

## 7.139.3 Typedef Documentation

**7.139.3.1 typedef off\_t TOsclFileOffset****7.139.3.2 typedef struct hostent TOsclHostent****7.139.3.3 typedef struct sockaddr\_in TOsclSockAddr****7.139.3.4 typedef socklen\_t TOsclSockAddrLen****7.139.3.5 typedef int TOsclSocket**

## 7.140 osclconfig\_io\_check.h File Reference

### Typedefs

- `typedef TOsclFileOffset __verify__TOsclFileOffset__defined__`

#### 7.140.1 Typedef Documentation

##### 7.140.1.1 `typedef TOsclFileOffset __verify__TOsclFileOffset__defined__`

type TOsclFileOffset should be defined as the type used for file size and offsets on the target platform.  
Example: `typedef size_t TOsclFileOffset;`

## 7.141 osclconfig\_ix86.h File Reference

This file contains configuration information for the ix86 processor family.

### Defines

- #define OSCL\_INTEGERS\_WORD\_ALIGNED 1
- #define OSCL\_BYTE\_ORDER\_BIG\_ENDIAN 0
- #define OSCL\_BYTE\_ORDER\_LITTLE\_ENDIAN 1

### 7.141.1 Detailed Description

This file contains configuration information for the ix86 processor family.

## 7.142 osclconfig\_lib.h File Reference

This file contains configuration information for the ANSI build.

```
#include "osclconfig_lib_check.h"
```

### Defines

- #define OSCL\_HAS\_RUNTIME\_LIB\_LOADING\_SUPPORT 1
- #define PV\_RUNTIME\_LIB\_FILENAME\_EXTENSION "so"
- #define OSCL\_LIB\_READ\_DEBUG\_LIBS 1
- #define PV\_DYNAMIC\_LOADING\_CONFIG\_FILE\_PATH "./"

#### 7.142.1 Detailed Description

This file contains configuration information for the ANSI build.

#### 7.142.2 Define Documentation

7.142.2.1 #define OSCL\_HAS\_RUNTIME\_LIB\_LOADING\_SUPPORT 1

7.142.2.2 #define OSCL\_LIB\_READ\_DEBUG\_LIBS 1

7.142.2.3 #define PV\_DYNAMIC\_LOADING\_CONFIG\_FILE\_PATH "./"

7.142.2.4 #define PV\_RUNTIME\_LIB\_FILENAME\_EXTENSION "so"

## **7.143 osclconfig\_lib\_check.h File Reference**

## 7.144 osclconfig\_limits\_typedefs.h File Reference

This file contains common typedefs based on the ANSI C limits.h header.

```
#include <limits.h>
```

### Defines

- #define OSCL\_CHAR\_IS\_UNSIGNED 1
- #define OSCL\_CHAR\_IS\_SIGNED 0

### 7.144.1 Detailed Description

This file contains common typedefs based on the ANSI C limits.h header.

This header file should work for any ANSI C compiler to determine the proper native C types to use for OSCL integer types.

### 7.144.2 Define Documentation

**7.144.2.1 #define OSCL\_CHAR\_IS\_SIGNED 0**

**7.144.2.2 #define OSCL\_CHAR\_IS\_UNSIGNED 1**

## 7.145 osclconfig\_memory.h File Reference

```
#include "osclconfig.h"
#include "osclconfig_ansi_memory.h"
#include "osclconfig_memory_check.h"
```

### Defines

- #define OSCL\_BYPASS\_MEMMGT 1
- #define OSCL\_HAS\_GLOBAL\_NEW\_DELETE 1
- #define PVMEM\_INST\_LEVEL 1
- #define OSCL\_HAS\_HEAP\_BASE\_SUPPORT 1
- #define OSCL\_HAS\_SYMBIAN\_MEMORY\_FUNCS 0

### 7.145.1 Define Documentation

7.145.1.1 #define OSCL\_BYPASS\_MEMMGT 1

7.145.1.2 #define OSCL\_HAS\_GLOBAL\_NEW\_DELETE 1

7.145.1.3 #define OSCL\_HAS\_HEAP\_BASE\_SUPPORT 1

7.145.1.4 #define OSCL\_HAS\_SYMBIAN\_MEMORY\_FUNCS 0

7.145.1.5 #define PVMEM\_INST\_LEVEL 1

---

## **7.146 osclconfig\_memory\_check.h File Reference**

## 7.147 osclconfig\_no\_os.h File Reference

### Defines

- #define OSCL\_HAS\_UNIX\_SUPPORT 0
- #define OSCL\_HAS\_MSWIN\_SUPPORT 0
- #define OSCL\_HAS\_MSWIN\_PARTIAL\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_SUPPORT 0
- #define OSCL\_HAS\_SAVAJE\_SUPPORT 0
- #define OSCL\_HAS\_PV\_C\_OS\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_ERRORTRAP 0
- #define OSCL\_HAS\_SYMBIAN\_MEMORY\_FUNCS 0
- #define OSCL\_HAS\_PV\_C\_OS\_API\_MEMORY\_FUNCS 0
- #define OSCL\_HAS\_PV\_C\_OS\_TIME\_FUNCS 0
- #define OSCL\_HAS\_UNIX\_TIME\_FUNCS 0
- #define OSCL\_HAS\_SYMBIAN\_TIMERS 0
- #define OSCL\_HAS\_SYMBIAN\_MATH 0
- #define OSCL\_HAS\_SYMBIAN\_SCHEDULER 0
- #define OSCL\_HAS\_SEM\_TIMEDWAIT\_SUPPORT 0
- #define OSCL\_HAS\_PTHREAD\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_COMPATIBLE\_IO\_FUNCTION 0
- #define OSCL\_HAS\_SAVAJE\_IO\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_SOCKET\_SERVER 0
- #define OSCL\_HAS\_SYMBIAN\_DNS\_SERVER 0
- #define OSCL\_HAS\_BERKELEY\_SOCKETS 0

## **7.148 osclconfig\_proc.h File Reference**

This file contains configuration information for the linux platform.

```
#include "osclconfig.h"  
#include "osclconfig_proc_unix_android.h"  
#include "osclconfig_proc_check.h"
```

### **7.148.1 Detailed Description**

This file contains configuration information for the linux platform.

## 7.149 osclconfig\_proc\_check.h File Reference

### Typedefs

- `typedef TOsclThreadId __verify__TOsclThreadId_defined__`
- `typedef TOsclThreadFuncRet __verify__TOsclThreadFuncRet_defined__`
- `typedef TOsclThreadFuncArg __verify__TOsclThreadFuncArg_defined__`
- `typedef TOsclThreadObject __verify__TOsclThreadObject_defined__`
- `typedef TOsclMutexObject __verify__TOsclMutexObject_defined__`
- `typedef TOsclSemaphoreObject __verify__TOsclSemaphoreObject_defined__`
- `typedef TOsclConditionObject __verify__TOsclConditionObject_defined__`

#### 7.149.1 Typedef Documentation

##### 7.149.1.1 `typedef TOsclConditionObject __verify__TOsclConditionObject_defined__`

type TOsclConditionObject should be defined as the type used as a condition variable on the target platform.  
Example: `typedef pthread_cond_t TOsclConditionObject;`

Note: Condition variables are only used with certain semaphore implementations. If the semaphore implementation does not require a condition variable, then this type can be defined as 'int' as follows: `typedef int TOsclConditionObject; //not used`

##### 7.149.1.2 `typedef TOsclMutexObject __verify__TOsclMutexObject_defined__`

type TOsclMutexObject should be defined as the type used as a mutex object or handle on the target platform. Example: `typedef pthread_mutex_t TOsclMutexObject;`

##### 7.149.1.3 `typedef TOsclSemaphoreObject __verify__TOsclSemaphoreObject_defined__`

type TOsclSemaphoreObject should be defined as the type used as a mutex object or handle on the target platform. Example: `typedef sem_t TOsclSemaphoreObject;`

##### 7.149.1.4 `typedef TOsclThreadFuncArg __verify__TOsclThreadFuncArg_defined__`

type TOsclThreadFuncArg should be defined as the type used as a thread function argument on the target platform. Example: `typedef LPVOID TOsclThreadFuncArg;`

##### 7.149.1.5 `typedef TOsclThreadFuncRet __verify__TOsclThreadFuncRet_defined__`

type TOsclThreadFuncRet should be defined as the type used as a thread function return value on the target platform. Example: `typedef DWORD TOsclThreadFuncRet;`

##### 7.149.1.6 `typedef TOsclThreadId __verify__TOsclThreadId_defined__`

type TOsclThreadId should be defined as the type used as a thread ID on the target platform. Example: `typedef DWORD TOsclThreadId;`

#### 7.149.1.7 **typedef TOsclThreadObject \_\_verify\_\_TOsclThreadObject\_defined\_\_**

type TOsclThreadObject should be defined as the type used as a thread object or handle on the target platform. Example: `typedef pthread_t TOsclThreadObject;`

## 7.150 osclconfig\_proc\_unix\_android.h File Reference

```
#include <pthread.h>
#include <errno.h>
#include <signal.h>
```

### Defines

- #define OSCL\_HAS\_SYMBIAN\_SCHEDULER 0
- #define OSCL\_HAS\_THREAD\_SUPPORT 1
- #define OSCL\_HAS\_NON\_PREEMPTIVE\_THREAD\_SUPPORT 0
- #define OSCL\_HAS\_SEM\_TIMEDWAIT\_SUPPORT 0
- #define OSCL\_HAS\_PTHREAD\_SUPPORT 1
- #define OSCL\_THREAD\_DECL

### TypeDefs

- typedef pthread\_t TOsclThreadId
- typedef void \* TOsclThreadFuncArg
- typedef void \* TOsclThreadFuncRet
- typedef pthread\_t TOsclThreadObject
- typedef pthread\_mutex\_t TOsclMutexObject
- typedef int TOsclSemaphoreObject
- typedef pthread\_cond\_t TOsclConditionObject

### 7.150.1 Define Documentation

- 7.150.1.1 `#define OSCL_HAS_NON_PREEMPTIVE_THREAD_SUPPORT 0`
- 7.150.1.2 `#define OSCL_HAS_PTHREAD_SUPPORT 1`
- 7.150.1.3 `#define OSCL_HAS_SEM_TIMEDWAIT_SUPPORT 0`
- 7.150.1.4 `#define OSCL_HAS_SYMBIAN_SCHEDULER 0`
- 7.150.1.5 `#define OSCL_HAS_THREAD_SUPPORT 1`
- 7.150.1.6 `#define OSCL_THREAD_DECL`

### 7.150.2 Typedef Documentation

- 7.150.2.1 `typedef pthread_cond_t TOsclConditionObject`
- 7.150.2.2 `typedef pthread_mutex_t TOsclMutexObject`
- 7.150.2.3 `typedef int TOsclSemaphoreObject`
- 7.150.2.4 `typedef void* TOsclThreadFuncArg`
- 7.150.2.5 `typedef void* TOsclThreadFuncRet`
- 7.150.2.6 `typedef pthread_t TOsclThreadId`
- 7.150.2.7 `typedef pthread_t TOsclThreadObject`

## 7.151 osclconfig\_proc\_unix\_common.h File Reference

```
#include <time.h>
#include <semaphore.h>
#include <pthread.h>
#include <errno.h>
```

### Defines

- #define OSCL\_HAS\_SYMBIAN\_SCHEDULER 0
- #define OSCL\_HAS\_THREAD\_SUPPORT 1
- #define OSCL\_HAS\_NON\_PREEMPTIVE\_THREAD\_SUPPORT 0
- #define OSCL\_HAS\_SEM\_TIMEDWAIT\_SUPPORT 1
- #define OSCL\_HAS\_PTHREAD\_SUPPORT 1
- #define OSCL\_THREAD\_DECL

### Typedefs

- typedef pthread\_t TOsclThreadId
- typedef void \* TOsclThreadFuncArg
- typedef void \* TOsclThreadFuncRet
- typedef pthread\_t TOsclThreadObject
- typedef pthread\_mutex\_t TOsclMutexObject
- typedef sem\_t TOsclSemaphoreObject
- typedef pthread\_cond\_t TOsclConditionObject

### 7.151.1 Define Documentation

- 7.151.1.1 `#define OSCL_HAS_NON_PREEMPTIVE_THREAD_SUPPORT 0`
- 7.151.1.2 `#define OSCL_HAS_PTHREAD_SUPPORT 1`
- 7.151.1.3 `#define OSCL_HAS_SEM_TIMEDWAIT_SUPPORT 1`
- 7.151.1.4 `#define OSCL_HAS_SYMBIAN_SCHEDULER 0`
- 7.151.1.5 `#define OSCL_HAS_THREAD_SUPPORT 1`
- 7.151.1.6 `#define OSCL_THREAD_DECL`

### 7.151.2 Typedef Documentation

- 7.151.2.1 `typedef pthread_cond_t TOsclConditionObject`
- 7.151.2.2 `typedef pthread_mutex_t TOsclMutexObject`
- 7.151.2.3 `typedef sem_t TOsclSemaphoreObject`
- 7.151.2.4 `typedef void* TOsclThreadFuncArg`
- 7.151.2.5 `typedef void* TOsclThreadFuncRet`
- 7.151.2.6 `typedef pthread_t TOsclThreadId`
- 7.151.2.7 `typedef pthread_t TOsclThreadObject`

## 7.152 osclconfig\_time.h File Reference

```
#include "osclconfig.h"
#include <time.h>
#include <sys/time.h>
#include <unistd.h>
#include "osclconfig_time_check.h"
```

### Defines

- #define OSCL\_HAS\_UNIX\_TIME\_FUNCS 1

### TypeDefs

- typedef timeval OsclBasicTimeStruct
- typedef tm OsclBasicDateTimeStruct

#### 7.152.1 Define Documentation

7.152.1.1 #define OSCL\_HAS\_UNIX\_TIME\_FUNCS 1

#### 7.152.2 Typedef Documentation

7.152.2.1 typedef tm OsclBasicDateTimeStruct

7.152.2.2 typedef struct timeval OsclBasicTimeStruct

## 7.153 osclconfig\_time\_check.h File Reference

### Typedefs

- `typedef OsclBasicTimeStruct __Validate__BasicTimeStruct__`
- `typedef OsclBasicDateTimeStruct __Validate__BasicTimeDateStruct__`

#### 7.153.1 Typedef Documentation

##### 7.153.1.1 `typedef OsclBasicDateTimeStruct __Validate__BasicTimeDateStruct__`

`OsclBasicDateTimeStruct` type should be defined to the platform-specific date + time type.

##### 7.153.1.2 `typedef OsclBasicTimeStruct __Validate__BasicTimeStruct__`

`OsclBasicTimeStruct` type should be defined to the platform-specific time of day type.

## 7.154 osclconfig\_unix\_android.h File Reference

```
#include <stdlib.h>
#include <stdarg.h>
#include <sys/types.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <pthread.h>
#include <ctype.h>
#include <math.h>
```

### Defines

- #define OSCL\_DISABLE\_INLINES 0
- #define OSCL\_HAS\_ANSI\_STDLIB\_SUPPORT 1
- #define OSCL\_HAS\_ANSI\_MATH\_SUPPORT 1
- #define OSCL\_HAS\_GLOBAL\_VARIABLE\_SUPPORT 1
- #define OSCL\_HAS\_ANSI\_STRING\_SUPPORT 1
- #define OSCL\_HAS\_ANSI\_WIDE\_STRING\_SUPPORT 0
- #define OSCL\_HAS\_ANSI\_STDIO\_SUPPORT 1
- #define OSCL\_MEMFRAG\_PTR\_BEFORE\_LEN 1
- #define OSCL\_HAS\_UNIX\_SUPPORT 1
- #define OSCL\_HAS\_MSWIN\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_SUPPORT 0
- #define OSCL\_HAS\_NATIVE\_INT64\_TYPE 1
- #define OSCL\_HAS\_NATIVE\_UINT64\_TYPE 1
- #define OSCL\_NATIVE\_INT64\_TYPE int64\_t
- #define OSCL\_NATIVE\_UINT64\_TYPE uint64\_t
- #define INT64(x) x##LL
- #define UINT64(x) x##ULL
- #define INT64\_HILO(high, low) (((high##LL))<<32)|low)
- #define UINT64\_HILO(high, low) (((high##ULL))<<32)|low)
- #define OSCL\_HAS\_UNICODE\_SUPPORT 1
- #define OSCL\_NATIVE\_WCHAR\_TYPE wchar\_t
- #define \_STRLIT(x) L ## x
- #define \_STRLIT\_CHAR(x) x
- #define \_STRLIT\_WCHAR(x) L ## x
- #define OSCL\_HAS\_TLS\_SUPPORT 1
- #define OSCL\_TLS\_IS\_KEYED 1
- #define OSCL\_TLS\_KEY\_CREATE\_FUNC(key) (pthread\_key\_create(&key,NULL)==0)
- #define OSCL\_TLS\_KEY\_DELETE\_FUNC(key) pthread\_key\_delete(key)
- #define OSCL\_TLS\_STORE\_FUNC(key, ptr) (pthread\_setspecific(key,(const void\*)ptr)==0)
- #define OSCL\_TLS\_GET\_FUNC(key) pthread\_getspecific(key)
- #define OSCL\_HAS\_BASIC\_LOCK 1

## Typedefs

- `typedef pthread_key_t TOsclTlsKey`
- `typedef pthread_mutex_t TOsclBasicLockObject`



### 7.154.1 Define Documentation

- 7.154.1.1 `#define _STRLIT(x) L ## x`
- 7.154.1.2 `#define _STRLIT_CHAR(x) x`
- 7.154.1.3 `#define _STRLIT_WCHAR(x) L ## x`
- 7.154.1.4 `#define INT64(x) x##LL`
- 7.154.1.5 `#define INT64_HILO(high, low) (((high##LL))<<32)|low)`
- 7.154.1.6 `#define OSCL_DISABLE_INLINES 0`
- 7.154.1.7 `#define OSCL_HAS_ANSI_MATH_SUPPORT 1`
- 7.154.1.8 `#define OSCL_HAS_ANSI_STDIO_SUPPORT 1`
- 7.154.1.9 `#define OSCL_HAS_ANSI_STDLIB_SUPPORT 1`
- 7.154.1.10 `#define OSCL_HAS_ANSI_STRING_SUPPORT 1`
- 7.154.1.11 `#define OSCL_HAS_ANSI_WIDE_STRING_SUPPORT 0`
- 7.154.1.12 `#define OSCL_HAS_BASIC_LOCK 1`
- 7.154.1.13 `#define OSCL_HAS_GLOBAL_VARIABLE_SUPPORT 1`
- 7.154.1.14 `#define OSCL_HAS_MSWIN_SUPPORT 0`
- 7.154.1.15 `#define OSCL_HAS_NATIVE_INT64_TYPE 1`
- 7.154.1.16 `#define OSCL_HAS_NATIVE_UINT64_TYPE 1`
- 7.154.1.17 `#define OSCL_HAS_SYMBIAN_SUPPORT 0`
- 7.154.1.18 `#define OSCL_HAS_TLS_SUPPORT 1`
- 7.154.1.19 `#define OSCL_HAS_UNICODE_SUPPORT 1`
- 7.154.1.20 `#define OSCL_HAS_UNIX_SUPPORT 1`
- 7.154.1.21 `#define OSCL_MEMFRAG_PTR_BEFORE_LEN 1`
- 7.154.1.22 `#define OSCL_NATIVE_INT64_TYPE int64_t`
- 7.154.1.23 `#define OSCL_NATIVE_UINT64_TYPE uint64_t`
- 7.154.1.24 `#define OSCL_NATIVE_WCHAR_TYPE wchar_t`
- 7.154.1.25 `#define OSCL_TLS_GET_FUNC(key) pthread_getspecific(key)`
- 7.154.1.26 `#define OSCL_TLS_IS_KEYED 1`
- 7.154.1.27 `#define OSCL_TLS_KEY_CREATE_FUNC(key) (pthread_key_create(&key,NULL)==0)`

## 7.155 osclconfig\_unix\_common.h File Reference

```
#include <stdlib.h>
#include <stdarg.h>
#include <sys/types.h>
#include <stdio.h>
#include <wchar.h>
#include <string.h>
#include <unistd.h>
#include <pthread.h>
#include <ctype.h>
#include <math.h>
```

### Defines

- #define OSCL\_DISABLE\_INLINES 0
- #define OSCL\_HAS\_ANSI\_STDLIB\_SUPPORT 1
- #define OSCL\_HAS\_ANSI\_MATH\_SUPPORT 1
- #define OSCL\_HAS\_GLOBAL\_VARIABLE\_SUPPORT 1
- #define OSCL\_HAS\_ANSI\_STRING\_SUPPORT 1
- #define OSCL\_HAS\_ANSI\_WIDE\_STRING\_SUPPORT 1
- #define OSCL\_HAS\_ANSI\_STDIO\_SUPPORT 1
- #define OSCL\_MEMFRAG\_PTR\_BEFORE\_LEN 1
- #define OSCL\_HAS\_UNIX\_SUPPORT 1
- #define OSCL\_HAS\_MSWIN\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_SUPPORT 0
- #define OSCL\_HAS\_NATIVE\_INT64\_TYPE 1
- #define OSCL\_HAS\_NATIVE\_UINT64\_TYPE 1
- #define OSCL\_NATIVE\_INT64\_TYPE int64\_t
- #define OSCL\_NATIVE\_UINT64\_TYPE uint64\_t
- #define INT64(x) x##LL
- #define UINT64(x) x##ULL
- #define INT64\_HILO(high, low) (((high##LL))<<32)|low)
- #define UINT64\_HILO(high, low) (((high##ULL))<<32)|low)
- #define OSCL\_HAS\_UNICODE\_SUPPORT 1
- #define OSCL\_NATIVE\_WCHAR\_TYPE wchar\_t
- #define \_STRLIT(x) L ## x
- #define \_STRLIT\_CHAR(x) x
- #define \_STRLIT\_WCHAR(x) L ## x
- #define OSCL\_HAS\_TLS\_SUPPORT 1
- #define OSCL\_TLS\_IS\_KEYED 1
- #define OSCL\_TLS\_KEY\_CREATE\_FUNC(key) (pthread\_key\_create(&key,NULL)==0)
- #define OSCL\_TLS\_KEY\_DELETE\_FUNC(key) pthread\_key\_delete(key)
- #define OSCL\_TLS\_STORE\_FUNC(key, ptr) (pthread\_setspecific(key,(const void\*)ptr)==0)
- #define OSCL\_TLS\_GET\_FUNC(key) pthread\_getspecific(key)
- #define OSCL\_HAS\_BASIC\_LOCK 1

## Typedefs

- `typedef pthread_key_t TOsclTlsKey`
- `typedef pthread_mutex_t TOsclBasicLockObject`



### 7.155.1 Define Documentation

- 7.155.1.1 `#define _STRLIT(x) L ## x`
- 7.155.1.2 `#define _STRLIT_CHAR(x) x`
- 7.155.1.3 `#define _STRLIT_WCHAR(x) L ## x`
- 7.155.1.4 `#define INT64(x) x##LL`
- 7.155.1.5 `#define INT64_HILO(high, low) (((high##LL))<<32)|low)`
- 7.155.1.6 `#define OSCL_DISABLE_INLINES 0`
- 7.155.1.7 `#define OSCL_HAS_ANSI_MATH_SUPPORT 1`
- 7.155.1.8 `#define OSCL_HAS_ANSI_STDIO_SUPPORT 1`
- 7.155.1.9 `#define OSCL_HAS_ANSI_STDLIB_SUPPORT 1`
- 7.155.1.10 `#define OSCL_HAS_ANSI_STRING_SUPPORT 1`
- 7.155.1.11 `#define OSCL_HAS_ANSI_WIDE_STRING_SUPPORT 1`
- 7.155.1.12 `#define OSCL_HAS_BASIC_LOCK 1`
- 7.155.1.13 `#define OSCL_HAS_GLOBAL_VARIABLE_SUPPORT 1`
- 7.155.1.14 `#define OSCL_HAS_MSWIN_SUPPORT 0`
- 7.155.1.15 `#define OSCL_HAS_NATIVE_INT64_TYPE 1`
- 7.155.1.16 `#define OSCL_HAS_NATIVE_UINT64_TYPE 1`
- 7.155.1.17 `#define OSCL_HAS_SYMBIAN_SUPPORT 0`
- 7.155.1.18 `#define OSCL_HAS_TLS_SUPPORT 1`
- 7.155.1.19 `#define OSCL_HAS_UNICODE_SUPPORT 1`
- 7.155.1.20 `#define OSCL_HAS_UNIX_SUPPORT 1`
- 7.155.1.21 `#define OSCL_MEMFRAG_PTR_BEFORE_LEN 1`
- 7.155.1.22 `#define OSCL_NATIVE_INT64_TYPE int64_t`
- 7.155.1.23 `#define OSCL_NATIVE_UINT64_TYPE uint64_t`
- 7.155.1.24 `#define OSCL_NATIVE_WCHAR_TYPE wchar_t`
- 7.155.1.25 `#define OSCL_TLS_GET_FUNC(key) pthread_getspecific(key)`
- 7.155.1.26 `#define OSCL_TLS_IS_KEYED 1`
- 7.155.1.27 `#define OSCL_TLS_KEY_CREATE_FUNC(key) (pthread_key_create(&key,NULL)==0)`

## 7.156 osclconfig\_util.h File Reference

```
#include "osclconfig.h"
#include <stdio.h>
#include <time.h>
#include <sys/time.h>
#include <unistd.h>
#include "osclconfig_util_check.h"
```

### Defines

- #define OSCL\_CLOCK\_HAS\_DRIFT\_CORRECTION 0
- #define OSCL\_HAS\_SYMBIAN\_TIMERS 0
- #define OSCL\_HAS\_SYMBIAN\_MATH 0
- #define OSCL RAND\_MAX RAND\_MAX
- #define SLEEP\_ONE\_SEC sleep(1)

#### 7.156.1 Define Documentation

- 7.156.1.1 #define OSCL\_CLOCK\_HAS\_DRIFT\_CORRECTION 0
- 7.156.1.2 #define OSCL\_HAS\_SYMBIAN\_MATH 0
- 7.156.1.3 #define OSCL\_HAS\_SYMBIAN\_TIMERS 0
- 7.156.1.4 #define OSCL RAND\_MAX RAND\_MAX
- 7.156.1.5 #define SLEEP\_ONE\_SEC sleep(1)

---

**7.157 osclconfig\_util\_check.h File Reference**

## 7.158 pvlogger.h File Reference

This file contains basic logger interfaces for common use across platforms.

```
#include "oscl_base.h"  
#include "oscl_vector.h"  
#include "oscl_shared_ptr.h"  
#include "oscl_base_alloc.h"
```

### Data Structures

- class [PVLogger](#)

### Defines

- #define [PVLOGMSG\\_INST\\_REL](#) 0
- #define [PVLOGMSG\\_INST\\_PROF](#) 1
- #define [PVLOGMSG\\_INST\\_HLDBG](#) 2
- #define [PVLOGMSG\\_INST\\_MLDBG](#) 3
- #define [PVLOGMSG\\_INST\\_LLDBG](#) 4
- #define [PVLOGGER\\_INST\\_LEVEL](#) 5
- #define [\\_PVLOGGER\\_LOGMSG\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [\\_PVLOGGER\\_LOGMSG\\_V\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [\\_PVLOGGER\\_LOGBIN\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [\\_PVLOGGER\\_LOGBIN\\_V\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER\\_INST\\_LEVEL\\_SUPPORT](#) 1
- #define [PVLOGGER\\_LOGMSG\\_PVLOGMSG\\_INST\\_REL\(LOGGER, LEVEL, MESSAGE\) \\_- PVLOGGER\\_LOGMSG\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER\\_LOGMSG\\_V\\_PVLOGMSG\\_INST\\_REL\(LOGGER, LEVEL, MESSAGE\) \\_- PVLOGGER\\_LOGMSG\\_V\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER\\_LOGBIN\\_PVLOGMSG\\_INST\\_REL\(LOGGER, LEVEL, MESSAGE\) \\_- PVLOGGER\\_LOGBIN\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER\\_LOGBIN\\_V\\_PVLOGMSG\\_INST\\_REL\(LOGGER, LEVEL, MESSAGE\) \\_- PVLOGGER\\_LOGBIN\\_V\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER\\_LOGMSG\\_PVLOGMSG\\_INST\\_PROF\(LOGGER, LEVEL, MESSAGE\) \\_- PVLOGGER\\_LOGMSG\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER\\_LOGMSG\\_V\\_PVLOGMSG\\_INST\\_PROF\(LOGGER, LEVEL, MESSAGE\) \\_- PVLOGGER\\_LOGMSG\\_V\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER\\_LOGBIN\\_PVLOGMSG\\_INST\\_PROF\(LOGGER, LEVEL, MESSAGE\) \\_- PVLOGGER\\_LOGBIN\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER\\_LOGBIN\\_V\\_PVLOGMSG\\_INST\\_PROF\(LOGGER, LEVEL, MESSAGE\) \\_- PVLOGGER\\_LOGBIN\\_V\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER\\_LOGMSG\\_PVLOGMSG\\_INST\\_HLDBG\(LOGGER, LEVEL, MESSAGE\) \\_- PVLOGGER\\_LOGMSG\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER\\_LOGMSG\\_V\\_PVLOGMSG\\_INST\\_HLDBG\(LOGGER, LEVEL, MESSAGE\) \\_- PVLOGGER\\_LOGMSG\\_V\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER\\_LOGBIN\\_PVLOGMSG\\_INST\\_HLDBG\(LOGGER, LEVEL, MESSAGE\) \\_- PVLOGGER\\_LOGBIN\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER\\_LOGBIN\\_V\\_PVLOGMSG\\_INST\\_HLDBG\(LOGGER, LEVEL, MESSAGE\) \\_- PVLOGGER\\_LOGBIN\\_V\(LOGGER, LEVEL, MESSAGE\)](#)

- #define **PVLOGGER\_LOGMSG\_PVLOGMSG\_INST\_MLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGMSG(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_INST\_MLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGMSG\_V(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_MLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_V\_INST\_MLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGMSG\_PVLOGMSG\_INST\_LLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGMSG(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_INST\_LLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGMSG\_V(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_LLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_LLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGMSG**(IL, LOGGER, LEVEL, MESSAGE) PVLOGGER\_LOGMSG\_## IL (LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGMSG\_V**(IL, LOGGER, LEVEL, MESSAGE) PVLOGGER\_LOGMSG\_V\_## IL (LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGBIN**(IL, LOGGER, LEVEL, MESSAGE) PVLOGGER\_LOGBIN\_## IL (LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGBIN\_V**(IL, LOGGER, LEVEL, MESSAGE) PVLOGGER\_LOGBIN\_V\_## IL (LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOG\_USE\_ONLY**(x) x
- #define **PVLOGGER\_ENABLE** 1

## Variables

- const int32 **PVLOGGER\_LEVEL\_UNINITIALIZED** = -1
- const **PVLogger::log\_level\_type** **PVLOGMSG\_EMERG** = 0
- const **PVLogger::log\_level\_type** **PVLOGMSG\_ALERT** = 1
- const **PVLogger::log\_level\_type** **PVLOGMSG\_CRIT** = 2
- const **PVLogger::log\_level\_type** **PVLOGMSG\_ERR** = 3
- const **PVLogger::log\_level\_type** **PVLOGMSG\_WARNING** = 4
- const **PVLogger::log\_level\_type** **PVLOGMSG\_NOTICE** = 5
- const **PVLogger::log\_level\_type** **PVLOGMSG\_INFO** = 6
- const **PVLogger::log\_level\_type** **PVLOGMSG\_STACK\_TRACE** = 7
- const **PVLogger::log\_level\_type** **PVLOGMSG\_DEBUG** = 8
- const **PVLogger::log\_level\_type** **PVLOGMSG\_FATAL\_ERROR** = **PVLOGMSG\_EMERG**
- const **PVLogger::log\_level\_type** **PVLOGMSG\_NONFATAL\_ERROR** = **PVLOGMSG\_ERR**
- const **PVLogger::log\_level\_type** **PVLOGMSG\_STATISTIC** = **PVLOGMSG\_INFO**
- const **PVLogger::log\_level\_type** **PVLOGMSG\_VERBOSE** = **PVLOGMSG\_DEBUG**

### 7.158.1 Detailed Description

This file contains basic logger interfaces for common use across platforms.

This is the main entry point header file for the logger library. It should be the only one users directly include.

## 7.158.2 Define Documentation

### 7.158.2.1 #define \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)

**Value:**

```
{\
  if (LOGGER) \
  {\
    if (LOGGER->IsActive(LEVEL)) \
    {\
      LOGGER->LogMsgBuffers MESSAGE; \
    }\
  }\
}
```

### 7.158.2.2 #define \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)

**Value:**

```
{\
  if (LOGGER) \
  {\
    if (LOGGER->IsActive(LEVEL)) \
    {\
      LOGGER->LogMsgBuffersV MESSAGE; \
    }\
  }\
}
```

### 7.158.2.3 #define \_PVLOGGER\_LOGMSG(LOGGER, LEVEL, MESSAGE)

**Value:**

```
{\
  if (LOGGER) \
  {\
    if (LOGGER->IsActive(LEVEL)) \
    {\
      LOGGER->LogMsgString MESSAGE; \
    }\
  }\
}
```

### 7.158.2.4 #define \_PVLOGGER\_LOGMSG\_V(LOGGER, LEVEL, MESSAGE)

**Value:**

```
{\
  if (LOGGER) \
  {\
    if (LOGGER->IsActive(LEVEL)) \
    {\
      LOGGER->LogMsgStringV MESSAGE; \
    }\
  }\
}
```

**7.158.2.5 #define PVLOGGER\_ENABLE 1**

In case logging is compiled out, there is no need to compile the logger runtime code either.

**7.158.2.6 #define PVLOGGER\_INST\_LEVEL 5****7.158.2.7 #define PVLOGGER\_INST\_LEVEL\_SUPPORT 1****7.158.2.8 #define PVLOGGER\_LOG\_USE\_ONLY(x) x**

Used to compile in/out lines of code that are used only for **PVLogger** macros.

This code will be removed at compile time when **PVLogger** is disabled, i.e. Release mode. So do not put in any code that is necessary for correct functionality of the module

**7.158.2.9 #define PVLOGGER\_LOGBIN(IL, LOGGER, LEVEL, MESSAGE)  
PVLOGGER\_LOGBIN\_ ## IL (LOGGER, LEVEL, MESSAGE)**

This is a binary API to log messages

**Parameters:**

**IL** Instrumentation level.

**LOGGER** Pointer to the logger object, that acts as the logging control/interface point

**LEVEL** Log level of the message

**MESSAGE** Log Message which includes the message id, and message buffers that need to be logged.

Example Usage: PVLOGGER\_LOGBIN (PVLOGMSG\_INST\_LLDBG, logger\_1, PVLOGMSG\_WARNING, (10, 3, msgBuf1Size, msgBuf1, msgBuf2Size, msgBuf2, msgBuf3Size, msgBuf3));

-This message contains THREE (ptr\_len, ptr) pairs. Log level of this msg is PVLOGMSG\_WARNING, message id is 10.

- 7.158.2.10 #define PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_HLDBG(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)
- 7.158.2.11 #define PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_LLDBG(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)
- 7.158.2.12 #define PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_MLDBG(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)
- 7.158.2.13 #define PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_PROF(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)
- 7.158.2.14 #define PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_REL(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)
- 7.158.2.15 #define PVLOGGER\_LOGBIN\_V(IL, LOGGER, LEVEL, MESSAGE)  
PVLOGGER\_LOGBIN\_V\_## IL (LOGGER, LEVEL, MESSAGE)
- 7.158.2.16 #define PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_HLDBG(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)
- 7.158.2.17 #define PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_LLDBG(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)
- 7.158.2.18 #define PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_PROF(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)
- 7.158.2.19 #define PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_REL(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)
- 7.158.2.20 #define PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_V\_INST\_MLDBG(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)
- 7.158.2.21 #define PVLOGGER\_LOGMSG(IL, LOGGER, LEVEL, MESSAGE)  
PVLOGGER\_LOGMSG\_## IL (LOGGER, LEVEL, MESSAGE)

This is the text based API to log messages

**Parameters:**

**IL** Instrumentation level.

**LOGGER** Pointer to the logger object, that acts as the logging control/interface point

**LEVEL** Log level of the message

**MESSAGE** Log Message which includes the message id, and any kind of formatting information

Example Usage: PVLOGGER\_LOGMSG(PVLOGMSG\_INST\_LLDBG, logger\_1, PVLOGMSG\_WARNING, (13, "Test Messsage to Node 1

")); -This message of log level PVLOGMSG\_WARNING, and has a message id of 13

- 7.158.2.22 `#define PVLOGGER_LOGMSG_PVLOGMSG_INST_HLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)`
- 7.158.2.23 `#define PVLOGGER_LOGMSG_PVLOGMSG_INST_LLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)`
- 7.158.2.24 `#define PVLOGGER_LOGMSG_PVLOGMSG_INST_MLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)`
- 7.158.2.25 `#define PVLOGGER_LOGMSG_PVLOGMSG_INST_PROF(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)`
- 7.158.2.26 `#define PVLOGGER_LOGMSG_PVLOGMSG_INST_REL(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)`
- 7.158.2.27 `#define PVLOGGER_LOGMSG_V(IL, LOGGER, LEVEL, MESSAGE)  
PVLOGGER_LOGMSG_V_## IL (LOGGER, LEVEL, MESSAGE)`
- 7.158.2.28 `#define PVLOGGER_LOGMSG_V_PVLOGMSG_INST_HLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)`
- 7.158.2.29 `#define PVLOGGER_LOGMSG_V_PVLOGMSG_INST_LLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)`
- 7.158.2.30 `#define PVLOGGER_LOGMSG_V_PVLOGMSG_INST_MLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)`
- 7.158.2.31 `#define PVLOGGER_LOGMSG_V_PVLOGMSG_INST_PROF(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)`
- 7.158.2.32 `#define PVLOGGER_LOGMSG_V_PVLOGMSG_INST_REL(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)`
- 7.158.2.33 `#define PVLOGMSG_INST_HLDBG 2`

#### High Level Debug Layer

This layer should contain messages that have very minimal impact on performance, but are at lower level (i.e., provide more information) than would be appropriate in a shipping product. The messages are probably used to gather information and validate proper functionality at a high level as might be appropriate for IOT, stress testing, or QA testing.

#### 7.158.2.34 `#define PVLOGMSG_INST_LLDBG 4`

#### Low Level Debug Layer

This layer should contain messages for early functional testing. The messages are typically at a very low-level and allow testing the functionality of individual modules and components. Messages at this layer will typically have a performance impact (sometimes significant) due to the fact that they are at such a low level.

### 7.158.2.35 #define PVLOGMSG\_INST\_MLDBG 3

Mid Level Debug Layer

This layer should contain messages that are useful in the middle stages of the development cycle where major components are being integrated. The components themselves should already be well-tested so the emphasis is on interfaces between these components and integration testing. Messages at this layer may have some performance impact.

### 7.158.2.36 #define PVLOGMSG\_INST\_PROF 1

Profile Layer

The profile layer is used for messages and information related to measuring and reporting performance-related information.

### 7.158.2.37 #define PVLOGMSG\_INST\_REL 0

Release Layer

The release layer should only be used for messages that should remain in the final release. In certain cases all messaging may be disabled depending on customer requirements. However, when allowed the release layer should contain information that will be useful diagnosing problems in a released product (perhaps after entering a diagnostic mode), but with absolutely minimal performance impact when disabled at runtime.

## 7.158.3 Variable Documentation

### 7.158.3.1 const int32 PVLOGGER\_LEVEL\_UNINITIALIZED = -1

### 7.158.3.2 const PVLogger::log\_level\_type PVLOGMSG\_ALERT = 1

action must be taken immediately

### 7.158.3.3 const PVLogger::log\_level\_type PVLOGMSG\_CRIT = 2

critical conditions

### 7.158.3.4 const PVLogger::log\_level\_type PVLOGMSG\_DEBUG = 8

debug-level messages

### 7.158.3.5 const PVLogger::log\_level\_type PVLOGMSG\_EMERG = 0

system is unusable

### 7.158.3.6 const PVLogger::log\_level\_type PVLOGMSG\_ERR = 3

error conditions

7.158.3.7 const **PVLogger::log\_level\_type** PVLOGMSG\_FATAL\_ERROR =  
**PVLOGMSG\_EMERG**

7.158.3.8 const **PVLogger::log\_level\_type** PVLOGMSG\_INFO = 6

informational

7.158.3.9 const **PVLogger::log\_level\_type** PVLOGMSG\_NONFATAL\_ERROR =  
**PVLOGMSG\_ERR**

7.158.3.10 const **PVLogger::log\_level\_type** PVLOGMSG\_NOTICE = 5

normal but significant condition

7.158.3.11 const **PVLogger::log\_level\_type** PVLOGMSG\_STACK\_TRACE = 7

function enter and exit

7.158.3.12 const **PVLogger::log\_level\_type** PVLOGMSG\_STATISTIC = **PVLOGMSG\_INFO**

7.158.3.13 const **PVLogger::log\_level\_type** PVLOGMSG\_VERBOSE = **PVLOGMSG\_DEBUG**

7.158.3.14 const **PVLogger::log\_level\_type** PVLOGMSG\_WARNING = 4

warning conditions

## 7.159 pvlogger\_accessories.h File Reference

```
#include "oscl_base.h"
#include "pvlogger.h"
```

### Data Structures

- class [AllPassFilter](#)
- class [PVLoggerAppender](#)
- class [PVLoggerFilter](#)
- class [PVLoggerLayout](#)

### Variables

- const [PVLoggerFilter::filter\\_status\\_type PVLOGGER\\_FILTER\\_ACCEPT = 1](#)
- const [PVLoggerFilter::filter\\_status\\_type PVLOGGER\\_FILTER\\_REJECT = 2](#)
- const [PVLoggerFilter::filter\\_status\\_type PVLOGGER\\_FILTER\\_NEUTRAL = 3](#)

#### 7.159.1 Variable Documentation

7.159.1.1 const [PVLoggerFilter::filter\\_status\\_type PVLOGGER\\_FILTER\\_ACCEPT = 1](#)

7.159.1.2 const [PVLoggerFilter::filter\\_status\\_type PVLOGGER\\_FILTER\\_NEUTRAL = 3](#)

7.159.1.3 const [PVLoggerFilter::filter\\_status\\_type PVLOGGER\\_FILTER\\_REJECT = 2](#)

## 7.160 pvlogger\_c.h File Reference

This file contains basic logger interfaces for common use across platforms. C-callable version.

```
#include "osclconfig.h"
```

### Defines

- #define PVLOGGER\_C\_INST\_LEVEL 5
- #define PVLOGMSG\_C\_INST\_REL 0
- #define PVLOGMSG\_C\_INST\_PROF 1
- #define PVLOGMSG\_C\_INST\_HLDBG 2
- #define PVLOGMSG\_C\_INST\_MLDBG 3
- #define PVLOGMSG\_C\_INST\_LLDBG 4
- #define PVLOGMSG\_C\_EMERG 0
- #define PVLOGMSG\_C\_ALERT 1
- #define PVLOGMSG\_C\_CRIT 2
- #define PVLOGMSG\_C\_ERR 3
- #define PVLOGMSG\_C\_WARNING 4
- #define PVLOGMSG\_C\_NOTICE 5
- #define PVLOGMSG\_C\_INFO 6
- #define PVLOGMSG\_C\_STACK\_TRACE 7
- #define PVLOGMSG\_C\_STACK\_DEBUG 8

### Functions

- OSCL\_IMPORT\_REF void \* [pvLogger\\_GetLoggerObject](#) (const char \*tag)
- OSCL\_IMPORT\_REF int [pvLogger\\_IsActive](#) (void \*logger, int log\_level)
- OSCL\_IMPORT\_REF void [pvLogger\\_LogMsgString](#) (void \*logger, int msgID, const char \*fmt,...)

### 7.160.1 Detailed Description

This file contains basic logger interfaces for common use across platforms. C-callable version.

This is the main entry point header file for the logger library. It should be the only one users directly include.

## 7.160.2 Define Documentation

- 7.160.2.1 `#define PVLOGGER_C_INST_LEVEL 5`
- 7.160.2.2 `#define PVLOGMSG_C_ALERT 1`
- 7.160.2.3 `#define PVLOGMSG_C_CRIT 2`
- 7.160.2.4 `#define PVLOGMSG_C_EMERG 0`
- 7.160.2.5 `#define PVLOGMSG_C_ERR 3`
- 7.160.2.6 `#define PVLOGMSG_C_INFO 6`
- 7.160.2.7 `#define PVLOGMSG_C_INST_HLDBG 2`
- 7.160.2.8 `#define PVLOGMSG_C_INST_LLDBG 4`
- 7.160.2.9 `#define PVLOGMSG_C_INST_MLDBG 3`
- 7.160.2.10 `#define PVLOGMSG_C_INST_PROF 1`
- 7.160.2.11 `#define PVLOGMSG_C_INST_REL 0`
- 7.160.2.12 `#define PVLOGMSG_C_NOTICE 5`
- 7.160.2.13 `#define PVLOGMSG_C_STACK_DEBUG 8`
- 7.160.2.14 `#define PVLOGMSG_C_STACK_TRACE 7`
- 7.160.2.15 `#define PVLOGMSG_C_WARNING 4`

## 7.160.3 Function Documentation

- 7.160.3.1 `OSCL_IMPORT_REF void* pvLogger_GetLoggerObject (const char * tag)`
- 7.160.3.2 `OSCL_IMPORT_REF int pvLogger_IsActive (void * logger, int log_level)`
- 7.160.3.3 `OSCL_IMPORT_REF void pvLogger_LogMsgString (void * logger, int msgID, const char * fmt, ...)`

## 7.161 pvlogger\_registry.h File Reference

```
#include "pvlogger.h"
#include "oscl_tagtree.h"
```

### Data Structures

- class [PVLoggerRegistry](#)

# Index

~AllPassFilter  
    AllPassFilter, 111  
~BufFragGroup  
    BufFragGroup, 117  
~BufferMgr  
    BufferMgr, 114  
~CallbackTimer  
    CallbackTimer, 120  
~CallbackTimerObserver  
    CallbackTimerObserver, 122  
~DNSRequestParam  
    DNSRequestParam, 129  
~GetHostByNameParam  
    GetHostByNameParam, 131  
~HeapBase  
    HeapBase, 133  
~MM\_AllocInfo  
    MM\_AllocInfo, 145  
~MM\_AllocNode  
    MM\_AllocNode, 146  
~MM\_Audit\_Imp  
    MM\_Audit\_Imp, 149  
~MediaData  
    MediaData, 138  
~MemAllocator  
    MemAllocator, 141  
~OSCLMemAutoPtr  
    OSCLMemAutoPtr, 421  
~OSCL\_FastString  
    OSCL\_FastString, 171  
~OSCL\_HeapString  
    osclutil, 80  
~OSCL\_HeapStringA  
    OSCL\_HeapStringA, 195  
~OSCL\_StackString  
    osclutil, 80  
~OSCL\_String  
    OSCL\_String, 252  
~OSCL\_wFastString  
    OSCL\_wFastString, 286  
~OSCL\_wHeapString  
    osclutil, 80  
~OSCL\_wHeapStringA  
    OSCL\_wHeapStringA, 291  
~OSCL\_wStackString  
    osclutil, 80  
~OSCL\_wString  
    OSCL\_wString, 296  
~OsclAcceptMethod  
    OsclAcceptMethod, 299  
~OsclActiveObject  
    OsclActiveObject, 302  
~OsclAllocDestructDealloc  
    OsclAllocDestructDealloc, 305  
~OsclAsyncFile  
    OsclAsyncFile, 308  
~OsclAsyncFileBuffer  
    OsclAsyncFileBuffer, 311  
~OsclBinIStream  
    OsclBinIStream, 315  
~OsclBinOStream  
    OsclBinOStream, 322  
~OsclBindMethod  
    OsclBindMethod, 313  
~OsclComponentRegistry  
    OsclComponentRegistry, 335  
~OsclComponentRegistryElement  
    OsclComponentRegistryElement, 337  
~OsclConnectMethod  
    OsclConnectMethod, 339  
~OsclDNS  
    OsclDNS, 342  
~OsclDNSI  
    OsclDNSI, 344  
~OsclDNSIBase  
    OsclDNSIBase, 347  
~OsclDNSObserver  
    OsclDNSObserver, 352  
~OsclDNSRequest  
    OsclDNSRequest, 353  
~OsclExclusiveArrayPtr  
    OsclExclusiveArrayPtr, 372  
~OsclExclusivePtr  
    OsclExclusivePtr, 375  
~OsclExclusivePtrA  
    OsclExclusivePtrA, 378  
~OsclExecSchedulerCommonBase  
    OsclExecSchedulerCommonBase, 386  
~OsclFileCache  
    OsclFileCache, 393

---

~OsclGetHostByNameMethod  
     OsclGetHostByNameMethod, 399  
 ~OsclIPSocketI  
     OsclIPSocketI, 404  
 ~OsclJump  
     OsclJump, 406  
 ~OsclListenMethod  
     OsclListenMethod, 407  
 ~OsclLockBase  
     OsclLockBase, 409  
 ~OsclMemAudit  
     OsclMemAudit, 414  
 ~OsclMemPoolAllocator  
     OsclMemPoolAllocator, 428  
 ~OsclMemPoolFixedChunkAllocator  
     OsclMemPoolFixedChunkAllocator, 430  
 ~OsclMemPoolFixedChunkAllocatorObserver  
     OsclMemPoolFixedChunkAllocator-  
         Observer, 433  
 ~OsclMemPoolResizableAllocator  
     OsclMemPoolResizableAllocator, 435  
 ~OsclMemPoolResizableAllocatorMemoryObserver  
     OsclMemPoolResizableAllocatorMemory-  
         Observer, 442  
 ~OsclMemPoolResizableAllocatorObserver  
     OsclMemPoolResizableAllocatorObserver,  
         443  
 ~OsclMemStatsNode  
     OsclMemStatsNode, 444  
 ~OsclMutex  
     OsclMutex, 445  
 ~OsclNativeFile  
     OsclNativeFile, 449  
 ~OsclNullLock  
     OsclNullLock, 453  
 ~OsclPriorityQueue  
     OsclPriorityQueue, 457  
 ~OsclPriorityQueueBase  
     OsclPriorityQueueBase, 460  
 ~OsclRecvFromMethod  
     OsclRecvFromMethod, 472  
 ~OsclRecvMethod  
     OsclRecvMethod, 476  
 ~OsclRefCounter  
     OsclRefCounter, 478  
 ~OsclRefCounterDA  
     OsclRefCounterDA, 480  
 ~OsclRefCounterMTDA  
     OsclRefCounterMTDA, 484  
 ~OsclRefCounterMTSA  
     OsclRefCounterMTSA, 486  
 ~OsclRefCounterMemFrag  
     OsclRefCounterMemFrag, 482  
 ~OsclRefCounterSA  
     OsclRefCounterSA, 488  
 ~OsclRegistryAccessClient  
     OsclRegistryAccessClient, 490  
 ~OsclRegistryClient  
     OsclRegistryClient, 495  
 ~OsclRegistryServTlsImpl  
     OsclRegistryServTlsImpl, 501  
 ~OsclSchedulerObserver  
     OsclSchedulerObserver, 503  
 ~OsclScopedLock  
     OsclScopedLock, 504  
 ~OsclSemaphore  
     OsclSemaphore, 507  
 ~OsclSendMethod  
     OsclSendMethod, 509  
 ~OsclSendToMethod  
     OsclSendToMethod, 511  
 ~OsclSharedPtr  
     OsclSharedPtr, 514  
 ~OsclShutdownMethod  
     OsclShutdownMethod, 516  
 ~OsclSingleton  
     OsclSingleton, 518  
 ~OsclSocketI  
     OsclSocketI, 522  
 ~OsclSocketIBase  
     OsclSocketIBase, 527  
 ~OsclSocketMethod  
     OsclSocketMethod, 532  
 ~OsclSocketObserver  
     OsclSocketObserver, 534  
 ~OsclSocketRequestAO  
     OsclSocketRequestAO, 537  
 ~OsclSocketServ  
     OsclSocketServ, 540  
 ~OsclSocketServIBase  
     OsclSocketServIBase, 545  
 ~OsclTCPSocket  
     OsclTCPSocket, 550  
 ~OsclTCPSocketI  
     OsclTCPSocketI, 556  
 ~OsclTLS  
     OsclTLS, 575  
 ~OsclTLSEx  
     OsclTLSEx, 577  
 ~OsclThread  
     OsclThread, 558  
 ~OsclThreadLock  
     OsclThreadLock, 562  
 ~OsclTimer  
     OsclTimer, 566  
 ~OsclTimerObject  
     OsclTimerObject, 570  
 ~OsclTimerObserver

OsclTimerObserver, 573  
 ~OsclUDPSocket  
   OsclUDPSocket, 584  
 ~OsclUDPSocketI  
   OsclUDPSocketI, 590  
 ~Oscl\_File  
   Oscl\_File, 176  
 ~Oscl\_FileFind  
   Oscl\_FileFind, 185  
 ~Oscl\_FileServer  
   Oscl\_FileServer, 188  
 ~Oscl\_Linked\_List  
   Oscl\_Linked\_List, 200  
 ~Oscl\_Linked\_List\_Base  
   Oscl\_Linked\_List\_Base, 205  
 ~Oscl\_MTLinked\_List  
   Oscl\_MTLinked\_List, 217  
 ~Oscl\_Queue  
   Oscl\_Queue, 228  
 ~Oscl\_Queue\_Base  
   Oscl\_Queue\_Base, 230  
 ~Oscl\_Rb\_Tree  
   Oscl\_Rb\_Tree, 235  
 ~Oscl\_TAlloc  
   Oscl\_TAlloc, 273  
 ~Oscl\_Tag  
   Oscl\_Tag, 256  
 ~Oscl\_TagTree  
   Oscl\_TagTree, 261  
 ~Oscl\_Vector  
   Oscl\_Vector, 277  
 ~Oscl\_Vector\_Base  
   Oscl\_Vector\_Base, 282  
 ~PVActiveBase  
   PVActiveBase, 594  
 ~PVLogger  
   PVLogger, 599  
 ~PVLoggerAppender  
   PVLoggerAppender, 604  
 ~PVLoggerFilter  
   PVLoggerFilter, 606  
 ~PVLoggerLayout  
   PVLoggerLayout, 607  
 ~PVLoggerRegistry  
   PVLoggerRegistry, 609  
 ~PVSchedulerStopper  
   PVSchedulerStopper, 612  
 ~PVThreadContext  
   PVThreadContext, 615  
 ~SendToParam  
   SendToParam, 621  
 ~OsclBasicAllocator  
   \_OsclBasicAllocator, 105  
 ~\_OsclHeapBase  
   \_OsclHeapBase, 107  
   (FILE\_OFFSET\_BITS  
     osclconfig\_io.h, 793  
   \_OSCL\_Abort  
     osclbase, 33  
   \_OSCL\_CLEANUP\_BASE\_CLASS  
     osclmemory, 47  
   \_OSCL\_TRAP\_NEW  
     osclmemory, 47  
   \_OsclBasicAllocator, 104  
   \_OsclBasicAllocator  
     ~\_OsclBasicAllocator, 105  
     allocate, 105  
     deallocate, 105  
   \_OsclHeapBase, 106  
     \_OsclHeapBase, 107  
   \_OsclHeapBase  
     ~\_OsclHeapBase, 107  
     \_OsclHeapBase, 107  
     PVCleanupStack, 107  
 \_OsclInteger64Transport  
   oscl\_int64\_utils.h, 684  
 \_Ownership  
   OSCLMemAutoPtr, 423  
 \_PVLOGGER\_LOGBIN  
   pvlogger.h, 827  
 \_PVLOGGER\_LOGBIN\_V  
   pvlogger.h, 827  
 \_PVLOGGER\_LOGMSG  
   pvlogger.h, 827  
 \_PVLOGGER\_LOGMSG\_V  
   pvlogger.h, 827  
 \_PV\_TRAP  
   oscl\_error\_imp\_fatalerror.h, 665  
   oscl\_error\_imp\_jumps.h, 666  
   osclerror, 84  
 \_PV\_TRAP\_NO\_TLS  
   oscl\_error\_imp\_fatalerror.h, 665  
   oscl\_error\_imp\_jumps.h, 666  
   osclerror, 84  
 \_Ptr  
   OsclExclusiveArrayPtr, 373  
   OsclExclusivePtr, 376  
   OsclExclusivePtrA, 379  
   OsclSingleton, 519  
   OsclTLS, 576  
   OsclTLSEx, 578  
 \_STRLIT  
   osclconfig\_unix\_android.h, 818  
   osclconfig\_unix\_common.h, 822  
 \_STRLIT\_CHAR  
   osclconfig\_unix\_android.h, 818  
   osclconfig\_unix\_common.h, 822  
 \_STRLIT\_WCHAR

---

```

osclconfig_unix_android.h, 818
osclconfig_unix_common.h, 822
__TFS__
  osclconfig.h, 783
__Validate_BasicTimeDateStruct__
  osclconfig_time_check.h, 814
__Validate_BasicTimeStruct__
  osclconfig_time_check.h, 814
__int16_check__
  osclconfig, 22
__int32_check__
  osclconfig, 22
__int8_check__
  osclconfig, 22
__uint16_check__
  osclconfig, 22
__uint32_check__
  osclconfig, 22
__uint8_check__
  osclconfig, 22
__verify_TOsclConditionObject_defined__
  osclconfig_proc_check.h, 807
__verify_TOsclFileOffset_defined__
  osclconfig_io_check.h, 798
__verify_TOsclMutexObject_defined__
  osclconfig_proc_check.h, 807
__verify_TOsclSemaphoreObject_defined__
  osclconfig_proc_check.h, 807
__verify_TOsclThreadFuncArg_defined__
  osclconfig_proc_check.h, 807
__verify_TOsclThreadFuncRet_defined__
  osclconfig_proc_check.h, 807
__verify_TOsclThreadId_defined__
  osclconfig_proc_check.h, 807
__verify_TOsclThreadObject_defined__
  osclconfig_proc_check.h, 807
_fixedCaches
  OsclFileCache, 393
_movableCache
  OsclFileCache, 393
_oscl_audit_calloc
  osclmemory, 56
_oscl_audit_free
  osclmemory, 56
_oscl_audit_malloc
  osclmemory, 56
_oscl_audit_new
  osclmemory, 56
_oscl_audit_realloc
  osclmemory, 57
_oscl_malloc
  osclmemory, 57
_oscl_default_audit_malloc
  osclmemory, 57
_oscl_default_audit_new
  osclmemory, 57
_oscl_default_audit_realloc
  osclmemory, 57
_oscl_free
  osclmemory, 57
_oscl_malloc
  osclmemory, 57
_oscl_realloc
  osclmemory, 57

a
  internalLeave, 134
Abort
  OsclDNSMethod, 350
  OsclDNSRequestAO, 355
  OsclSocketMethod, 532
  OsclSocketRequestAO, 537
AbortAll
  OsclDNSMethod, 350
  OsclSocketMethod, 532
Accept
  OsclAcceptMethod, 299
  OsclAcceptRequest, 300
  OsclSocketI, 522
  OsclSocketIBase, 527
  OsclTCPSocket, 550
  OsclTCPSocketI, 556
AcceptParam, 108
  AcceptParam, 108
AcceptParam
  AcceptParam, 108
  iBlankSocket, 108
AcceptRequest
  OsclAcceptMethod, 299
Activate
  OsclDNSRequest, 353
  OsclSocketRequest, 535
  PVActiveBase, 594
Add
  OsclSocketServRequestList, 546
  OsclTimerQ, 574
add_element
  Oscl_Linked_List, 201
  Oscl_Linked_List_Base, 205
  Oscl_MTLLinked_List, 218
add_ref
  CHeapRep, 126
add_to_front
  Oscl_Linked_List, 201
  Oscl_Linked_List_Base, 205
  Oscl_MTLLinked_List, 218

```

addAllocNode  
   MM\_Audit\_Imp, 149  
 AddAppender  
   PVLogger, 599  
 AddFilter  
   PVLogger, 599  
 AddFixedCache  
   Oscl\_File, 176  
   OsclFileCache, 393  
 AddFragment  
   BufFragGroup, 117  
 AddLocalFragment  
   MediaData, 138  
 addnewmempoolbuffer  
   OsclMemPoolResizableAllocator, 435  
 addRef  
   Oscl\_DefAllocWithRefCounter, 168  
   OsclMemPoolFixedChunkAllocator, 430  
   OsclMemPoolResizableAllocator, 435  
   OsclRefCounter, 478  
   OsclRefCounterDA, 481  
   OsclRefCounterMTDA, 485  
   OsclRefCounterMTSA, 487  
   OsclRefCounterSA, 489  
 address  
   Oscl\_TAlloc, 273  
 AddToExecTimerQ  
   OsclExecSchedulerCommonBase, 386  
 AddToScheduler  
   OsclActiveObject, 302  
   OsclTimerObject, 570  
   PVActiveBase, 594  
 After  
   OsclTimerObject, 570  
 Alloc  
   OsclIPSocketI, 404  
   OsclSocketMethod, 532  
   OsclSocketRequestAO, 537  
 ALLOC\_AND\_CONSTRUCT  
   osclbase, 30  
 alloc\_and\_construct  
   Oscl\_TAlloc, 273  
 alloc\_and\_construct\_fl  
   Oscl\_TAlloc, 273  
 ALLOC\_NODE\_FLAG  
   osclmemory, 59  
 alloc\_type  
   PVLogger, 599  
   PVLoggerRegistry, 609  
 ALLOCATE  
   osclbase, 30  
 allocate  
   \_OsclBasicAllocator, 105  
   MemAllocator, 141  
   Oscl\_Alloc, 165  
   Oscl\_DefAlloc, 167  
   Oscl\_Opaque\_Type\_Alloc, 221  
   Oscl\_Opaque\_Type\_Alloc\_LL, 222  
   Oscl\_TAlloc, 273  
   OsclErrorAllocator, 365  
   OsclMemAllocator, 411  
   OsclMemAllocDestructDealloc, 412  
   OSCLMemAutoPtr, 422  
   OsclMemBasicAllocator, 424  
   OsclMemBasicAllocDestructDealoc, 425  
   OsclMemPoolFixedChunkAllocator, 430  
   OsclMemPoolResizableAllocator, 436  
   OsclReadyAlloc, 468  
 allocate\_fl  
   Oscl\_Alloc, 165  
   Oscl\_DefAlloc, 167  
   Oscl\_TAlloc, 273  
   OsclMemAllocator, 411  
   OsclMemAllocDestructDealloc, 412  
   OsclReadyAlloc, 468  
 allocateblock  
   OsclMemPoolResizableAllocator, 436  
 allocator, 109  
 allocNum  
   MM\_AllocInfo, 145  
   MM\_AllocQueryInfo, 147  
 AllPassFilter, 110  
   AllPassFilter, 111  
 AllPassFilter  
   ~AllPassFilter, 111  
   AllPassFilter, 111  
   filter\_status\_type, 110  
   FilterOpaqueMessge, 111  
   FilterString, 111  
   log\_level\_type, 110  
   message\_id\_type, 110  
 ALREADY\_SUSPENDED\_ERROR  
   OsclProcStatus, 461  
 Append  
   OsclPtr, 463  
 append  
   CFastRep, 124  
   CHeapRep, 126  
   CStackRep, 128  
 APPEND\_MEDIA\_AT\_END  
   osclutil, 80  
 append\_rep  
   CHeapRep, 126  
   OSCL\_String, 252  
   OSCL\_wString, 296  
 AppendBuffers  
   PVLoggerAppender, 604  
 AppendNext

BufFragGroup, 117  
 AppendString  
     PVLoggerAppender, 604  
 assign  
     CHearRep, 126  
 assign\_vector  
     Oscl\_Vector\_Base, 282  
 asyncfilereadcancel\_test  
     Oscl\_File, 181  
 asyncfilereadwrite\_test  
     Oscl\_File, 181  
 Attach  
     OsclBinStream, 328  
 audit\_type  
     OsclMemGlobalAuditObject, 426  
 available\_localbuf  
     MediaData, 139  
  
 back  
     Oscl\_Queue, 228  
     Oscl\_Vector, 278  
 BAD\_THREADID\_ADDR\_ERROR  
     OsclProcStatus, 461  
 base\_link\_type  
     Oscl\_Rb\_Tree\_Base, 237  
     Oscl\_Rb\_Tree\_Const\_Iterator, 239  
     Oscl\_Rb\_Tree\_Iterator, 242  
     Oscl\_Rb\_Tree\_Node\_Base, 245  
 begin  
     Oscl\_Map, 211  
     Oscl\_Rb\_Tree, 235  
     Oscl\_TagTree, 261  
     Oscl\_Vector, 278  
 BeginScheduling  
     OsclExecSchedulerCommonBase, 386  
 BeginStats  
     OsclExecSchedulerCommonBase, 386  
 BFG\_SUCCESS  
     BufFragStatusClass, 119  
 big\_endian\_to\_host  
     osclbase, 33  
 Bind  
     osclbase, 33  
     OsclBindMethod, 313  
     OsclBindRequest, 314  
     OsclIPSocketI, 404  
     OsclSocketI, 522  
     OsclSocketIBase, 527  
     OsclTCPSocket, 550  
     OsclUDPSocket, 585  
 bind  
     BufferState, 115  
 BindAsync  
     OsclSocketIBase, 527  
  
 OsclTCPSocket, 550  
 OsclTCPSocketI, 556  
 OsclUDPSocket, 585  
 OsclUDPSocketI, 590  
 BindParam, 112  
     BindParam, 112  
 BindParam  
     BindParam, 112  
     iAddr, 112  
 BindRequest  
     OsclBindMethod, 313  
 black  
     Oscl\_Rb\_Tree\_Node\_Base, 245  
 BlockingLoopL  
     OsclExecSchedulerCommonBase, 386  
 bSetFailure  
     MM\_AllocInfo, 145  
 Buffer  
     OsclAsyncFileBuffer, 311  
 buffer  
     CFastRep, 124  
     CHearRep, 126  
     CStackRep, 128  
 buffer\_states  
     BufFragGroup, 118  
 BufferFragment, 113  
 BufferFreeFuncPtr  
     osclutil, 66  
 BufferMgr, 114  
 BufferMgr  
     ~BufferMgr, 114  
     BufferReleased, 114  
 BufferReleased  
     BufferMgr, 114  
 BufferState, 115  
     BufferState, 115  
 BufferState  
     bind, 115  
     BufferState, 115  
     decrement\_refcnt, 115  
     get\_buf\_mgr, 115  
     get\_free\_function, 115  
     get\_ptr, 115  
     get\_refcount, 115  
     increment\_refcnt, 115  
     reset, 115  
 BufFragGroup, 116  
     BufFragGroup, 117  
 BufFragGroup  
     ~BufFragGroup, 117  
     AddFragment, 117  
     AppendNext, 117  
     buffer\_states, 118  
     BufFragGroup, 117

Clear, 117  
 fragments, 118  
 GetLength, 117  
 GetMaxFrags, 118  
 GetNext, 118  
 GetNumFrags, 118  
 length, 118  
 next, 118  
 num\_frags, 118  
**BufFragStatusClass**, 119  
 BFG\_SUCCESS, 119  
 EMPTY\_FRAGMENT, 119  
 FIXED\_FRAG\_LOC\_FULL, 119  
 INTERNAL\_ERROR, 119  
 INVALID\_ID, 119  
 NOT\_ENOUGH\_SPACE, 119  
 NULL\_INPUT, 119  
 TOO\_MANY\_FRAGS, 119  
**BufFragStatusClass**  
 status\_t, 119  
**bufsize**  
 Oscl\_Queue\_Base, 232  
 Oscl\_Vector\_Base, 284  
**BYTES\_IN\_UUID\_ARRAY**  
 oscl\_uuid.h, 780

**c**  
 OsclPriorityQueue, 459

**c\_bool**  
 osclbase, 32

**c\_str**  
 StrPtrLen, 629  
 WStrPtrLen, 639

**Callback**  
 OsclReadyQ, 471

**callback\_timer\_type**  
 OsclTimer, 566

**CallbackTimer**, 120  
 CallbackTimer, 120

**CallbackTimer< Alloc >**  
 OsclTimer, 567

**CallbackTimerObserver**, 122

**CallbackTimerObserver**  
 ~CallbackTimerObserver, 122  
 TimerBaseElapsed, 122

**CallRunExec**  
 OsclExecSchedulerCommonBase, 386

**Cancel**  
 OsclActiveObject, 302  
 OsclTimer, 566

OsclTimerObject, 570  
 PVActiveBase, 594

**CancelAccept**  
 OsclSocketIBase, 528  
 OsclTCPSocket, 550  
 OsclTCPSocketI, 556

**CancelBind**  
 OsclSocketIBase, 528  
 OsclTCPSocket, 551  
 OsclTCPSocketI, 556  
 OsclUDPSocket, 585  
 OsclUDPSocketI, 590

**CancelConnect**  
 OsclSocketIBase, 528  
 OsclTCPSocket, 551  
 OsclTCPSocketI, 556

**CancelFreeChunkAvailableCallback**  
 OsclMemPoolFixedChunkAllocator, 430  
 OsclMemPoolResizableAllocator, 436

**CancelFreeMemoryAvailableCallback**  
 OsclMemPoolResizableAllocator, 436

**CancelFxn**  
 OsclDNSIBase, 347  
 OsclSocketIBase, 528

**CancelGetHostByName**  
 OsclDNS, 342  
 OsclDNSIBase, 347

**CancelListen**  
 OsclSocketIBase, 528  
 OsclTCPSocket, 551  
 OsclTCPSocketI, 556

**CancelMethod**  
 OsclDNSMethod, 350  
 OsclSocketMethod, 532

**CancelRecv**  
 OsclSocketIBase, 528  
 OsclTCPSocket, 551  
 OsclTCPSocketI, 556

**CancelRecvFrom**  
 OsclSocketIBase, 528  
 OsclUDPSocket, 585  
 OsclUDPSocketI, 590

**CancelRequest**  
 OsclDNSRequest, 353  
 OsclSocketRequest, 535

**CancelSend**  
 OsclSocketIBase, 528  
 OsclTCPSocket, 551  
 OsclTCPSocketI, 556

**CancelSendTo**  
 OsclSocketIBase, 528  
 OsclUDPSocket, 585  
 OsclUDPSocketI, 590

**CancelShutdown**

OsclSocketIBase, 528  
 OsclTCPSocket, 551  
 OsclTCPSocketI, 556  
 capacity  
   Oscl\_Queue\_Base, 231  
   Oscl\_Vector\_Base, 282  
   OsclFileCacheBuffer, 395  
 CFastRep, 123  
   CFastRep, 124  
 CFastRep  
   append, 124  
   buffer, 124  
   CFastRep, 124  
   maxsize, 124  
   set\_r, 124  
   set\_w, 124  
   size, 124  
   writable, 124  
 chartype  
   OSCL\_FastString, 171  
   OSCL\_HeapString, 192  
   OSCL\_HeapStringA, 194  
   OSCL\_StackString, 249  
   OSCL\_String, 252  
   OSCL\_wFastString, 285  
   OSCL\_wHeapString, 288  
   OSCL\_wHeapStringA, 291  
   OSCL\_wStackString, 294  
   OSCL\_wString, 296  
 CHeapRep, 125  
   CHeapRep, 126  
 CHeapRep  
   add\_ref, 126  
   append, 126  
   append\_rep, 126  
   assign, 126  
   buffer, 126  
   CHeapRep, 126  
   maxsize, 126  
   refcount, 126  
   remove\_ref, 126  
   set, 126  
   set\_rep, 126  
   size, 126  
 check\_fence  
   MM\_AllocBlockFence, 142  
 check\_list  
   Oscl\_Linked\_List, 201  
   Oscl\_Linked\_List\_Base, 205  
 checkSum  
   StrCSumPtrLen, 626  
 CheckSumType  
   StrCSumPtrLen, 626  
 children  
   Oscl\_TagTree::Node, 271  
   children\_type  
     Oscl\_TagTree, 261  
     Oscl\_TagTree::Node, 271  
 ChooseCurCache  
   Oscl\_File::OsclCacheObserver, 182  
 CleanInUse  
   OsclAsyncFileBuffer, 311  
 Cleanup  
   OsclErrorTrap, 367  
   OsclInit, 401  
   OsclMem, 410  
   OsclScheduler, 502  
   PVLogger, 600  
 CleanupExecQ  
   OsclExecSchedulerCommonBase, 386  
 CleanupParam  
   OsclSocketRequestAO, 537  
 CleanupStatQ  
   OsclExecSchedulerCommonBase, 386  
 Clear  
   BufFragGroup, 117  
   MediaData, 138  
   OsclTimer, 566  
 clear  
   Oscl\_Map, 211  
   Oscl\_Queue, 228  
   Oscl\_Queue\_Base, 231  
   Oscl\_Rb\_Tree, 235  
   Oscl\_TagTree, 262  
   Oscl\_Vector, 278  
 Close  
   Oscl\_File, 176  
   Oscl\_FileFind, 185  
   Oscl\_FileServer, 188  
   OsclAsyncFile, 308  
   OsclDNSI, 344  
   OsclDNSIBase, 347  
   OsclFileCache, 393  
   OsclIPSocketI, 404  
   OsclMutex, 445  
   OsclNativeFile, 449  
   OsclRegistryAccessClient, 490  
   OsclRegistryClient, 495  
   OsclRegistryClientImpl, 498  
   OsclRegistryServTlsImpl, 501  
   OsclSemaphore, 507  
   OsclSocketI, 522  
   OsclSocketIBase, 528  
   OsclSocketServ, 540  
   OsclSocketServI, 542  
   OsclSocketServIBase, 545  
   OsclSocketServRequestList, 546  
   OsclTCPSocket, 551

OsclTCPSocketI, [556](#)  
 OsclUDPSocket, [585](#)  
 OsclUDPSocketI, [590](#)  
**CloseSession**  
 OsclComponentRegistry, [335](#)  
**color**  
 Oscl\_Rb\_Tree\_Node\_Base, [246](#)  
**color\_type**  
 Oscl\_Rb\_Tree\_Node\_Base, [245](#)  
**comp**  
 Oscl\_Map::value\_compare, [215](#)  
 OsclPriorityQueue, [459](#)  
**compare**  
 OsclCompareLess, [333](#)  
 OsclReadyCompare, [469](#)  
 OsclTimerCompare, [568](#)  
**compare\_data**  
 Oscl\_Opaque\_Type\_Alloc\_LL, [222](#)  
**compare\_EQ**  
 Oscl\_Opaque\_Type\_Compare, [224](#)  
 OsclPriorityQueue, [457](#)  
**compare\_LT**  
 Oscl\_Opaque\_Type\_Compare, [224](#)  
 OsclPriorityQueue, [457](#)  
**CompareId**  
 OsclThread, [558](#)  
**Complete**  
 OsclDNSRequest, [353](#)  
 OsclSocketRequest, [535](#)  
**COMPUTE\_MEM\_ALIGN\_SIZE**  
 osclmemory, [48](#)  
**Connect**  
 Oscl\_FileServer, [188](#)  
 OsclConnectMethod, [339](#)  
 OsclConnectRequest, [340](#)  
 OsclRegistryAccessClient, [490](#)  
 OsclRegistryClient, [495](#)  
 OsclRegistryClientImpl, [498](#)  
 OsclRegistryServTlsImpl, [501](#)  
 OsclSocketI, [522](#)  
 OsclSocketIBase, [528](#)  
 OsclSocketServ, [540](#)  
 OsclSocketServI, [542](#)  
 OsclSocketServIBase, [545](#)  
 OsclTCPSocket, [552](#)  
 OsclTCPSocketI, [556](#)  
**ConnectParam**, [127](#)  
 ConnectParam, [127](#)  
**ConnectParam**  
 ConnectParam, [127](#)  
 iAddr, [127](#)  
**ConnectRequest**  
 OsclConnectMethod, [339](#)  
**const\_iterator**  
 Oscl\_Map, [210](#)  
 Oscl\_Rb\_Tree, [235](#)  
 Oscl\_Rb\_Tree\_Const\_Iterator, [239](#)  
 Oscl\_TagTree::const\_iterator, [265](#)  
 Oscl\_Vector, [277](#)  
**const\_pointer**  
 Oscl\_Rb\_Tree, [235](#)  
 Oscl\_TAlloc, [273](#)  
**const\_reference**  
 Oscl\_Map, [210](#)  
 Oscl\_Queue, [228](#)  
 Oscl\_Rb\_Tree, [235](#)  
 Oscl\_TAlloc, [273](#)  
 Oscl\_Vector, [277](#)  
 OsclPriorityQueue, [457](#)  
**Construct**  
 OsclReadyQ, [471](#)  
 OsclTimerQ, [574](#)  
**construct**  
 Oscl\_Linked\_List\_Base, [205](#)  
 Oscl\_Opaque\_Type\_Alloc, [221](#)  
 Oscl\_Opaque\_Type\_Alloc\_LL, [222](#)  
 Oscl\_Queue\_Base, [231](#)  
 Oscl\_TAlloc, [273](#)  
 Oscl\_Vector\_Base, [282](#)  
 OsclPriorityQueueBase, [460](#)  
**ConstructL**  
 OsclDNSMethod, [350](#)  
 OsclDNSRequestAO, [355](#)  
 OsclExecSchedulerCommonBase, [386](#)  
 OsclIPSocketI, [404](#)  
 OsclSocketMethod, [532](#)  
 OsclSocketRequestAO, [537](#)  
**ConstructStatQ**  
 OsclExecSchedulerCommonBase, [386](#)  
**container\_type**  
 OsclPriorityQueue, [457](#)  
**Contains**  
 Oscl\_File::OsclFixedCacheParam, [183](#)  
 OsclFileCacheBuffer, [395](#)  
**count**  
 Oscl\_Map, [211](#)  
 Oscl\_Rb\_Tree, [235](#)  
 Oscl\_TagTree, [262](#)  
**CPVInterfaceProxy**  
 OsclErrorTrapImp, [369](#)  
**Create**  
 GetHostByNameParam, [131](#)  
 OsclMutex, [445](#)  
 OsclSemaphore, [507](#)  
 OsclThread, [559](#)  
**CreateMemPool**  
 OsclMemPoolAllocator, [428](#)  
**createmempool**

OsclMemPoolFixedChunkAllocator, 430  
 CreatePVLogger  
   PVLoggerRegistry, 610  
 createStatsNode  
   MM\_Audit\_Imp, 149  
 CStackRep, 128  
   CStackRep, 128  
 CStackRep  
   append, 128  
   buffer, 128  
   CStackRep, 128  
   maxsize, 128  
   set, 128  
   size, 128  
 CTIME\_BUFFER\_SIZE  
   osclbase, 43  
 CtimeStrBuf  
   osclbase, 32  
 Current  
   OsclExecScheduler, 380  
 currentPos  
   OsclFileCacheBuffer, 395  
  
 data  
   LinkedListElement, 135  
 data1  
   OsclUuid, 592  
 data2  
   OsclUuid, 592  
 data3  
   OsclUuid, 592  
 data4  
   OsclUuid, 592  
 deallocate  
   \_OsclBasicAllocator, 105  
   MemAllocator, 141  
   Oscl\_Dealloc, 166  
   Oscl\_DefAlloc, 167  
   Oscl\_Opaque\_Type\_Alloc, 221  
   Oscl\_Opaque\_Type\_Alloc\_LL, 222  
   Oscl\_TAlloc, 273  
   OsclErrorAllocator, 365  
   OsclMemAllocator, 411  
   OsclMemAllocDestructDealloc, 412  
   OSCLMemAutoPtr, 422  
   OsclMemBasicAllocator, 424  
   OsclMemBasicAllocDestructDealloc, 425  
   OsclMemPoolFixedChunkAllocator, 431  
   OsclMemPoolResizableAllocator, 436  
   OsclReadyAlloc, 468  
 deallocateblock  
   OsclMemPoolResizableAllocator, 436  
 decrement\_refcnt  
   BufferState, 115  
  
 DEFAULT\_MM\_AUDIT\_MODE  
   osclmemory, 49  
 DEFAULT\_POSTFILL\_PATTERN  
   osclmemory, 49  
 DEFAULT\_PREFILL\_PATTERN  
   osclmemory, 49  
 Delete  
   Oscl\_DefAllocWithRefCounter, 168  
   OsclAsyncFile, 308  
   OsclBuf, 332  
 Depth  
   OsclReadyQ, 471  
 depth  
   Oscl\_TagTree::Node, 271  
 dequeue\_element  
   Oscl\_Linked\_List, 201  
   Oscl\_MTLinked\_List, 218  
 Des  
   OsclBuf, 332  
 DesC  
   OsclBuf, 332  
 Destroy  
   DNSRequestParam, 129  
   GetHostNameParam, 131  
   PVActiveBase, 594  
 destroy  
   Oscl\_Linked\_List\_Base, 205  
   Oscl\_Opaque\_Type\_Alloc, 221  
   Oscl\_Opaque\_Type\_Alloc\_LL, 222  
   Oscl\_Queue\_Base, 231  
   Oscl\_TAlloc, 273  
   Oscl\_Vector, 278  
   Oscl\_Vector\_Base, 282  
 destroyallmempoolbuffers  
   OsclMemPoolResizableAllocator, 436  
 DestroyMemPool  
   OsclMemPoolAllocator, 428  
 destroymempool  
   OsclMemPoolFixedChunkAllocator, 431  
 destruct\_and\_dealloc  
   Oscl\_TAlloc, 273  
   OsclDestructDealloc, 341  
   OsclMemAllocDestructDealloc, 412  
   OsclMemBasicAllocDestructDealloc, 425  
 difference\_type  
   Oscl\_Rb\_Tree, 235  
 DIR\_TYPE  
   Oscl\_FileFind, 184  
 DisableAppenderInheritance  
   PVLogger, 600  
 DiscardAcceptedSocket  
   OsclAcceptMethod, 299  
 DNSRequestParam, 129  
   DNSRequestParam, 129

OsclDNSI, 345  
 OsclDNSRequestAO, 356  
**DNSRequestParam**  
 ~DNSRequestParam, 129  
 Destroy, 129  
 DNSRequestParam, 129  
 iDNSRequest, 130  
 iFxn, 130  
 InThread, 129  
 iRefCount, 130  
 RemoveRef, 130  
**DoCancel**  
 OsclActiveObject, 303  
 OsclDNSRequestAO, 355  
 OsclSocketRequestAO, 537  
 OsclTimerObject, 570  
 PVActiveBase, 594  
**E\_BUFFER\_TOO\_SMALL**  
 Oscl\_FileFind, 185  
**E\_INVALID\_ARG**  
 Oscl\_FileFind, 184  
**E\_INVALID\_STATE**  
 Oscl\_FileFind, 184  
**E\_MEMORY\_ERROR**  
 Oscl\_FileFind, 185  
**E\_NO\_MATCH**  
 Oscl\_FileFind, 185  
**E\_NOT\_IMPLEMENTED**  
 Oscl\_FileFind, 185  
**E\_OK**  
 Oscl\_FileFind, 184  
**E\_OTHER**  
 Oscl\_FileFind, 185  
**E\_PATH\_NOT\_FOUND**  
 Oscl\_FileFind, 184  
**E\_PATH\_TOO\_LONG**  
 Oscl\_FileFind, 184  
**element\_type**  
 Oscl\_FileFind, 184  
**elems**  
 Oscl\_Queue\_Base, 232  
 Oscl\_Vector\_Base, 284  
**empty**  
 Oscl\_Map, 211  
 Oscl\_Queue\_Base, 231  
 Oscl\_Rb\_Tree, 235  
 Oscl\_TagTree, 262  
 Oscl\_Vector\_Base, 282  
 OsclPriorityQueue, 458  
**EMPTY\_FRAGMENT**  
 BufFragStatusClass, 119  
**EnableKill**  
 OsclThread, 559  
  
 enablenullpointerreturn  
 OsclMemPoolFixedChunkAllocator, 431  
 OsclMemPoolResizableAllocator, 436  
**End**  
 OsclFileStats, 397  
**end**  
 Oscl\_Map, 211  
 Oscl\_Rb\_Tree, 235  
 Oscl\_TagTree, 262  
 Oscl\_Vector, 278  
**EndOfFile**  
 Oscl\_File, 176  
 OsclAsyncFile, 308  
 OsclFileCache, 393  
 OsclNativeFile, 449  
**endPos**  
 OsclFileCacheBuffer, 395  
**EndScheduling**  
 OsclExecSchedulerCommonBase, 386  
**EndStats**  
 OsclExecSchedulerCommonBase, 386  
**EnterThreadContext**  
 PVThreadContext, 615  
**eof**  
 OsclBinStream, 328  
**EOF\_STATE**  
 OsclBinStream, 328  
**EOsclFileOp\_Close**  
 osclio, 94  
**EOsclFileOp\_EndOfFile**  
 osclio, 94  
**EOsclFileOp\_Flush**  
 osclio, 94  
**EOsclFileOp\_Last**  
 osclio, 95  
**EOsclFileOp\_NativeClose**  
 osclio, 94  
**EOsclFileOp\_NativeEndOfFile**  
 osclio, 95  
**EOsclFileOp\_NativeFlush**  
 osclio, 95  
**EOsclFileOp\_NativeOpen**  
 osclio, 94  
**EOsclFileOp\_NativeRead**  
 osclio, 94  
**EOsclFileOp\_NativeSeek**  
 osclio, 95  
**EOsclFileOp\_NativeSize**  
 osclio, 95  
**EOsclFileOp\_NativeTell**  
 osclio, 95  
**EOsclFileOp\_NativeWrite**  
 osclio, 94  
**EOsclFileOp\_Open**

osclio, 94  
 EOscIFileOp\_Read  
     osclio, 94  
 EOscIFileOp\_Seek  
     osclio, 94  
 EOscIFileOp\_Size  
     osclio, 94  
 EOscIFileOp\_Tell  
     osclio, 94  
 EOscIFileOp\_Write  
     osclio, 94  
 eOscIProcError  
     OsclProcStatus, 461  
 EOscISocket\_DataRecv  
     oscl\_socket\_stats.h, 752  
 EOscISocket\_DataSent  
     oscl\_socket\_stats.h, 752  
 EOscISocket\_Except  
     oscl\_socket\_stats.h, 751  
 EOscISocket\_OS  
     oscl\_socket\_stats.h, 751  
 EOscISocket\_Readable  
     oscl\_socket\_stats.h, 751  
 EOscISocket\_RequestAO\_Canceled  
     oscl\_socket\_stats.h, 751  
 EOscISocket\_RequestAO\_Error  
     oscl\_socket\_stats.h, 751  
 EOscISocket\_RequestAO\_Success  
     oscl\_socket\_stats.h, 751  
 EOscISocket\_RequestAO\_Timeout  
     oscl\_socket\_stats.h, 751  
 EOscISocket\_ServPoll  
     oscl\_socket\_stats.h, 751  
 EOscISocket\_ServRequestCancelIssued  
     oscl\_socket\_stats.h, 752  
 EOscISocket\_ServRequestComplete  
     oscl\_socket\_stats.h, 752  
 EOscISocket\_ServRequestIssued  
     oscl\_socket\_stats.h, 751  
 EOscISocket\_Writable  
     oscl\_socket\_stats.h, 751  
 EOscISocketServ\_LastEvent  
     oscl\_socket\_stats.h, 751  
 EOscISocketServ\_LoopsockError  
     oscl\_socket\_stats.h, 752  
 EOscISocketServ\_LoopsockOk  
     oscl\_socket\_stats.h, 752  
 EOscISocketServ\_SelectActivity  
     oscl\_socket\_stats.h, 751  
 EOscISocketServ\_SelectNoActivity  
     oscl\_socket\_stats.h, 751  
 EOscISocketServ\_SelectRescheduleAsap  
     oscl\_socket\_stats.h, 751  
 EOscISocketServ\_SelectReschedulePoll  
     oscl\_socket\_stats.h, 751  
 oscl\_socket\_stats.h, 751  
 EOOtherExecStats\_Last  
     OsclExecSchedulerCommonBase, 385  
 EOOtherExecStats\_NativeOS  
     OsclExecSchedulerCommonBase, 385  
 EOOtherExecStats\_QueueTime  
     OsclExecSchedulerCommonBase, 385  
 EOOtherExecStats\_ReleaseTime  
     OsclExecSchedulerCommonBase, 385  
 EOOtherExecStats\_WaitTime  
     OsclExecSchedulerCommonBase, 385  
 EPriorityHigh  
     OsclActiveObject, 302  
 EPriorityHighest  
     OsclActiveObject, 302  
 EPriorityIdle  
     OsclActiveObject, 302  
 EPriorityLow  
     OsclActiveObject, 302  
 EPriorityNominal  
     OsclActiveObject, 302  
 EPVDNSCancel  
     osclio, 95  
 EPVDNSFailure  
     osclio, 95  
 EPVDNSGetHostByName  
     osclio, 95  
 EPVDNSPending  
     osclio, 95  
 EPVDNSSuccess  
     osclio, 95  
 EPVDNSTimeout  
     osclio, 95  
 EPVSocket\_Last  
     oscl\_socket\_types.h, 756  
 EPVSocketAccept  
     oscl\_socket\_types.h, 756  
 EPVSocketBind  
     oscl\_socket\_types.h, 756  
 EPVSocketBothShutdown  
     oscl\_socket\_types.h, 756  
 EPVSocketCancel  
     oscl\_socket\_types.h, 755  
 EPVSocketConnect  
     oscl\_socket\_types.h, 756  
 EPVSocketFailure  
     oscl\_socket\_types.h, 755  
 EPVSocketListen  
     oscl\_socket\_types.h, 756  
 EPVSocketPending  
     oscl\_socket\_types.h, 755  
 EPVSocketRecv  
     oscl\_socket\_types.h, 756  
 EPVSocketRecvFrom  
     oscl\_socket\_types.h, 756

oscl\_socket\_types.h, 756  
 EPVSocketRecvShutdown  
     oscl\_socket\_types.h, 756  
 EPVSocketSend  
     oscl\_socket\_types.h, 756  
 EPVSocketSendShutdown  
     oscl\_socket\_types.h, 756  
 EPVSocketSendTo  
     oscl\_socket\_types.h, 756  
 EPVSocketShutdown  
     oscl\_socket\_types.h, 756  
 EPVSocketSuccess  
     oscl\_socket\_types.h, 755  
 EPVSocketTimeout  
     oscl\_socket\_types.h, 755  
 EPVThreadContext\_InThread  
     osclproc, 102  
 EPVThreadContext\_NonOsclThread  
     osclproc, 102  
 EPVThreadContext\_OsclThread  
     osclproc, 102  
 EPVThreadContext\_Undetermined  
     osclproc, 102  
 equal\_range  
     Oscl\_Map, 211  
     Oscl\_Rb\_Tree, 235  
 erase  
     Oscl\_Map, 212  
     Oscl\_Rb\_Tree, 235  
     Oscl\_TagTree, 262  
     Oscl\_Vector, 278  
     Oscl\_Vector\_Base, 282, 283  
 Error  
     OsclExecSchedulerCommonBase, 386  
 error\_type  
     Oscl\_FileFind, 184  
 ESocketServ\_Connected  
     OsclSocketServIBase, 544  
 ESocketServ\_Error  
     OsclSocketServIBase, 545  
 ESocketServ\_Idle  
     OsclSocketServIBase, 544  
 ESymbianAccessMode\_Rfile  
     Oscl\_File, 175  
 ESymbianAccessMode\_RfileBuf  
     Oscl\_File, 175  
 EXCEED\_MAX\_COUNT\_VARIABLE\_-  
     ERROR  
     OsclProcStatus, 462  
 EXCEED\_MAX\_SEM\_COUNT\_ERROR  
     OsclProcStatus, 462  
 Exit  
     OsclThread, 559  
 ExitThreadContext  
     PVThreadContext, 615  
 extract\_string  
     osclutil, 66  
 fail  
     OsclBinStream, 329  
 FAIL\_STATE  
     OsclBinStream, 328  
 FENCE\_PATTERN  
     osclmemory, 49  
 FILE\_TYPE  
     Oscl\_FileFind, 184  
 fileName  
     MM\_AllocQueryInfo, 147  
 filePosition  
     OsclFileCacheBuffer, 395  
 FileSize  
     OsclFileCache, 393  
 fill\_fence  
     MM\_AllocBlockFence, 142  
 FillFromFile  
     OsclFileCacheBuffer, 395  
 filter\_status\_type  
     AllPassFilter, 110  
     PVLogger, 599  
     PVLoggerFilter, 605  
 FilterOpaqueMessge  
     AllPassFilter, 111  
     PVLoggerFilter, 606  
 FilterString  
     AllPassFilter, 111  
     PVLoggerFilter, 606  
 Find  
     OsclComponentRegistryData, 336  
 find  
     Oscl\_Map, 212  
     Oscl\_Rb\_Tree, 235  
     Oscl\_TagTree, 262  
 find\_heap  
     OsclPriorityQueue, 458  
     OsclPriorityQueueBase, 460  
 FindExact  
     OsclComponentRegistry, 335  
 FindFirst  
     Oscl\_FileFind, 185  
 findfreeblock  
     OsclMemPoolResizableAllocator, 437  
 FindHierarchical  
     OsclComponentRegistry, 335  
 FindNext  
     Oscl\_FileFind, 186  
 FindPVBase  
     OsclExecSchedulerCommonBase, 386  
 first

Oscl\_Pair, 226  
 firstFragPtr  
   OsclBinStream, 330  
 FIXED\_FRAG\_LOC\_FULL  
   BufFragStatusClass, 119  
 Flush  
   Oscl\_File, 176  
   OsclAsyncFile, 308  
   OsclFileCache, 393  
   OsclNativeFile, 449  
 FormatOpaqueMessage  
   PVLoggerLayout, 607  
 FormatString  
   PVLoggerLayout, 607  
 fragments  
   BufFragGroup, 118  
 fragsLeft  
   OsclBinStream, 330  
 freeblockavailable  
   OsclMemPoolResizableAllocatorObserver,  
     443  
 freebytes  
   oscl\_fsstat, 190  
 freechunkavailable  
   OsclMemPoolFixedChunkAllocator-  
     Observer, 433  
 freememoryavailable  
   OsclMemPoolResizableAllocatorMemory-  
     Observer, 442  
 front  
   Oscl\_Queue, 229  
   Oscl\_Vector, 279  
 Fxn  
   OsclSocketRequest, 535  
  
 get  
   OsclBinIStream, 315  
   OsclExclusiveArrayPtr, 372  
   OsclExclusivePtr, 375  
   OsclExclusivePtrA, 378  
   OSCLMemAutoPtr, 422  
 get\_buf\_mgr  
   BufferState, 115  
 get\_count  
   OsclSharedPtr, 514  
 get\_cstr  
   OSCL\_FastString, 171  
   OSCL\_HeapStringA, 195  
   OSCL\_String, 252  
   OSCL\_wFastString, 286  
   OSCL\_wHeapStringA, 291  
   OSCL\_wString, 296  
   osclutil, 66, 67  
 get\_data

Oscl\_Opaque\_Type\_Alloc\_LL, 223  
 get\_element  
   Oscl\_Linked\_List, 201  
   Oscl\_Linked\_List\_Base, 205  
   Oscl\_MTLLinked\_List, 218  
 get\_first  
   Oscl\_Linked\_List, 201  
   Oscl\_Linked\_List\_Base, 206  
 get\_free\_function  
   BufferState, 115  
 get\_index  
   Oscl\_Linked\_List, 202  
   Oscl\_Linked\_List\_Base, 206  
   Oscl\_MTLLinked\_List, 218  
 get\_int64\_lower32  
   Oscl\_Int64\_Utils, 198  
 get\_int64\_middle32  
   Oscl\_Int64\_Utils, 198  
 get\_int64\_upper32  
   Oscl\_Int64\_Utils, 198  
 get\_local\_time  
   TimeValue, 632  
 get\_lower32  
   NTPTime, 163  
 get\_maxsize  
   OSCL\_FastString, 171  
   OSCL\_HeapStringA, 195  
   OSCL\_String, 252  
   OSCL\_wFastString, 286  
   OSCL\_wHeapStringA, 291  
   OSCL\_wString, 296  
   osclutil, 67  
 get\_middle32  
   NTPTime, 163  
 get\_next  
   Oscl\_Linked\_List, 202  
   Oscl\_Linked\_List\_Base, 206  
   Oscl\_Opaque\_Type\_Alloc\_LL, 223  
 get\_num\_elements  
   Oscl\_Linked\_List, 202  
 get\_ptr  
   BufferState, 115  
 get\_pv8601\_str\_time  
   TimeValue, 632  
 get\_refcount  
   BufferState, 115  
 get\_registry  
   TLSStorageOps, 636  
 get\_rfc822\_gmtime\_str  
   TimeValue, 632  
 get\_sec  
   TimeValue, 633  
 get\_size  
   OSCL\_FastString, 172

OSCL\_HeapStringA, 195  
 OSCL\_String, 252  
 OSCL\_wFastString, 286  
 OSCL\_wHeapStringA, 291  
 OSCL\_wString, 296  
 osclutil, 67, 68  
**get\_str**  
   OSCL\_FastString, 172  
   OSCL\_HeapStringA, 196  
   OSCL\_String, 253  
   OSCL\_wFastString, 286  
   OSCL\_wHeapStringA, 291  
   OSCL\_wString, 296  
   osclutil, 68  
**get\_str\_ctime**  
   TimeValue, 633  
**get\_timeval\_ptr**  
   TimeValue, 633  
**get\_uint64\_lower32**  
   Oscl\_Int64\_Utils, 198  
**get\_uint64\_middle32**  
   Oscl\_Int64\_Utils, 198  
**get\_uint64\_upper32**  
   Oscl\_Int64\_Utils, 198  
**get\_upper32**  
   NTPTime, 163  
**get\_usec**  
   TimeValue, 633  
**get\_value**  
   NTPTime, 163  
**GetAcceptedSocket**  
   OsclAcceptMethod, 299  
**GetAcceptedSocketL**  
   OsclTCPSocket, 552  
   OsclTCPSocketI, 556  
**getAllocatedSize**  
   OsclMemPoolResizableAllocator, 437  
**getAuditRoot**  
   MM\_Audit\_Imp, 149  
**GetAvailableBufferSize**  
   MediaData, 138  
**getAvailableSize**  
   OsclMemPoolResizableAllocator, 437  
**getBufferSize**  
   OsclMemPoolResizableAllocator, 437  
**GetBufferState**  
   osclutil, 68  
**getCapacity**  
   OsclRefCounterMemFrag, 483  
**getCheckSum**  
   StrCSumPtrLen, 626  
**getCount**  
   Oscl\_DefAllocWithRefCounter, 168  
   OsclRefCounter, 478  
   OsclRefCounterDA, 481  
   OsclRefCounterMemFrag, 483  
   OsclRefCounterMTDA, 485  
   OsclRefCounterMTSA, 487  
   OsclRefCounterSA, 489  
**GetElementType**  
   Oscl\_FileFind, 186  
**GetError**  
   Oscl\_File, 177  
   OsclNativeFile, 449  
**GetErrorTrapImp**  
   OsclErrorTrap, 367  
**GetFactories**  
   OsclRegistryAccessClient, 490  
   OsclRegistryClientImpl, 498  
   OsclRegistryServTlsImpl, 501  
**GetFactory**  
   OsclRegistryAccessClient, 490  
   OsclRegistryClientImpl, 498  
   OsclRegistryServTlsImpl, 501  
**GetFragment**  
   osclutil, 69  
**getGlobalMemAuditObject**  
   OsclMemGlobalAuditObject, 426  
**getHead**  
   OsclDoubleListBase, 360  
**GetHostByName**  
   OsclDNS, 343  
   OsclDNSI, 344  
   OsclDNSIBase, 347  
   OsclGetHostByNameMethod, 399  
**GetHostByNameParam**, 131  
**GetHostByNameParam**  
   ~GetHostByNameParam, 131  
   Create, 131  
   Destroy, 131  
   iAddr, 131  
   iName, 131  
**GetHostByNameSuccess**  
   OsclDNSI, 344  
   OsclDNSIBase, 347  
**GetId**  
   OsclExecSchedulerCommonBase, 386  
   OsclThread, 559  
**getInstance**  
   OsclSingletonRegistry, 520  
   OsclTLSRegistry, 579  
   OsclTLSRegistryEx, 580  
**getLargestContiguousFreeBlockSize**  
   OsclMemPoolResizableAllocator, 437  
**GetLastError**  
   Oscl\_FileFind, 186  
**getLeaveCode**  
   OsclException, 370

GetLength  
     BufFragGroup, 117  
 GetLocalBufsize  
     MediaData, 139  
 GetLocalFragment  
     MediaData, 139  
 GetLock  
     OsclMemAudit, 415  
 GetLoggerObject  
     PVLogger, 600  
 GetLogLevel  
     PVLogger, 600  
 GetMaxFrags  
     BufFragGroup, 118  
 GetMediaFragment  
     MediaData, 139  
 GetMediaSize  
     MediaData, 139  
 getMemFrag  
     OsclRefCounterMemFrag, 483  
 getMemFragPtr  
     OsclRefCounterMemFrag, 483  
 getMemFragSize  
     OsclRefCounterMemFrag, 483  
 getMemPoolBufferAllocatedSize  
     OsclMemPoolResizableAllocator, 437  
 getMemPoolBufferSize  
     OsclMemPoolResizableAllocator, 437  
 GetName  
     OsclExecSchedulerCommonBase, 386  
 GetNext  
     BufFragGroup, 118  
 GetNumAppenders  
     PVLogger, 600  
 GetNumFrags  
     BufFragGroup, 118  
 GetNumMediaFrags  
     MediaData, 139  
 getOffset  
     OsclDoubleListBase, 360  
 GetParent  
     PVLogger, 601  
 GetPriority  
     OsclThread, 560  
 GetPVLoggerObject  
     PVLoggerRegistry, 610  
 GetPVLoggerRegistry  
     PVLoggerRegistry, 610  
 GetReadAsyncNumElements  
     OsclNativeFile, 449  
 GetRecvData  
     OsclIPSocketI, 404  
     OsclRecvFromMethod, 472  
     OsclRecvFromRequest, 474  
     OsclRecvMethod, 476  
     OsclRecvRequest, 477  
     OsclTCPSocket, 552  
     OsclTCPSocketI, 556  
     OsclUDPSocket, 586  
     OsclUDPSocketI, 590  
 GetRefCounter  
     OsclSharedPtr, 514  
 getRefCounter  
     OsclRefCounterMemFrag, 483  
 GetRep  
     OsclSharedPtr, 514  
 GetScheduler  
     OsclExecSchedulerCommonBase, 386  
 GetSendData  
     OsclIPSocketI, 404  
     OsclSendMethod, 509  
     OsclSendRequest, 510  
     OsclSendToMethod, 511  
     OsclSendToRequest, 512  
     OsclTCPSocket, 552  
     OsclTCPSocketI, 556  
     OsclUDPSocket, 586  
     OsclUDPSocketI, 590  
 GetShutdown  
     OsclSocketIBase, 528  
 getSize  
     MM\_Audit\_Imp, 149  
 GetSocketError  
     OsclDNSRequestAO, 355  
     OsclSocketRequestAO, 537  
 getTagActualSize  
     MM\_Audit\_Imp, 149  
 GetTimestamp  
     MediaData, 139  
 good  
     OsclBinStream, 329  
 GOOD\_STATE  
     OsclBinStream, 328  
 Handle  
     Oscl\_File, 177  
     OsclFileHandle, 396  
 HandleDNSEvent  
     OsclDNSObserver, 352  
 HandleSocketEvent  
     OsclSocketObserver, 534  
 HasAsyncBind  
     OsclSocketIBase, 528  
 HasAsyncListen  
     OsclSocketIBase, 528  
 HasAsyncRead  
     OsclNativeFile, 449  
 hash

OSCL\_String, 253  
 OSCL\_wString, 296  
**HasThisOffset**  
 OsclAsyncFileBuffer, 311  
**HaveRoomInCurrentBlock**  
 OsclBinStream, 329  
**Head**  
 OsclDoubleList, 358  
 OsclPriorityList, 455  
**head**  
 Oscl\_Linked\_List\_Base, 207  
**HeapBase**, 132  
 HeapBase, 133  
**HeapBase**  
 ~HeapBase, 133  
 HeapBase, 133  
**host\_to\_big\_endian**  
 osclbase, 33  
**host\_to\_little\_endian**  
 osclbase, 34  
  
**iActive**  
 OsclDNSRequest, 353  
**iAddedNum**  
 PVActiveBase, 596  
**iAddr**  
 BindParam, 112  
 ConnectParam, 127  
 GetHostByNameParam, 131  
 RecvFromParam, 617  
 SendToParam, 621  
**iAddress**  
 OsclIPSocketI, 405  
**iAlloc**  
 OsclDNSIBase, 347  
 OsclDNSMethod, 351  
 OsclExecSchedulerCommonBase, 390  
 OsclIPSocketI, 405  
 OsclSocketIBase, 530  
 OsclSocketServIBase, 545  
**iAllocatedSz**  
 OsclMemPoolResizableAllocator::Mem-  
 PoolBufferInfo, 441  
**iAOPriority**  
 TReadyQueLink, 637  
**iAsyncReadBufferSize**  
 OsclNativeFileParams, 451  
**iBlankSocket**  
 AcceptParam, 108  
**iBlockBuffer**  
 OsclMemPoolResizableAllocator::Mem-  
 PoolBlockInfo, 440  
**iBlockInfoAlignedSize**  
 OsclMemPoolResizableAllocator, 439  
  
 iBlockingMode  
 OsclExecSchedulerCommonBase, 390  
**iBlockPostFence**  
 OsclMemPoolResizableAllocator::Mem-  
 PoolBlockInfo, 440  
**iBlockPreFence**  
 OsclMemPoolResizableAllocator::Mem-  
 PoolBlockInfo, 440  
**iBlockSize**  
 OsclMemPoolResizableAllocator::Mem-  
 PoolBlockInfo, 440  
**iBuffer**  
 OsclBuf, 332  
**iBufferInfoAlignedSize**  
 OsclMemPoolResizableAllocator, 439  
**iBufferPostFence**  
 OsclMemPoolResizableAllocator::Mem-  
 PoolBufferInfo, 441  
**iBufferPrefence**  
 OsclMemPoolResizableAllocator::Mem-  
 PoolBufferInfo, 441  
**iBufferSize**  
 OsclMemPoolResizableAllocator::Mem-  
 PoolBufferInfo, 441  
**iBufRecv**  
 RecvFromParam, 617  
 RecvParam, 619  
**iBufSend**  
 SendParam, 620  
 SendToParam, 621  
**iBusy**  
 PVActiveBase, 596  
**iCancel**  
 OsclSocketServRequestQElem, 548  
**iCBase**  
 OsclTrapStackItem, 583  
**iCheckFreeMemoryAvailable**  
 OsclMemPoolResizableAllocator, 439  
**iCheckNextAvailable**  
 OsclMemPoolResizableAllocator, 439  
**iCheckNextAvailableFreeChunk**  
 OsclMemPoolFixedChunkAllocator, 432  
**iChunkSize**  
 OsclMemPoolFixedChunkAllocator, 432  
**iChunkSizeMemAligned**  
 OsclMemPoolFixedChunkAllocator, 432  
**iComponentId**  
 OsclComponentRegistryElement, 337  
**iComponentIdCounter**  
 OsclComponentRegistry, 335  
**iContainer**  
 OsclFileCacheBuffer, 395  
 OsclSocketMethod, 533  
 OsclSocketRequestAO, 539

Id  
   OsclAsyncFileBuffer, 311  
   OsclSocketRequestAO, 538  
   PVThreadContext, 615  
 iData  
   OsclComponentRegistry, 335  
 iDebugLogger  
   OsclExecSchedulerCommonBase, 390  
 iDefAlloc  
   OsclExecSchedulerCommonBase, 390  
 iDelta  
   OsclExecSchedulerCommonBase, 390  
 iDNSFxN  
   OsclDNSMethod, 351  
 iDNSI  
   OsclDNSRequestAO, 356  
 iDNSMethod  
   OsclDNSRequestAO, 356  
 iDNSObserver  
   OsclDNSMethod, 351  
 iDNSRequest  
   DNSRequestParam, 130  
 iDNSRequestAO  
   OsclDNSMethod, 351  
   OsclDNSRequest, 353  
 iDNSRequestParam  
   OsclDNSRequest, 353  
 iDoStop  
   OsclExecSchedulerCommonBase, 390  
 iDoSuspend  
   OsclExecSchedulerCommonBase, 390  
 iEnableNullPtrReturn  
   OsclMemPoolFixedChunkAllocator, 432  
   OsclMemPoolResizableAllocator, 439  
 iEndAddr  
   OsclMemPoolResizableAllocator::Mem-  
     PoolBufferInfo, 441  
 iErrAlloc  
   OsclSelect, 506  
 iErrorTrapImp  
   OsclExecSchedulerCommonBase, 390  
 iExecTimerQ  
   OsclExecSchedulerCommonBase, 390  
 iExpectedNumBlocksPerBuffer  
   OsclMemPoolResizableAllocator, 439  
 iFactory  
   OsclComponentRegistryElement, 337  
   OsclRegistryAccessElement, 494  
 iFilePosition  
   Oscl\_File::OsclFixedCacheParam, 183  
 iFlags  
   RecvFromParam, 617  
   RecvParam, 619  
   SendParam, 620  
 SendToParam, 621  
 iFreeMemChunkList  
   OsclMemPoolFixedChunkAllocator, 432  
 iFreeMemContextData  
   OsclMemPoolResizableAllocator, 439  
 iFreeMemPoolObserver  
   OsclMemPoolResizableAllocator, 439  
 ifront  
   Oscl\_Queue\_Base, 232  
 iFxn  
   DNSRequestParam, 130  
   SocketRequestParam, 624  
 iGrandTotalTicks  
   OsclExecSchedulerCommonBase, 390  
 iHead  
   OsclDoubleListBase, 360  
   OsclDoubleRunner, 361  
 iHeapCheck  
   OsclSelect, 506  
 iHigh  
   OsclInteger64Transport, 402  
 iHow  
   ShutdownParam, 622  
 iId  
   OsclComponentRegistryElement, 337  
   OsclDNSMethod, 351  
   OsclIPSocketI, 405  
 iIsIn  
   TReadyQueLink, 637  
 iJumpData  
   OsclErrorTrapImp, 369  
 iLeave  
   OsclErrorTrapImp, 369  
 iLen  
   PVSockBufRecv, 613  
   PVSockBufSend, 614  
 iLength  
   OsclBuf, 332  
 iLogger  
   OsclDNSMethod, 351  
   OsclDNSRequestAO, 356  
   OsclExecSchedulerCommonBase, 390  
   OsclIPSocketI, 405  
   OsclSocketServIBase, 545  
 iLogPerfIndentStr  
   OsclExecSchedulerCommonBase, 390  
 iLogPerfIndentStrLen  
   OsclExecSchedulerCommonBase, 390  
 iLogPerfTotal  
   OsclExecSchedulerCommonBase, 390  
 iLow  
   OsclInteger64Transport, 402  
 iMaxLen  
   PVSockBufRecv, 613

iMaxLength  
     OsclBuf, 332

iMaxNewMemPoolBufferSz  
     OsclMemPoolResizableAllocator, 439

iMemPool  
     OsclMemPoolFixedChunkAllocator, 432

iMemPoolAllocator  
     OsclMemPoolFixedChunkAllocator, 432

iMemPoolBufferAllocator  
     OsclMemPoolResizableAllocator, 439

iMemPoolBufferList  
     OsclMemPoolResizableAllocator, 439

iMemPoolBufferNumLimit  
     OsclMemPoolResizableAllocator, 439

iMemPoolBufferSize  
     OsclMemPoolResizableAllocator, 439

iMimeType  
     OsclRegistryAccessElement, 494

iMultiMaxLen  
     RecvFromParam, 617

iMutex  
     OsclComponentRegistry, 335

iName  
     GetHostNameParam, 131  
     OsclExecSchedulerCommonBase, 390  
     PVActiveBase, 596

iNativeAccessMode  
     OsclNativeFileParams, 451

iNativeBufferSize  
     OsclNativeFileParams, 451

iNativeMode  
     OsclExecSchedulerCommonBase, 390

IncLogPerf  
     OsclExecSchedulerCommonBase, 387

increment\_refcnt  
     BufferState, 115

iNext  
     OsclDoubleLink, 357  
     OsclDoubleRunner, 361  
     OsclTrapStackItem, 583

iNextAvailableContextData  
     OsclMemPoolFixedChunkAllocator, 432  
     OsclMemPoolResizableAllocator, 439

iNextFreeBlock  
     OsclMemPoolResizableAllocator::Mem-  
       PoolBlockInfo, 440  
     OsclMemPoolResizableAllocator::Mem-  
       PoolBufferInfo, 441

Init  
     OsclErrorTrap, 367  
     OsclInit, 401  
     OsclMem, 410  
     OsclScheduler, 502  
     PVLogger, 601

InitExecQ  
     OsclExecSchedulerCommonBase, 387

Insert  
     OsclDoubleListBase, 360  
     OsclPriorityList, 455

insert  
     Oscl\_Map, 212  
     Oscl\_TagTree, 263  
     Oscl\_Vector, 279  
     Oscl\_Vector\_Base, 283

insert\_unique  
     Oscl\_Rb\_Tree, 235

InsertAfter  
     OsclDoubleLink, 357

InsertBefore  
     OsclDoubleLink, 357

InsertHead  
     OsclDoubleList, 358  
     OsclDoubleListBase, 360

InsertTail  
     OsclDoubleList, 358  
     OsclDoubleListBase, 360

InstallScheduler  
     OsclExecSchedulerCommonBase, 387

INT64  
     osclconfig\_unix\_android.h, 818  
     osclconfig\_unix\_common.h, 822

int64  
     osclbase, 32

INT64\_HILO  
     osclconfig\_unix\_android.h, 818  
     osclconfig\_unix\_common.h, 822

INTERNAL\_ERROR  
     BuffFragStatusClass, 119

internalLeave, 134  
     osclerror, 84

internalLeave  
     a, 134

InThread  
     DNSRequestParam, 129

iNumAOAdded  
     OsclExecSchedulerCommonBase, 390

iNumChunk  
     OsclMemPoolFixedChunkAllocator, 432

iNumOfRun  
     OsclAsyncFile, 309

iNumOfRunErr  
     OsclAsyncFile, 309

iNumOutstanding  
     OsclMemPoolResizableAllocator::Mem-  
       PoolBufferInfo, 441

iNumSessions  
     OsclComponentRegistry, 335

INVALID\_ACCESS\_ERROR

OsclProcStatus, 462  
**INVALID\_ARGUMENT\_ERROR**  
 OsclProcStatus, 462  
**INVALID\_FUNCTION\_ERROR**  
 OsclProcStatus, 462  
**INVALID\_HANDLE\_ERROR**  
 OsclProcStatus, 462  
**INVALID\_ID**  
 BufFragStatusClass, 119  
**INVALID\_OPERATION\_ERROR**  
 OsclProcStatus, 462  
**INVALID\_PARAM\_ERROR**  
 OsclProcStatus, 461  
**INVALID\_POINTER\_ERROR**  
 OsclProcStatus, 462  
**INVALID\_PRIORITY\_ERROR**  
 OsclProcStatus, 461  
**INVALID\_THREAD\_ERROR**  
 OsclProcStatus, 461  
**INVALID\_THREAD\_ID\_ERROR**  
 OsclProcStatus, 461  
**INVALID\_TYPE**  
 Oscl\_FileFind, 184  
**iObserver**  
 OsclIPSocketI, 405  
 OsclMemPoolFixedChunkAllocator, 432  
 OsclMemPoolResizableAllocator, 439  
**iOffset**  
 OsclDoubleListBase, 360  
 OsclDoubleRunner, 361  
**iOpCount**  
 OsclFileStatsItem, 398  
**iOsclBase**  
 OsclSelect, 506  
**iOsclErrorTrap**  
 OsclSelect, 506  
**iOsclLogger**  
 OsclSelect, 506  
**iOsclMemory**  
 OsclSelect, 506  
**iOsclScheduler**  
 OsclSelect, 506  
**iOtherExecStats**  
 OsclExecSchedulerCommonBase, 390  
**iOutputFile**  
 OsclSelect, 506  
**iPacketLen**  
 RecvFromParam, 617  
**iPacketSource**  
 RecvFromParam, 617  
**ipAddr**  
 OsclNetworkAddress, 452  
**iParam**  
 OsclFileStatsItem, 398  
  
 OsclSocketRequest, 535  
 OsclSocketRequestAO, 539  
**iParam2**  
 OsclFileStatsItem, 398  
**iParamSize**  
 OsclSocketRequestAO, 539  
**iParentBuffer**  
 OsclMemPoolResizableAllocator::Mem-  
 PoolBlockInfo, 440  
**iPrev**  
 OsclDoubleLink, 357  
**iPrevFreeBlock**  
 OsclMemPoolResizableAllocator::Mem-  
 PoolBlockInfo, 440  
**iPriority**  
 OsclPriorityLink, 454  
**iPtr**  
 PVSockBufRecv, 613  
 PVSockBufSend, 614  
**iPVActiveStats**  
 PVActiveBase, 596  
**iPVReadyQLink**  
 PVActiveBase, 596  
**iPVStatQ**  
 OsclExecSchedulerCommonBase, 390  
**iPVStats**  
 OsclExecSchedulerCommonBase, 390  
**iQSize**  
 ListenParam, 136  
**iReadyQ**  
 OsclExecSchedulerCommonBase, 390  
**irear**  
 Oscl\_Queue\_Base, 232  
**iRefCount**  
 DNSRequestParam, 130  
 OsclMemPoolFixedChunkAllocator, 432  
 OsclMemPoolResizableAllocator, 439  
**iRequestedAvailableFreeMemSize**  
 OsclMemPoolResizableAllocator, 439  
**iRequestedNextAvailableSize**  
 OsclMemPoolResizableAllocator, 439  
**iResumeSem**  
 OsclExecSchedulerCommonBase, 390  
**is\_writable**  
 OSCL\_String, 253  
 OSCL\_wString, 297  
**is\_zero**  
 TimeValue, 633  
**IsActive**  
 PVLogger, 601  
**IsAdded**  
 PVActiveBase, 594  
**isAllocNodePtr**  
 MM\_AllocBlockHdr, 143

IsBusy  
   OsclActiveObject, 303  
   OsclTimerObject, 571  
 iSchedulerAlloc  
   OsclSelect, 506  
 iSchedulerName  
   OsclSelect, 506  
 iSchedulerReserve  
   OsclSelect, 506  
 isCIEquivalentTo  
   StrCSumPtrLen, 626  
   StrPtrLen, 629  
   WStrPtrLen, 639  
 isCIPrefixOf  
   StrPtrLen, 629  
 iSelect  
   OsclSocketServRequestQElem, 548  
 IsEmpty  
   OsclDoubleListBase, 360  
 iSeqNum  
   TReadyQueLink, 637  
 iServerError  
   OsclSocketServIBase, 545  
 iServerState  
   OsclSocketServIBase, 545  
 isFixed  
   OsclFileCacheBuffer, 395  
 IsHead  
   OsclDoubleList, 358  
   OsclPriorityList, 455  
 IsIn  
   OsclReadyQ, 471  
   OsclTimerQ, 574  
 IsInAnyQ  
   PVActiveBase, 595  
 IsInstalled  
   OsclExecSchedulerCommonBase, 387  
 IsInUse  
   OsclAsyncFileBuffer, 311  
 iSize  
   Oscl\_File::OsclFixedCacheParam, 183  
 isLetter  
   StrPtrLen, 629  
 IsLocalData  
   MediaData, 139  
 iSocket  
   OsclIPSocketI, 405  
 iSocketError  
   OsclDNSRequestAO, 356  
   OsclSocketRequestAO, 539  
 iSocketFxn  
   OsclSocketMethod, 533  
 iSocketI  
   OsclSocketRequest, 535  
 iSocketRequest  
   OsclSocketServRequestQElem, 548  
 iSocketRequestAO  
   OsclSocketMethod, 533  
   OsclSocketRequest, 535  
 iSocketServ  
   OsclDNSIBase, 347  
   OsclIPSocketI, 405  
   OsclSocketIBase, 530  
 isOpen  
   OsclSocketIBase, 528  
 IsReady  
   OsclDNSIBase, 347  
 IsSameThreadContext  
   PVThreadContext, 615  
 IsServConnected  
   OsclSocketServIBase, 545  
 IsServerThread  
   OsclSocketServI, 543  
 isSetFailure  
   MM\_Audit\_Imp, 150  
 IsStarted  
   OsclExecSchedulerCommonBase, 387  
 IsTail  
   OsclDoubleList, 358  
   OsclPriorityList, 455  
 iStartAddr  
   OsclMemPoolResizableAllocator::Mem-  
     PoolBufferInfo, 441  
 iStartTick  
   OsclFileStatsItem, 398  
 iStatus  
   PVActiveBase, 596  
 iStopper  
   OsclExecSchedulerCommonBase, 390  
 iStopperCrit  
   OsclExecSchedulerCommonBase, 390  
 IsUpdated  
   OsclFileCacheBuffer, 395  
 iSuspended  
   OsclExecSchedulerCommonBase, 390  
 IsValid  
   OsclAsyncFileBuffer, 311  
 iTAny  
   OsclTrapStackItem, 583  
 iterator  
   Oscl\_Linked\_List\_Base, 207  
   Oscl\_Map, 210  
   Oscl\_Rb\_Tree, 235  
   Oscl\_Rb\_Tree\_Iterator, 242  
   Oscl\_TagTree::iterator, 268  
   Oscl\_Vector, 277  
   OsclPriorityQueue, 457  
 iThreadContext

OsclExecSchedulerCommonBase, 390  
 PVActiveBase, 596  
**iTime**  
 OsclExecSchedulerCommonBase, 390  
**iTimeCompareThreshold**  
 OsclExecSchedulerCommonBase, 390  
**iTimeQueuedTicks**  
 TReadyQueLink, 637  
**iTimeToRunTicks**  
 TReadyQueLink, 637  
**iTotalPercent**  
 OsclExecSchedulerCommonBase, 390  
**iTotalTicks**  
 OsclFileStatsItem, 398  
**iTotalTicksTemp**  
 OsclExecSchedulerCommonBase, 390  
**iTrapOperation**  
 OsclTrapStackItem, 583  
**iTrapStack**  
 OsclErrorTrapImp, 369  
**iVec**  
 OsclComponentRegistryData, 336  
**iXferLen**  
 SendParam, 620  
 SendToParam, 621  
**Join**  
 OsclIPSocketI, 404  
 OsclSocketI, 522  
 OsclSocketIBase, 528  
 OsclUDPSocket, 586  
**Jump**  
 OsclJump, 406  
**key\_comp**  
 Oscl\_Map, 213  
**key\_compare**  
 Oscl\_Map, 210  
**key\_type**  
 Oscl\_Map, 210  
 Oscl\_Rb\_Tree, 235  
**largeasyncfilereadwrite\_test**  
 Oscl\_File, 181  
**Leave**  
 OsclError, 363  
**LeaveIfError**  
 OsclError, 363  
**LeaveIfNull**  
 OsclError, 363  
**Left**  
 OsclPtrC, 466  
**left**  
 Oscl\_Rb\_Tree\_Node\_Base, 246  
**len**  
 OsclMemoryFragment, 427  
 StrPtrLen, 629  
 WStrPtrLen, 639  
**Length**  
 OsclAsyncFileBuffer, 311  
 OsclBuf, 332  
 OsclPtr, 463  
 OsclPtrC, 466  
**length**  
 BuffFragGroup, 118  
 OsclBinStream, 330  
 StrPtrLen, 629  
 WStrPtrLen, 639  
**lineNo**  
 MM\_AllocInfo, 145  
 MM\_AllocQueryInfo, 147  
**link\_type**  
 Oscl\_Rb\_Tree, 235  
 Oscl\_Rb\_Tree\_Const\_Iterator, 239  
 Oscl\_Rb\_Tree\_Iterator, 242  
 Oscl\_Rb\_Tree\_Node, 244  
**LinkedListElement**, 135  
 LinkedListElement, 135  
**LinkedListElement**  
 data, 135  
 LinkedListElement, 135  
 next, 135  
**Listen**  
 OsclListenMethod, 407  
 OsclListenRequest, 408  
 OsclSocketI, 522  
 OsclSocketIBase, 528  
 OsclTCPSocket, 553  
 OsclTCPSocketI, 556  
**ListenAsync**  
 OsclSocketIBase, 528  
 OsclTCPSocket, 553  
 OsclTCPSocketI, 557  
**ListenParam**, 136  
 ListenParam, 136  
**ListenParam**  
 iQSize, 136  
 ListenParam, 136  
**ListenRequest**  
 OsclListenMethod, 407  
**little\_endian\_to\_host**  
 oscibase, 34  
**localbuf**  
 MediaData, 139  
**Lock**  
 OsclLockBase, 409  
 OsclMutex, 446  
 OsclNullLock, 453

OsclThreadLock, [562](#)  
 lockAndGetInstance  
   OsclSingletonRegistry, [520](#)  
 Log  
   OsclFileStats, [397](#)  
 log\_level\_type  
   AllPassFilter, [110](#)  
   PVLogger, [599](#)  
   PVLoggerFilter, [605](#)  
   PVLoggerRegistry, [609](#)  
 LogAll  
   OsclFileStats, [397](#)  
 Logger  
   OsclSocketI, [522](#)  
 LogMsgBuffers  
   PVLogger, [601](#)  
 LogMsgBuffersV  
   PVLogger, [601](#)  
 LogMsgString  
   PVLogger, [602](#)  
 LogMsgStringV  
   PVLogger, [602](#)  
 LoopbackSocket  
   OsclSocketServI, [543](#)  
 lower\_bound  
   Oscl\_Map, [213](#)  
   Oscl\_Rb\_Tree, [235](#)  
 MakeAddr  
   OsclSocketI, [523](#)  
 makeValidTag  
   MM\_Audit\_Imp, [150](#)  
 map\_type  
   Oscl\_TagTree, [261](#)  
 mapit  
   Oscl\_TagTree::const\_iterator, [265](#)  
   Oscl\_TagTree::iterator, [268](#)  
 mapiter  
   Oscl\_TagTree::const\_iterator, [265](#)  
   Oscl\_TagTree::iterator, [268](#)  
 Match  
   OsclComponentRegistryElement, [337](#)  
 max\_size  
   Oscl\_Map, [213](#)  
   Oscl\_Rb\_Tree, [235](#)  
 MAX\_THRDS\_REACHED\_ERROR  
   OsclProcStatus, [461](#)  
 maximum  
   Oscl\_Rb\_Tree\_Node\_Base, [246](#)  
 MaxLen  
   OsclNameString, [447](#)  
 maxsize  
   CFastRep, [124](#)  
   CHheapRep, [126](#)  
 CStackRep, [128](#)  
 mbchar  
   osclbase, [32](#)  
 MediaData, [137](#)  
   MediaData, [138](#)  
 MediaData  
   ~MediaData, [138](#)  
   AddLocalFragment, [138](#)  
   available\_localbuf, [139](#)  
   Clear, [138](#)  
   GetAvailableBufferSize, [138](#)  
   GetLocalBufsize, [139](#)  
   GetLocalFragment, [139](#)  
   GetMediaFragment, [139](#)  
   GetMediaSize, [139](#)  
   GetNumMediaFrags, [139](#)  
   GetTimestamp, [139](#)  
   IsLocalData, [139](#)  
   localbuf, [139](#)  
   MediaData, [138](#)  
   num\_reserved\_fragments, [139](#)  
   SetTimestamp, [139](#)  
   timestamp, [139](#)  
 MediaStatusClass, [140](#)  
 MediaTimestamp  
   osclutil, [66](#)  
 MEM\_ALIGN\_SIZE  
   osclmemory, [49](#)  
 MemAllocator, [141](#)  
 MemAllocator  
   ~MemAllocator, [141](#)  
   allocate, [141](#)  
   deallocate, [141](#)  
   pointer, [141](#)  
 memoryPoolBufferMgmtOverhead  
   OsclMemPoolResizableAllocator, [437](#)  
 message\_id\_type  
   AllPassFilter, [110](#)  
   PVLogger, [599](#)  
   PVLoggerAppender, [604](#)  
   PVLoggerFilter, [605](#)  
   PVLoggerLayout, [607](#)  
 MethodDone  
   OsclDNSMethod, [350](#)  
   OsclSocketMethod, [532](#)  
 MICROSECONDS  
   osclbase, [33](#)  
 MILLISECONDS  
   osclbase, [33](#)  
 MIN\_FENCE\_SIZE  
   osclmemory, [49](#)  
 minimum  
   Oscl\_Rb\_Tree\_Node\_Base, [246](#)  
 MM\_AddTag

---

MM\_Audit\_Imp, 150  
 OsclMemAudit, 415  
**MM\_ALLOC\_MAX\_QUERY\_FILENAME\_-LEN**  
 osclmemory, 49  
**MM\_ALLOC\_MAX\_QUERY\_TAG\_LEN**  
 osclmemory, 49  
**MM\_allocate**  
 MM\_Audit\_Imp, 150  
 OsclMemAudit, 415  
**MM\_AllocBlockFence, 142**  
 MM\_AllocBlockFence, 142  
**MM\_AllocBlockFence**  
 check\_fence, 142  
 fill\_fence, 142  
 MM\_AllocBlockFence, 142  
 pad, 142  
**MM\_AllocBlockHdr, 143**  
 MM\_AllocBlockHdr, 143  
**MM\_AllocBlockHdr**  
 isAllocNodePtr, 143  
 MM\_AllocBlockHdr, 143  
 pad, 143  
 pNode, 143  
 pRootNode, 143  
 setAllocNodeFlag, 143  
 size, 143  
**MM\_AllocInfo, 144**  
 MM\_AllocInfo, 145  
**MM\_AllocInfo**  
 ~MM\_AllocInfo, 145  
 allocNum, 145  
 bSetFailure, 145  
 lineNo, 145  
 MM\_AllocInfo, 145  
 operator delete, 145  
 operator new, 145  
 pFileName, 145  
 pMemBlock, 145  
 pStatsNode, 145  
 size, 145  
**MM\_AllocNode, 146**  
 MM\_AllocNode, 146  
**MM\_AllocNode**  
 ~MM\_AllocNode, 146  
 MM\_AllocNode, 146  
 operator delete, 146  
 operator new, 146  
 pAllocInfo, 146  
 pNext, 146  
 pPrev, 146  
**MM\_AllocNodeAutoPtr**  
 osclmemory, 56  
**MM\_AllocQueryInfo, 147**  
**MM\_AllocQueryInfo**  
 allocNum, 147  
 fileName, 147  
 lineNo, 147  
 pMemBlock, 147  
 size, 147  
 tag, 147  
**MM\_AUDIT\_ALLOC\_NODE\_ENABLE\_-FLAG**  
 osclmemory, 49  
**MM\_AUDIT\_ALLOC\_NODE\_SUPPORT**  
 osclmemory, 49  
**MM\_AUDIT\_FAILURE\_SIMULATION\_-SUPPORT**  
 osclmemory, 49  
**MM\_AUDIT\_FENCE\_SUPPORT**  
 osclmemory, 49  
**MM\_AUDIT\_FILL\_SUPPORT**  
 osclmemory, 49  
**MM\_Audit\_Imp, 148**  
 ~MM\_Audit\_Imp, 149  
 addAllocNode, 149  
 createStatsNode, 149  
 getAuditRoot, 149  
 getSize, 149  
 getTagActualSize, 149  
 isSetFailure, 150  
 makeValidTag, 150  
**MM\_AddTag, 150**  
**MM\_allocate, 150**  
**MM\_Audit\_Imp, 149**  
**MM\_CreateAllocNodeInfo, 150**  
**MM\_deallocate, 150**  
**MM\_GetAllocNo, 150**  
**MM\_GetAllocNodeInfo, 151**  
**MM\_GetExistingTag, 151**  
**MM\_GetMode, 151**  
**MM\_GetNumAllocNodes, 151**  
**MM\_GetOverheadStats, 151**  
**MM\_GetPostfillPattern, 151**  
**MM\_GetPrefillPattern, 151**  
**MM\_GetRootNode, 152**  
**MM\_GetStats, 152**  
**MM\_GetStatsInDepth, 152**  
**MM\_GetTagName, 152**  
**MM\_GetTreeNodes, 152**  
**MM\_ReleaseAllocNodeInfo, 152**  
**MM\_SetFailurePoint, 153**  
**MM\_SetMode, 153**  
**MM\_SetPostfillPattern, 153**  
**MM\_SetPrefillPattern, 153**  
**MM\_SetTagLevel, 153**  
**MM\_UnsetFailurePoint, 153**  
**MM\_Validate, 153**

pruneSubtree, 154  
 removeALLAllocNodes, 154  
 removeAllocNode, 154  
 retrieveParentTag, 154  
 retrieveParentTagLength, 154  
 updateStatsNode, 154  
 updateStatsNodeInFailure, 154  
 validate, 154  
 validate\_all\_heap, 154  
**MM\_AUDIT\_INCLUDE\_ALL\_HEAP\_-VALIDATION**  
 osclmemory, 49  
**MM\_AUDIT\_POSTFILL\_FLAG**  
 osclmemory, 49  
**MM\_AUDIT\_PREFILL\_FLAG**  
 osclmemory, 49  
**MM\_AUDIT\_SUPPRESS\_FILENAME\_FLAG**  
 osclmemory, 49  
**MM\_AUDIT\_VALIDATE\_ALL\_HEAP\_FLAG**  
 osclmemory, 49  
**MM\_AUDIT\_VALIDATE\_BLOCK**  
 osclmemory, 49  
**MM\_AUDIT\_VALIDATE\_ON\_FREE\_FLAG**  
 osclmemory, 49  
**MM\_AuditOverheadStats**, 156  
**MM\_AuditOverheadStats**  
 per\_allocation\_overhead, 156  
 stats\_overhead, 156  
**MM\_CreateAllocNodeInfo**  
 MM\_Audit\_Imp, 150  
 OsclMemAudit, 415  
**MM\_deallocate**  
 MM\_Audit\_Imp, 150  
 OsclMemAudit, 415  
**MM\_FailInsertParam**, 157  
 MM\_FailInsertParam, 157  
**MM\_FailInsertParam**  
 MM\_FailInsertParam, 157  
 nAllocNum, 157  
 operator delete, 157  
 operator new, 157  
 reset, 157  
 xsubi, 157  
**MM\_GetAllocNo**  
 MM\_Audit\_Imp, 150  
 OsclMemAudit, 415  
**MM\_GetAllocNodeInfo**  
 MM\_Audit\_Imp, 151  
 OsclMemAudit, 415  
**MM\_GetExistingTag**  
 MM\_Audit\_Imp, 151  
 OsclMemAudit, 416  
**MM\_GetMode**  
 MM\_Audit\_Imp, 151  
 OsclMemAudit, 416  
**MM\_GetNumAllocNodes**  
 MM\_Audit\_Imp, 151  
 OsclMemAudit, 416  
**MM\_GetOverheadStats**  
 MM\_Audit\_Imp, 151  
 OsclMemAudit, 416  
**MM\_GetPostfillPattern**  
 MM\_Audit\_Imp, 151  
 OsclMemAudit, 416  
**MM\_GetPrefillPattern**  
 MM\_Audit\_Imp, 151  
 OsclMemAudit, 416  
**MM\_GetRefCount**  
 OsclMemAudit, 416  
**MM\_GetRootNode**  
 MM\_Audit\_Imp, 152  
 OsclMemAudit, 417  
**MM\_GetStats**  
 MM\_Audit\_Imp, 152  
 OsclMemAudit, 417  
**MM\_GetStatsInDepth**  
 MM\_Audit\_Imp, 152  
 OsclMemAudit, 417  
**MM\_GetTagName**  
 MM\_Audit\_Imp, 152  
 OsclMemAudit, 417  
**MM\_GetTreeNodes**  
 MM\_Audit\_Imp, 152  
 OsclMemAudit, 417  
**MM\_ReleaseAllocNodeInfo**  
 MM\_Audit\_Imp, 152  
 OsclMemAudit, 417  
**MM\_SetFailurePoint**  
 MM\_Audit\_Imp, 153  
 OsclMemAudit, 417  
**MM\_SetMode**  
 MM\_Audit\_Imp, 153  
 OsclMemAudit, 418  
**MM\_SetPostfillPattern**  
 MM\_Audit\_Imp, 153  
 OsclMemAudit, 418  
**MM\_SetPrefillPattern**  
 MM\_Audit\_Imp, 153  
 OsclMemAudit, 418  
**MM\_SetTagLevel**  
 MM\_Audit\_Imp, 153  
 OsclMemAudit, 418  
**MM\_Stats\_CB**, 158  
 MM\_Stats\_CB, 158  
 num\_child\_nodes, 158  
 operator delete, 158  
 operator new, 158  
 pStats, 158

tag, 158  
**MM\_Stats\_t**, 159  
 MM\_Stats\_t, 160  
 numAllocFails, 160  
 numAllocs, 160  
 numBytes, 160  
 operator delete, 160  
 operator new, 160  
 peakNumAllocs, 160  
 peakNumBytes, 160  
 reset, 160  
 totalNumAllocs, 160  
 totalNumBytes, 160  
 update, 160  
**MM\_StatsNodeTagTreeType**  
 osclmemory, 56  
**MM\_UnsetFailurePoint**  
 MM\_Audit\_Imp, 153  
 OsclMemAudit, 418  
**MM\_Validate**  
 MM\_Audit\_Imp, 153  
 OsclMemAudit, 418  
**MMAuditCharAutoPtr**  
 osclmemory, 56  
**MMAuditUint8AutoPtr**  
 osclmemory, 56  
**Mode**  
 OsclNativeFile, 449  
**mode**  
 oscl\_stat\_buf, 250  
**MODE\_APPEND**  
 Oscl\_File, 175  
**MODE\_BINARY**  
 Oscl\_File, 175  
**MODE\_READ**  
 Oscl\_File, 175  
**MODE\_READ\_PLUS**  
 Oscl\_File, 175  
**MODE\_READWRITE**  
 Oscl\_File, 175  
**MODE\_TEXT**  
 Oscl\_File, 175  
**mode\_type**  
 Oscl\_File, 175  
**move\_to\_end**  
 Oscl\_Linked\_List, 202  
 Oscl\_Linked\_List\_Base, 206  
 Oscl\_MTLinkedList, 218  
**move\_to\_front**  
 Oscl\_Linked\_List, 202  
 Oscl\_Linked\_List\_Base, 206  
 Oscl\_MTLinkedList, 219  
**MSEC\_PER\_SEC**  
 osclbase, 43  
**MSEC\_TO\_MICROSEC**  
 oscl\_socket\_method.h, 740  
**MsecToTicks**  
 OsclTickCount, 563  
**MUTEX\_LOCKED\_ERROR**  
 OsclProcStatus, 462  
**nAllocNum**  
 MM\_FailInsertParam, 157  
**New**  
 Oscl\_DefAllocWithRefCounter, 169  
**NewL**  
 OsclAcceptMethod, 299  
 OsclAsyncFile, 308  
 OsclAsyncFileBuffer, 311  
 OsclBindMethod, 313  
 OsclBuf, 332  
 OsclConnectMethod, 339  
 OsclDNS, 343  
 OsclDNSI, 345  
 OsclGetHostByNameMethod, 399  
 OsclListenMethod, 407  
 OsclRecvFromMethod, 472  
 OsclRecvMethod, 476  
 OsclSendMethod, 509  
 OsclSendToMethod, 511  
 OsclShutdownMethod, 516  
 OsclSocketI, 523  
 OsclSocketServ, 541  
 OsclSocketServI, 543  
 OsclTCPSocket, 553  
 OsclTCPSocketI, 557  
 OsclUDPSocket, 586  
 OsclUDPSocketI, 590  
**NewRequest**  
 OsclDNSRequestAO, 355  
 OsclSocketRequestAO, 538  
**next**  
 BuffFragGroup, 118  
 LinkedListElement, 135  
**nextFragPtr**  
 OsclBinStream, 330  
**NO\_PERMISSION\_ERROR**  
 OsclProcStatus, 461  
**Node**  
 Oscl\_TagTree::Node, 271  
**node**  
 Oscl\_Rb\_Tree\_Const\_Iterator, 239  
 Oscl\_Rb\_Tree\_Iterator, 242  
**node\_ptr**  
 Oscl\_TagTree, 261  
**node\_type**  
 Oscl\_TagTree, 261  
**NOT\_ENOUGH\_MEMORY\_ERROR**

OsclProcStatus, 461  
 NOT\_ENOUGH\_RESOURCES\_ERROR  
   OsclProcStatus, 461  
 NOT\_ENOUGH\_SPACE  
   BufFragStatusClass, 119  
 NOT\_IMPLEMENTED  
   OsclProcStatus, 462  
 NOT\_SUSPENDED\_ERROR  
   OsclProcStatus, 461  
 notifyfreeblockavailable  
   OsclMemPoolResizableAllocator, 437  
 notifyfreechunkavailable  
   OsclMemPoolFixedChunkAllocator, 431  
 notifyfreememoryavailable  
   OsclMemPoolResizableAllocator, 437  
 NTPTime, 161  
   get\_lower32, 163  
   get\_middle32, 163  
   get\_upper32, 163  
   get\_value, 163  
 NTPTime, 162, 163  
 operator+=, 163  
 operator-, 163  
 operator=, 163, 164  
 set\_from\_system\_time, 164  
 set\_to\_current\_time, 164  
 TimeValue, 635  
 to\_system\_time, 164  
 NULL  
   osclbase, 30  
 NULL\_INPUT  
   BufFragStatusClass, 119  
 NULL\_TERM\_CHAR  
   osclbase, 30  
 num\_child\_nodes  
   MM\_Stats\_CB, 158  
 num\_elements  
   Oscl\_Linked\_List\_Base, 207  
 num\_fragments  
   BufFragGroup, 118  
 num\_reserved.fragments  
   MediaData, 139  
 numAllocFails  
   MM\_Stats\_t, 160  
 numAllocs  
   MM\_Stats\_t, 160  
 numBytes  
   MM\_Stats\_t, 160  
 numelems  
   Oscl\_Queue\_Base, 232  
   Oscl\_Vector\_Base, 284  
 numFrags  
   OsclBinStream, 330  
 octet  
   osclbase, 32  
 Offset  
   OsclAsyncFileBuffer, 311  
 Open  
   Oscl\_File, 177  
   OsclAsyncFile, 308, 309  
   OsclDNSI, 345  
   OsclDNSIBase, 347  
   OsclFileCache, 393  
   OsclNativeFile, 449  
   OsclSocketI, 523  
   OsclSocketIBase, 529  
   OsclSocketServRequestList, 546  
 OpenSession  
   OsclComponentRegistry, 335  
 operator \*  
   Oscl\_Rb\_Tree\_Const\_Iterator, 239  
   Oscl\_Rb\_Tree\_Iterator, 242  
   Oscl\_TagTree::const\_iterator, 265  
   Oscl\_TagTree::iterator, 268  
   OsclExclusiveArrayPtr, 372  
   OsclExclusivePtr, 375  
   OsclExclusivePtrA, 378  
   OSCLMemAutoPtr, 422  
   OsclSharedPtr, 514  
   OsclSingleton, 518  
   OsclTLS, 575  
   OsclTLSEx, 577  
 operator \*=  
   TimeValue, 634  
 operator delete  
   MM\_AllocInfo, 145  
   MM\_AllocNode, 146  
   MM\_FailInsertParam, 157  
   MM\_Stats\_CB, 158  
   MM\_Stats\_t, 160  
   oscl\_mem.h, 694  
   OsclErrorAllocator, 366  
   osclmemory, 57  
   OsclMemStatsNode, 444  
 operator delete[]  
   osclmemory, 57  
 operator new  
   MM\_AllocInfo, 145  
   MM\_AllocNode, 146  
   MM\_FailInsertParam, 157  
   MM\_Stats\_CB, 158  
   MM\_Stats\_t, 160  
   oscl\_mem.h, 694  
   osclconfig\_global\_placement\_new.h, 790  
   OsclErrorAllocator, 366  
   osclmemory, 57  
   OsclMemStatsNode, 444

operator new[]  
     osclmemory, 57  
 operator T \*  
     OsclDoubleRunner, 361  
 operator TheClass \*  
     OsclSharedPtr, 515  
 operator!=  
     Oscl\_Rb\_Tree\_Const\_Iterator, 239  
     Oscl\_Rb\_Tree\_Iterator, 242  
     OSCL\_String, 253  
     Oscl\_TagTree::const\_iterator, 265  
     Oscl\_TagTree::iterator, 268  
     OSCL\_wString, 297  
     OsclAOStatus, 306  
     OsclUuid, 592  
     StrCSumPtrLen, 626  
     StrPtrLen, 629  
     TimeValue, 635  
     WStrPtrLen, 639  
 operator()  
     Oscl\_Less, 199  
     Oscl\_Map::value\_compare, 215  
     Oscl\_Select1st, 247  
     Oscl\_Tag\_Base, 259  
 operator++  
     Oscl\_Rb\_Tree\_Const\_Iterator, 239  
     Oscl\_Rb\_Tree\_Iterator, 242  
     Oscl\_TagTree::const\_iterator, 265  
     Oscl\_TagTree::iterator, 268  
     OsclDoubleRunner, 361  
 operator+=  
     NTPTime, 163  
     OSCL\_String, 253  
     OSCL\_wString, 297  
     TimeValue, 634  
 operator-  
     NTPTime, 163  
     osclbase, 34  
 operator-  
     Oscl\_Rb\_Tree\_Const\_Iterator, 239  
     Oscl\_Rb\_Tree\_Iterator, 242  
     Oscl\_TagTree::const\_iterator, 265  
     Oscl\_TagTree::iterator, 268  
     OsclDoubleRunner, 361  
 operator-=  
     TimeValue, 634  
 operator->  
     Oscl\_Rb\_Tree\_Const\_Iterator, 239  
     Oscl\_Rb\_Tree\_Iterator, 242  
     Oscl\_TagTree::const\_iterator, 265  
     Oscl\_TagTree::iterator, 268  
     OsclExclusiveArrayPtr, 372  
     OsclExclusivePtr, 375  
     OsclExclusivePtrA, 378  
     OsclMemAutoPtr, 422  
     OsclRefCounterMemFrag, 483  
     OsclSharedPtr, 515  
     osclutil, 69, 70  
     OsclUuid, 592  
     StrCSumPtrLen, 626  
     StrPtrLen, 629  
     TimeValue, 634  
     WStrPtrLen, 639  
 operator==  
     Oscl\_Rb\_Tree\_Const\_Iterator, 239  
     Oscl\_Rb\_Tree\_Iterator, 242  
     OSCL\_String, 254  
     Oscl\_TagTree::const\_iterator, 265  
     Oscl\_TagTree::iterator, 268  
     OSCL\_wString, 297  
     OsclAOStatus, 306  
     osclbase, 34

OsclNetworkAddress, 452  
 OsclUuid, 592  
 StrCSumPtrLen, 626  
 StrPtrLen, 629  
 TimeValue, 635  
 WStrPtrLen, 639  
 operator>  
   OSCL\_String, 254  
   OSCL\_wString, 297  
   OsclAOStatus, 306  
   TimeValue, 635  
 operator>=  
   OSCL\_String, 254  
   OSCL\_wString, 297  
   OsclAOStatus, 306  
   TimeValue, 635  
 operator>>  
   OsclBinIStreamBigEndian, 318  
   OsclBinIStreamLittleEndian, 321  
 operator[]  
   Oscl\_Map, 213  
   OSCL\_String, 254  
   Oscl\_TagTree, 263  
   Oscl\_Vector, 279  
   OSCL\_wString, 297  
 OSCL Base, 23  
 OSCL config, 19  
 OSCL Error, 81  
 OSCL Init, 103  
 OSCL IO, 91  
 OSCL Memory, 44  
 OSCL Proc, 99  
 OSCL Util, 60  
 OSCL\_ABS  
   osclbase, 30  
 oscl\_abs  
   osclutil, 70  
 OSCL\_AF\_INET  
   osclconfig\_io.h, 793  
 Oscl\_Alloc, 165  
   allocate, 165  
   allocate\_fl, 165  
 OSCL\_ALLOC\_DELETE  
   osclmemory, 49  
 OSCL\_ALLOC\_NEW  
   osclmemory, 50  
 oscl\_aostatus.h, 640  
 OSCL\_ARRAY\_DELETE  
   osclmemory, 50  
 OSCL\_ARRAY\_NEW  
   osclmemory, 50  
 OSCL\_ASCII\_CASE\_MAGIC\_BIT  
   osclutil, 80  
 oscl\_asin  
 osclutil, 70  
 OSCL\_ASSERT  
   osclbase, 30  
 OSCL\_Assert  
   osclbase, 34  
 oscl\_assert.h, 641  
 OSCL\_ASSERT\_ALWAYS  
   osclconfig, 20  
 oscl\_atan  
   osclutil, 70  
 OSCL\_AUDIT\_ARRAY\_NEW  
   osclmemory, 50  
 OSCL\_AUDIT\_CALLOC  
   osclmemory, 51  
 OSCL\_AUDIT\_MALLOC  
   osclmemory, 51  
 OSCL\_AUDIT\_NEW  
   osclmemory, 51  
 OSCL\_AUDIT\_REALLOC  
   osclmemory, 52  
 OSCL\_BAD\_ALLOC\_EXCEPTION\_CODE  
   osclerror, 84  
 oscl\_base.h, 642  
 oscl\_base\_alloc.h, 643  
 oscl\_base\_macros.h, 644  
 OSCL\_BEGIN\_PACKED  
   osclbase, 30  
   osclconfig.h, 783  
 oscl\_bin\_stream.h, 645  
 OSCL\_BYPASS\_MEMMGT  
   osclconfig\_memory.h, 803  
 oscl\_byte\_order.h, 646  
 OSCL\_BYTE\_ORDER\_BIG\_ENDIAN  
   osclconfig, 20  
 OSCL\_BYTE\_ORDER\_LITTLE\_ENDIAN  
   osclconfig, 20  
 OSCL\_CALLOC  
   osclmemory, 52  
 oscl\_malloc  
   osclmemory, 52  
 OSCL\_CATCH  
   osclerror, 84  
 OSCL\_CATCH\_ANY  
   osclerror, 85  
 OSCL\_CHAR\_IS\_SIGNED  
   osclconfig\_limits\_typedefs.h, 802  
 OSCL\_CHAR\_IS\_UNSIGNED  
   osclconfig\_limits\_typedefs.h, 802  
 oscl\_chdir  
   osclio, 95  
 oscl\_Clstrcmp  
   osclbase, 34, 35  
 oscl\_Clstrncmp  
   osclbase, 35

**OSCL\_CLEANUP\_BASE\_CLASS**  
 osclmemory, 52

**OSCL\_CLOCK\_HAS\_DRIFT\_CORRECTION**  
 osclconfig\_util.h, 823

**OSCL\_COND\_EXPORT\_REF**  
 osclbase, 30

**OSCL\_COND\_IMPORT\_REF**  
 osclbase, 30

**OSCL\_CONST\_CAST**  
 osclbase, 30

**oscl\_cos**  
 osclutil, 70

**Oscl\_Dealloc**, 166  
 deallocate, 166

**Oscl\_DefAlloc**, 167

**Oscl\_DefAlloc**  
 allocate, 167  
 allocate\_fl, 167  
 deallocate, 167

**oscl\_defalloc.h**, 647

**Oscl\_DefAllocWithRefCounter**, 168

**Oscl\_DefAllocWithRefCounter**  
 addRef, 168  
 Delete, 168  
 getCount, 168  
 New, 169  
 removeRef, 169

**OSCL\_DEFAULT\_FREE**  
 osclmemory, 53

**OSCL\_DEFAULT\_MALLOC**  
 osclmemory, 53

**OSCL\_DELETE**  
 osclmemory, 53

**Oscl\_DeleteFile**  
 Oscl\_FileServer, 188, 189

**OSCL\_DISABLE\_INLINES**  
 osclconfig\_unix\_android.h, 818  
 osclconfig\_unix\_common.h, 822

**OSCL\_DISABLE\_WARNING\_RETURN\_-  
TYPE\_NOT\_UDT**  
 osclbase, 30  
 osclmemory, 53

**OSCL\_DISABLE\_WARNING\_TRUNCATE\_-  
DEBUG\_MESSAGE**  
 oscl\_map.h, 688  
 oscl\_mem.h, 694  
 oscl\_mem\_audit.h, 697  
 oscl\_mem\_audit\_internals.h, 698  
 oscl\_mem\_auto\_ptr.h, 699  
 oscl\_tagtree.h, 767  
 oscl\_tree.h, 776  
 osclbase, 30  
 osclmemory, 53

**oscl\_dll.h**, 648

**OSCL\_DLL\_ENTRY\_POINT**  
 osclbase, 30

**OSCL\_DLL\_ENTRY\_POINT\_DEFAULT**  
 osclbase, 31

**oscl\_dns.h**, 649

**oscl\_dns\_gethostbyname.h**, 650

**oscl\_dns\_imp.h**, 651

**oscl\_dns\_imp\_base.h**, 652

**oscl\_dns\_imp\_pv.h**, 653

**oscl\_dns\_method.h**, 654

**oscl\_dns\_param.h**, 655  
 TDNSRequestParamAllocator, 655

**oscl\_dns\_request.h**, 656

**oscl\_dns\_tuneables.h**, 657  
 PV\_DNS\_IS\_THREAD, 657  
 PV\_DNS\_SERVER, 657

**oscl\_double\_list.h**, 658

**OSCL\_DYNAMIC\_CAST**  
 osclbase, 31

**OSCL\_END\_PACKED**  
 osclbase, 31  
 osclconfig.h, 783

**OSCL\_ERR\_NONE**  
 osclerror, 85

**oscl\_errno.h**, 659

**oscl\_error.h**, 660

**oscl\_error\_allocator.h**, 661

**oscl\_error\_codes.h**, 662

**oscl\_error\_imp.h**, 663

**oscl\_error\_imp\_cppexceptions.h**, 664

**oscl\_error\_imp\_fatalerror.h**, 665  
 \_PV\_TRAP, 665  
 \_PV\_TRAP\_NO\_TLS, 665  
 PVError\_DoLeave, 665

**oscl\_error\_imp\_jumps.h**, 666  
 \_PV\_TRAP, 666  
 \_PV\_TRAP\_NO\_TLS, 666  
 PVError\_DoLeave, 667

**oscl\_error\_trapcleanup.h**, 668

**oscl\_exception.h**, 669

**OSCL\_EXCEPTSET\_FLAG**  
 oscl\_socket\_serv\_imp\_pv.h, 748

**oscl\_exclusive\_ptr.h**, 670

**oscl\_exp**  
 osclutil, 71

**OSCL\_FastString**, 170  
 OSCL\_FastString, 171

**OSCL\_FastString**  
 ~OSCL\_FastString, 171  
 chartype, 171  
 get\_cstr, 171  
 get\_maxsize, 171  
 get\_size, 172  
 get\_str, 172

operator=, 172  
 OSCL\_FastString, 171  
 OSCL\_String, 172  
 set, 172  
 set\_length, 172  
**Oscl\_File**  
   ESymbianAccessMode\_Rfile, 175  
   ESymbianAccessMode\_RfileBuf, 175  
   MODE\_APPEND, 175  
   MODE\_BINARY, 175  
   MODE\_READ, 175  
   MODE\_READ\_PLUS, 175  
   MODE\_READWRITE, 175  
   MODE\_TEXT, 175  
   SEEKCUR, 175  
   SEEKEND, 175  
   SEEKSET, 175  
**Oscl\_File**, 174  
   ~Oscl\_File, 176  
   AddFixedCache, 176  
   asyncfilereadcancel\_test, 181  
   asyncfilereadwrite\_test, 181  
   Close, 176  
   EndOfFile, 176  
   Flush, 176  
   GetError, 177  
   Handle, 177  
   largeasyncfilereadwrite\_test, 181  
   mode\_type, 175  
   Open, 177  
   Oscl\_File, 175, 176  
   Oscl\_FileServer, 189  
   OsclFileCache, 181  
   OsclFileCacheBuffer, 181  
   OsclFileHandle, 396  
   Read, 178  
   RemoveFixedCache, 178  
   Seek, 178  
   seek\_type, 175  
   SetAsyncReadBufferSize, 178  
   SetCacheObserver, 178  
   SetFileHandle, 179  
   SetLoggingEnable, 179  
   SetNativeAccessMode, 179  
   SetNativeBufferSize, 179  
   SetPVCacheSize, 180  
   SetSummaryStatsLoggingEnable, 180  
   Size, 180  
   Tell, 180  
   TSymbianAccessMode, 175  
   Write, 180  
**Oscl\_File::OsclCacheObserver**, 182  
**Oscl\_File::OsclCacheObserver**  
   ChooseCurCache, 182  
  
 Oscl\_File::OsclFixedCacheParam, 183  
 Oscl\_File::OsclFixedCacheParam  
   Contains, 183  
   iFilePosition, 183  
   iSize, 183  
 oscl\_file\_async\_read.h, 671  
 OSCL\_FILE\_BUFFER\_MAX\_SIZE  
   osclconfig\_io.h, 793  
 oscl\_file\_cache.h, 672  
 OSCL\_FILE\_CHAR\_PATH\_DELIMITER  
   osclo, 93  
 oscl\_file\_dir\_utils.h, 673  
 oscl\_file\_find.h, 675  
 oscl\_file\_handle.h, 676  
 oscl\_file\_io.h, 677  
 oscl\_file\_native.h, 678  
 oscl\_file\_server.h, 679  
 oscl\_file\_stats.h, 680  
 OSCL\_FILE\_STATS\_LOGGER\_NODE  
   osclo, 93  
 oscl\_file\_types.h, 681  
 OSCL\_FILE\_WCHAR\_PATH\_DELIMITER  
   osclo, 93  
**Oscl\_FileFind**  
   DIR\_TYPE, 184  
   E\_BUFFER\_TOO\_SMALL, 185  
   E\_INVALID\_ARG, 184  
   E\_INVALID\_STATE, 184  
   E\_MEMORY\_ERROR, 185  
   E\_NO\_MATCH, 185  
   E\_NOT\_IMPLEMENTED, 185  
   E\_OK, 184  
   E\_OTHER, 185  
   E\_PATH\_NOT\_FOUND, 184  
   E\_PATH\_TOO\_LONG, 184  
   FILE\_TYPE, 184  
   INVALID\_TYPE, 184  
**Oscl\_FileFind**, 184  
   Oscl\_FileFind, 185  
**Oscl\_FileFind**  
   ~Oscl\_FileFind, 185  
   Close, 185  
   element\_type, 184  
   error\_type, 184  
   FindFirst, 185  
   FindNext, 186  
   GetElementType, 186  
   GetLastError, 186  
   Oscl\_FileFind, 185  
 OSCL\_FILEMGMT\_E\_ALREADY\_EXISTS  
   osclo, 94  
 OSCL\_FILEMGMT\_E\_NO\_MATCH  
   osclo, 94  
 OSCL\_FILEMGMT\_E\_NOT\_EMPTY

osclio, 94  
**OSCL\_FILEMGMT\_E\_NOT\_ - IMPLEMENTED**  
 osclio, 94  
**OSCL\_FILEMGMT\_E\_OK**  
 osclio, 94  
**OSCL\_FILEMGMT\_E\_PATH\_NOT\_FOUND**  
 osclio, 94  
**OSCL\_FILEMGMT\_E\_PATH\_TOO\_LONG**  
 osclio, 94  
**OSCL\_FILEMGMT\_E\_PERMISSION\_ - DENIED**  
 osclio, 94  
**OSCL\_FILEMGMT\_E\_SYS\_SPECIFIC**  
 osclio, 94  
**OSCL\_FILEMGMT\_E\_UNKNOWN**  
 osclio, 94  
**OSCL\_FILEMGMT\_ERR\_TYPE**  
 osclio, 94  
**OSCL\_FILEMGMT\_MODE\_DIR**  
 osclio, 94  
**OSCL\_FILEMGMT\_MODES**  
 osclio, 94  
**OSCL\_FILEMGMT\_PERMS**  
 osclio, 94  
**OSCL\_FILEMGMT\_PERMS\_EXECUTE**  
 osclio, 94  
**OSCL\_FILEMGMT\_PERMS\_READ**  
 osclio, 94  
**OSCL\_FILEMGMT\_PERMS\_WRITE**  
 osclio, 94  
**Oscl\_FileServer**, 188  
   Oscl\_FileServer, 188  
**Oscl\_FileServer**  
   ~Oscl\_FileServer, 188  
   Close, 188  
   Connect, 188  
   Oscl\_DeleteFile, 188, 189  
   Oscl\_File, 189  
   Oscl\_FileServer, 188  
   OsclNativeFile, 189  
**OSCL\_FIRST\_CATCH**  
   osclerror, 85  
**OSCL\_FIRST\_CATCH\_ANY**  
   osclerror, 85  
**oscl\_floor**  
   osclutil, 71  
**OSCL\_FREE**  
   osclmemory, 53  
**oscl\_free**  
   osclmemory, 53  
**OSCL\_FSSSTAT**  
   oscilio, 93  
**oscl\_fsstat**, 190  
  
 freebytes, 190  
 totalbytes, 190  
**OSCL\_FUNCTION\_PTR**  
   osclconfig\_compiler\_warnings.h, 786  
**oscl\_getcwd**  
   oscilio, 95, 96  
**OSCL\_GetLastError**  
   osclerror, 89  
**OSCL\_HAS\_ANDROID\_FILE\_IO\_SUPPORT**  
   osclconfig.h, 783  
**OSCL\_HAS\_ANDROID\_SUPPORT**  
   osclconfig.h, 783  
**OSCL\_HAS\_ANSI\_64BIT\_FILE\_IO\_ - SUPPORT**  
   osclconfig\_io.h, 793  
**OSCL\_HAS\_ANSI\_FILE\_IO\_SUPPORT**  
   osclconfig\_io.h, 793  
**OSCL\_HAS\_ANSI\_MATH\_SUPPORT**  
   osclconfig\_unix\_android.h, 818  
   osclconfig\_unix\_common.h, 822  
**OSCL\_HAS\_ANSI\_MEMORY\_FUNCS**  
   osclconfig\_ansi\_memory.h, 784  
**OSCL\_HAS\_ANSI\_STDIO\_SUPPORT**  
   osclconfig\_unix\_android.h, 818  
   osclconfig\_unix\_common.h, 822  
**OSCL\_HAS\_ANSI\_STDLIB\_SUPPORT**  
   osclconfig\_unix\_android.h, 818  
   osclconfig\_unix\_common.h, 822  
**OSCL\_HAS\_ANSI\_STRING\_SUPPORT**  
   osclconfig\_unix\_android.h, 818  
   osclconfig\_unix\_common.h, 822  
**OSCL\_HAS\_ANSI\_WIDE\_STRING\_ - SUPPORT**  
   osclconfig\_unix\_android.h, 818  
   osclconfig\_unix\_common.h, 822  
**OSCL\_HAS\_BASIC\_LOCK**  
   osclconfig\_unix\_android.h, 818  
   osclconfig\_unix\_common.h, 822  
**OSCL\_HAS\_BERKELEY\_SOCKETS**  
   osclconfig, 20  
   osclconfig\_io.h, 793  
**OSCL\_HAS\_ERRNO\_H**  
   osclconfig\_error.h, 787  
**OSCL\_HAS\_EXCEPTIONS**  
   osclconfig\_error.h, 787  
**OSCL\_HAS\_GLOB**  
   osclconfig\_io.h, 793  
**OSCL\_HAS\_GLOBAL\_NEW\_DELETE**  
   osclconfig\_memory.h, 803  
   osclmemory, 53  
**OSCL\_HAS\_GLOBAL\_VARIABLE\_ - SUPPORT**  
   osclconfig\_unix\_android.h, 818  
   osclconfig\_unix\_common.h, 822

---

**OSCL\_HAS\_HEAP\_BASE\_SUPPORT**  
 osclconfig\_memory.h, 803

**OSCL\_HAS\_LARGE\_FILE\_SUPPORT**  
 osclconfig\_io.h, 793

**OSCL\_HAS\_MSWIN\_FILE\_IO\_SUPPORT**  
 osclconfig\_io.h, 793

**OSCL\_HAS\_MSWIN\_PARTIAL\_SUPPORT**  
 osclconfig, 21

**OSCL\_HAS\_MSWIN\_SUPPORT**  
 osclconfig, 21  
 osclconfig\_unix\_android.h, 818  
 osclconfig\_unix\_common.h, 822

**OSCL\_HAS\_NATIVE\_FILE\_CACHE\_-ENABLE**  
 osclconfig\_io.h, 793

**OSCL\_HAS\_NATIVE\_INT64\_TYPE**  
 osclconfig\_unix\_android.h, 818  
 osclconfig\_unix\_common.h, 822

**OSCL\_HAS\_NATIVE\_UINT64\_TYPE**  
 osclconfig\_unix\_android.h, 818  
 osclconfig\_unix\_common.h, 822

**OSCL\_HAS\_NON\_PREEMPTIVE\_-THREAD\_SUPPORT**  
 osclconfig\_proc\_unix\_android.h, 810  
 osclconfig\_proc\_unix\_common.h, 812

**OSCL\_HAS\_PTHREAD\_SUPPORT**  
 osclconfig, 21  
 osclconfig\_proc\_unix\_android.h, 810  
 osclconfig\_proc\_unix\_common.h, 812

**OSCL\_HAS\_PV\_C\_OS\_API\_MEMORY\_-FUNCS**  
 osclconfig, 21

**OSCL\_HAS\_PV\_C\_OS\_SUPPORT**  
 osclconfig, 21

**OSCL\_HAS\_PV\_C\_OS\_TIME\_FUNCS**  
 osclconfig, 21

**OSCL\_HAS\_PV\_FILE\_CACHE**  
 osclconfig\_io.h, 793

**OSCL\_HAS\_RUNTIME\_LIB\_LOADING\_-SUPPORT**  
 osclconfig\_lib.h, 800

**OSCL\_HAS\_SAVAJE\_IO\_SUPPORT**  
 osclconfig, 21

**OSCL\_HAS\_SAVAJE\_SUPPORT**  
 osclconfig, 21

**OSCL\_HAS\_SEM\_TIMEDWAIT\_SUPPORT**  
 osclconfig, 21  
 osclconfig\_proc\_unix\_android.h, 810  
 osclconfig\_proc\_unix\_common.h, 812

**OSCL\_HAS\_SETJMP\_H**  
 osclconfig\_error.h, 787

**OSCL\_HAS\_SINGLETON\_SUPPORT**  
 osclbase, 31

**OSCL\_HAS\_SOCKET\_SUPPORT**  
 osclconfig\_io.h, 793

**OSCL\_HAS\_SYMBIAN\_COMPATIBLE\_IO\_-FUNCTION**  
 osclconfig, 21  
 osclconfig\_io.h, 793

**OSCL\_HAS\_SYMBIAN\_DNS\_SERVER**  
 osclconfig, 21  
 osclconfig\_io.h, 793

**OSCL\_HAS\_SYMBIAN\_ERRORTRAP**  
 osclconfig, 21  
 osclconfig\_error.h, 787

**OSCL\_HAS\_SYMBIAN\_MATH**  
 osclconfig, 21  
 osclconfig\_util.h, 823

**OSCL\_HAS\_SYMBIAN\_MEMORY\_FUNCS**  
 osclconfig, 21  
 osclconfig\_memory.h, 803

**OSCL\_HAS\_SYMBIAN\_SCHEDULER**  
 osclconfig, 21  
 osclconfig\_proc\_unix\_android.h, 810  
 osclconfig\_proc\_unix\_common.h, 812

**OSCL\_HAS\_SYMBIAN\_SOCKET\_SERVER**  
 osclconfig, 21  
 osclconfig\_io.h, 793

**OSCL\_HAS\_SYMBIAN\_SUPPORT**  
 osclconfig, 21  
 osclconfig\_unix\_android.h, 818  
 osclconfig\_unix\_common.h, 822

**OSCL\_HAS\_SYMBIAN\_TIMERS**  
 osclconfig, 21  
 osclconfig\_util.h, 823

**OSCL\_HAS\_THREAD\_SUPPORT**  
 osclconfig\_proc\_unix\_android.h, 810  
 osclconfig\_proc\_unix\_common.h, 812

**OSCL\_HAS\_TLS\_SUPPORT**  
 osclconfig\_unix\_android.h, 818  
 osclconfig\_unix\_common.h, 822

**OSCL\_HAS\_UNICODE\_SUPPORT**  
 osclconfig\_unix\_android.h, 818  
 osclconfig\_unix\_common.h, 822

**OSCL\_HAS\_UNIX\_SUPPORT**  
 osclconfig, 21  
 osclconfig\_unix\_android.h, 818  
 osclconfig\_unix\_common.h, 822

**OSCL\_HAS\_UNIX\_TIME\_FUNCS**  
 osclconfig, 21  
 osclconfig\_time.h, 813

**oscl\_heapbase.h, 682**

**OSCL\_HeapString, 191**  
 osclutil, 71

**OSCL\_HeapString**  
 chartype, 192

**OSCL\_String, 192**

**OSCL\_HeapStringA, 193**

OSCL\_HeapStringA, 194, 195  
**OSCL\_HeapStringA**  
 ~OSCL\_HeapStringA, 195  
 chartype, 194  
 get\_cstr, 195  
 get\_maxsize, 195  
 get\_size, 195  
 get\_str, 196  
 operator=, 196  
 OSCL\_HeapStringA, 194, 195  
 OSCL\_String, 196  
 set, 196  
 oscl\_init.h, 683  
**OSCL\_INLINE**  
 osclbase, 31  
**Oscl\_Int64\_Utils**, 197  
 get\_int64\_lower32, 198  
 get\_int64\_middle32, 198  
 get\_int64\_upper32, 198  
 get\_uint64\_lower32, 198  
 get\_uint64\_middle32, 198  
 get\_uint64\_upper32, 198  
 set\_int64, 198  
 set\_uint64, 198  
 oscl\_int64\_utils.h, 684  
 \_OscIInteger64Transport, 684  
**OSCL\_INTEGERS\_WORD\_ALIGNED**  
 osclconfig, 21  
**OSCL\_IO\_EXTENSION\_MAXLEN**  
 osclio, 93  
**OSCL\_IO\_FILENAME\_MAXLEN**  
 osclio, 93  
 oscl\_ip\_socket.h, 685  
**OSCL\_IPPROTO\_TCP**  
 osclconfig\_io.h, 793  
**OSCL\_IPPROTO\_UDP**  
 osclconfig\_io.h, 793  
 oscl\_isdigit  
 osclutil, 66  
**OSCL\_IsErrnoSupported**  
 osclerror, 89  
 oscl\_isLetter  
 osclbase, 35  
**OSCL\_JUMP\_MAX\_JUMP\_MARKS**  
 osclerror, 85  
**OSCL\_LAST\_CATCH**  
 osclerror, 85  
**OSCL\_LEAVE**  
 osclerror, 86  
**Oscl\_Less**, 199  
 operator(), 199  
**OSCL\_LIB\_READ\_DEBUG\_LIBS**  
 osclconfig\_lib.h, 800  
**Oscl\_Linked\_List**, 200  
 ~Oscl\_Linked\_List, 200  
 add\_element, 201  
 add\_to\_front, 201  
 check\_list, 201  
 dequeue\_element, 201  
 get\_element, 201  
 get\_first, 201  
 get\_index, 202  
 get\_next, 202  
 get\_num\_elements, 202  
 move\_to\_end, 202  
 move\_to\_front, 202  
 Oscl\_Linked\_List, 200  
 remove\_element, 203  
 oscl\_linked\_list.h, 686  
**Oscl\_Linked\_List\_Base**, 204  
 ~Oscl\_Linked\_List\_Base, 205  
 add\_element, 205  
 add\_to\_front, 205  
 check\_list, 205  
 construct, 205  
 destroy, 205  
 get\_element, 205  
 get\_first, 206  
 get\_index, 206  
 get\_next, 206  
 head, 207  
 iterator, 207  
 move\_to\_end, 206  
 move\_to\_front, 206  
 num\_elements, 207  
 remove\_element, 207  
 sizeof\_T, 207  
 tail, 207  
 oscl\_lock\_base.h, 687  
**oscl\_log**  
 osclutil, 72  
**oscl\_log10**  
 osclutil, 72  
**OSCL\_MALLOC**  
 osclmemory, 54  
**oscl\_malloc**  
 osclmemory, 54  
**Oscl\_Map**, 208  
 begin, 211  
 clear, 211  
 const\_iterator, 210  
 const\_reference, 210  
 count, 211  
 empty, 211  
 end, 211  
 equal\_range, 211  
 erase, 212  
 find, 212

---

```

insert, 212
iterator, 210
key_comp, 213
key_compare, 210
key_type, 210
lower_bound, 213
max_size, 213
operator=, 213
operator[], 213
Oscl_Map, 210
pair_citerator_citerator, 210
pair_iterator_bool, 210
pair_iterator_iterator, 210
pointer, 210
reference, 210
self, 210
size, 213
size_type, 210
upper_bound, 213, 214
value_comp, 214
value_type, 210
oscl_map.h, 688
  OSCL_DISABLE_WARNING_-
    TRUNCATE_DEBUG_MESSAGE,
    688
Oscl_Map::value_compare, 215
  comp, 215
  operator(), 215
  Oscl_Map< Key, T, Alloc, Compare >, 215
  value_compare, 215
Oscl_Map< Key, T, Alloc, Compare >
  Oscl_Map::value_compare, 215
oscl_math.h, 689
OSCL_MAX
  osclbase, 31
OSCL_MAX_TRAP_LEVELS
  oscrror, 86
oscl_media_data.h, 690
oscl_media_status.h, 691
oscl_mem.h, 692
  operator delete, 694
  operator new, 694
  OSCL_DISABLE_WARNING_-
    TRUNCATE_DEBUG_MESSAGE,
    694
oscl_mem_align.h, 695
oscl_mem_aligned_size
  osclmemory, 57
  OsclMemPoolAllocator, 428
oscl_mem_audit.h, 696
  OSCL_DISABLE_WARNING_-
    TRUNCATE_DEBUG_MESSAGE,
    697
oscl_mem_audit_internals.h, 698
  OSCL_DISABLE_WARNING_-
    TRUNCATE_DEBUG_MESSAGE,
    698
oscl_mem_auto_ptr.h, 699
  OSCL_DISABLE_WARNING_-
    TRUNCATE_DEBUG_MESSAGE,
    699
oscl_mem_basic_functions.h, 700
oscl_mem_inst.h, 701
oscl_mem_mempool.h, 702
oscl_memcmp
  osclmemory, 58
oscl_memcpy
  osclmemory, 58
  OSCL_MEMFRAG_PTR_BEFORE_LEN
    osclconfig_unix_android.h, 818
    osclconfig_unix_common.h, 822
oscl_memmove
  osclmemory, 58
oscl_memmove32
  osclmemory, 58
oscl_mempool_allocator.h, 703
oscl_memset
  osclmemory, 59
oscl_memsize_t
  osclconfig_ansi_memory.h, 784
OSCL_MIN
  osclbase, 31
oscl_mkdir
  osclio, 96
Oscl_MTLLinked_List, 217
  ~Oscl_MTLLinked_List, 217
  add_element, 218
  add_to_front, 218
  dequeue_element, 218
  get_element, 218
  get_index, 218
  move_to_end, 218
  move_to_front, 219
  Oscl_MTLLinked_List, 217
  remove_element, 219
  the_list, 219
oscl_mutex.h, 704
  OsclNoYieldMutex, 704
oscl_namestring.h, 705
OSCL_NATIVE_INT64_TYPE
  osclconfig_unix_android.h, 818
  osclconfig_unix_common.h, 822
OSCL_NATIVE_UINT64_TYPE
  osclconfig.h, 783
  osclconfig_unix_android.h, 818
  osclconfig_unix_common.h, 822
OSCL_NATIVE_WCHAR_TYPE
  osclconfig_unix_android.h, 818

```

osclconfig\_unix\_common.h, 822  
**OSCL\_NEW**  
 osclmemory, 54  
 oscl\_opaque\_type.h, 706  
**Oscl\_Opaque\_Type\_Alloc**, 221  
 allocate, 221  
 construct, 221  
 deallocate, 221  
 destroy, 221  
**Oscl\_Opaque\_Type\_Alloc\_LL**, 222  
 allocate, 222  
 compare\_data, 222  
 construct, 222  
 deallocate, 222  
 destroy, 222  
 get\_data, 223  
 get\_next, 223  
 set\_next, 223  
**Oscl\_Opaque\_Type\_Compare**, 224  
 compare\_EQ, 224  
 compare\_LT, 224  
 swap, 224  
**OSCL\_PACKED\_STRUCT\_BEGIN**  
 osclconfig.h, 783  
**OSCL\_PACKED\_STRUCT\_END**  
 osclconfig.h, 783  
**OSCL\_PACKED\_VAR**  
 osclbase, 31  
 osclconfig.h, 783  
**Oscl\_Pair**, 226  
 first, 226  
**Oscl\_Pair**, 226  
 second, 226  
**OSCL\_PERF\_SUMMARY\_LOGGING**  
 osclproc, 101  
**OSCL\_PLACEMENT\_NEW**  
 osclmemory, 54  
 oscl\_pow  
 osclutil, 72  
 oscl\_pqueue.h, 707  
 oscl\_pqueue\_test  
 OsclPriorityQueue, 459  
 oscl\_proctstatus.h, 708  
**Oscl\_Queue**, 227  
 ~Oscl\_Queue, 228  
 back, 228  
 clear, 228  
 const\_reference, 228  
 front, 229  
**Oscl\_Queue**, 228  
 pointer, 228  
 pop, 229  
 push, 229  
 reference, 228  
 size\_type, 228  
 value\_type, 228  
 oscl\_queue.h, 709  
**Oscl\_Queue\_Base**, 230  
 ~Oscl\_Queue\_Base, 230  
 bufsize, 232  
 capacity, 231  
 clear, 231  
 construct, 231  
 destroy, 231  
 elems, 232  
 empty, 231  
 ifront, 232  
 irear, 232  
 numelems, 232  
 pop, 231  
 push, 231  
 reserve, 231  
 size, 231  
 sizeof\_T, 232  
 oscl\_rand.h, 710  
**OSCL\_RAND\_MAX**  
 osclconfig\_util.h, 823  
**Oscl\_Rb\_Tree**, 233  
 ~Oscl\_Rb\_Tree, 235  
 begin, 235  
 clear, 235  
 const\_iterator, 235  
 const\_pointer, 235  
 const\_reference, 235  
 count, 235  
 difference\_type, 235  
 empty, 235  
 end, 235  
 equal\_range, 235  
 erase, 235  
 find, 235  
 insert\_unique, 235  
 iterator, 235  
 key\_type, 235  
 link\_type, 235  
 lower\_bound, 235  
 max\_size, 235  
 operator=, 235  
 Oscl\_Rb\_Tree, 235  
 pointer, 235  
 reference, 235  
 size, 235  
 size\_type, 235  
 upper\_bound, 235  
 value\_type, 235  
**Oscl\_Rb\_Tree\_Base**, 237  
 base\_link\_type, 237  
 rebalance, 237

rebalance\_for\_erase, 237  
 rotate\_left, 237  
 rotate\_right, 237  
**Oscl\_Rb\_Tree\_Const\_Iterator**, 238  
     base\_link\_type, 239  
     const\_iterator, 239  
     link\_type, 239  
     node, 239  
     operator \*, 239  
     operator!=, 239  
     operator++, 239  
     operator-, 239  
     operator->, 239  
     operator==, 239  
**Oscl\_Rb\_Tree\_Const\_Iterator**, 239  
     pointer, 239  
     reference, 239  
     self, 239  
     value\_type, 239  
**Oscl\_Rb\_Tree\_Iterator**, 241  
     base\_link\_type, 242  
     iterator, 242  
     link\_type, 242  
     node, 242  
     operator \*, 242  
     operator!=, 242  
     operator++, 242  
     operator-, 242  
     operator->, 242  
     operator==, 242  
**Oscl\_Rb\_Tree\_Iterator**, 242  
     pointer, 242  
     reference, 242  
     self, 242  
     value\_type, 242  
**Oscl\_Rb\_Tree\_Node**, 244  
     link\_type, 244  
     value, 244  
     value\_type, 244  
**Oscl\_Rb\_Tree\_Node\_Base**  
     black, 245  
     red, 245  
**Oscl\_Rb\_Tree\_Node\_Base**, 245  
     base\_link\_type, 245  
     color, 246  
     color\_type, 245  
     left, 246  
     maximum, 246  
     minimum, 246  
     parent, 246  
     RedBl, 245  
     right, 246  
**OSCL\_READSET\_FLAG**  
     oscl\_socket\_serv\_imp\_pv.h, 748

    OSCL\_REALLOC  
         osclmemory, 54  
**oscl\_realloc**  
         osclmemory, 54  
     oscl\_refcounter.h, 711  
     oscl\_refcounter\_memfrag.h, 712  
     oscl\_registry\_access\_client.h, 713  
     oscl\_registry\_client.h, 714  
     oscl\_registry\_client\_impl.h, 715  
     oscl\_registry\_serv\_impl.h, 716  
     oscl\_registry\_serv\_impl\_global.h, 717  
     oscl\_registry\_serv\_impl\_tls.h, 718  
     oscl\_registry\_types.h, 719  
**OSCL\_REINTERPRET\_CAST**  
     osclbase, 31  
**OSCL\_RELEASE\_BUILD**  
     osclconfig.h, 783  
**oscl\_rename**  
     osclo, 96, 97  
**OSCL\_REQUEST\_ERR\_CANCEL**  
     osclproc, 102  
**OSCL\_REQUEST\_ERR\_GENERAL**  
     osclproc, 102  
**OSCL\_REQUEST\_ERR\_NONE**  
     osclproc, 102  
**OSCL\_REQUEST\_PENDING**  
     osclproc, 102  
**oscl\_rmdir**  
     osclo, 97  
**oscl\_scheduler.h**, 720  
**oscl\_scheduler\_ao.h**, 721  
**oscl\_scheduler\_aobase.h**, 722  
**oscl\_scheduler\_readyq.h**, 723  
**oscl\_scheduler\_threadcontext.h**, 724  
**oscl\_scheduler\_tuneables.h**, 725  
**oscl\_scheduler\_types.h**, 726  
**OSCL\_SD\_BOTH**  
     osclconfig\_io.h, 793  
**OSCL\_SD\_RECEIVE**  
     osclconfig\_io.h, 793  
**OSCL\_SD\_SEND**  
     osclconfig\_io.h, 793  
**Oscl\_Select1st**, 247  
     operator(), 247  
**oscl\_semaphore.h**, 727  
**OSCL\_SetLastError**  
     osclerror, 89  
**oscl\_shared\_ptr.h**, 728  
**oscl\_sin**  
     osclutil, 72  
**oscl\_singleton.h**, 729  
     OSCL\_SINGLETON\_ID\_CPM\_PLUGIN, 730  
     OSCL\_SINGLETON\_ID\_LAST, 730

OSCL\_SINGLETON\_ID\_OMX, [730](#)  
 OSCL\_SINGLETON\_ID\_-  
   OMXMASTERCORE, [730](#)  
 OSCL\_SINGLETON\_ID\_OSCLMEM,  
   [730](#)  
 OSCL\_SINGLETON\_ID\_-  
   OSCLREGISTRY, [730](#)  
 OSCL\_SINGLETON\_ID\_-  
   PAYLOADPARSER, [730](#)  
 OSCL\_SINGLETON\_ID\_-  
   PVERRORTRAP, [730](#)  
 OSCL\_SINGLETON\_ID\_PVLOGGER,  
   [730](#)  
 OSCL\_SINGLETON\_ID\_-  
   PVMFRECOGNIZER, [730](#)  
 OSCL\_SINGLETON\_ID\_-  
   PVSCHEDULER, [730](#)  
 OSCL\_SINGLETON\_ID\_-  
   SDPMEDIAPARSER, [730](#)  
 OSCL\_SINGLETON\_ID\_TEST, [730](#)  
 OSCL\_SINGLETON\_ID\_TICKCOUNT,  
   [730](#)  
 OSCL\_SINGLETON\_ID\_CPM\_PLUGIN  
   oscl\_singleton.h, [730](#)  
 OSCL\_SINGLETON\_ID\_LAST  
   oscl\_singleton.h, [730](#)  
 OSCL\_SINGLETON\_ID\_OMX  
   oscl\_singleton.h, [730](#)  
 OSCL\_SINGLETON\_ID\_-  
   OMXMASTERCORE  
   oscl\_singleton.h, [730](#)  
 OSCL\_SINGLETON\_ID\_OSCLMEM  
   oscl\_singleton.h, [730](#)  
 OSCL\_SINGLETON\_ID\_OSCLREGISTRY  
   oscl\_singleton.h, [730](#)  
 OSCL\_SINGLETON\_ID\_PAYLOADPARSER  
   oscl\_singleton.h, [730](#)  
 OSCL\_SINGLETON\_ID\_PVERRORTRAP  
   oscl\_singleton.h, [730](#)  
 OSCL\_SINGLETON\_ID\_PVLOGGER  
   oscl\_singleton.h, [730](#)  
 OSCL\_SINGLETON\_ID\_-  
   PVMFRECOGNIZER  
   oscl\_singleton.h, [730](#)  
 OSCL\_SINGLETON\_ID\_PVSCHEDULER  
   oscl\_singleton.h, [730](#)  
 OSCL\_SINGLETON\_ID\_-  
   SDPMEDIAPARSER  
   oscl\_singleton.h, [730](#)  
 OSCL\_SINGLETON\_ID\_TEST  
   oscl\_singleton.h, [730](#)  
 OSCL\_SINGLETON\_ID\_TICKCOUNT  
   oscl\_singleton.h, [730](#)  
 oscl\_snprintf

osclutil, [72](#)  
 oscl\_snprintf.h, [731](#)  
 OSCL SOCK DATAGRAM  
   osclconfig\_io.h, [793](#)  
 OSCL SOCK STREAM  
   osclconfig\_io.h, [793](#)  
 oscl\_socket.h, [732](#)  
 oscl\_socket\_accept.h, [733](#)  
 oscl\_socket\_bind.h, [734](#)  
 oscl\_socket\_connect.h, [735](#)  
 oscl\_socket\_imp.h, [736](#)  
 oscl\_socket\_imp\_base.h, [737](#)  
 oscl\_socket\_imp\_pv.h, [738](#)  
   PVSOCK\_ERR\_BAD\_PARAM, [738](#)  
   PVSOCK\_ERR\_NOT\_IMPLEMENTED,  
     [738](#)  
   PVSOCK\_ERR\_SERV\_NOT\_-  
     CONNECTED, [738](#)  
   PVSOCK\_ERR SOCK NO SERV, [738](#)  
   PVSOCK\_ERR SOCK NOT\_-  
     CONNECTED, [738](#)  
   PVSOCK\_ERR SOCK NOT\_OPEN, [738](#)  
 oscl\_socket\_listen.h, [739](#)  
   OSCL\_SOCKET\_LISTEN\_H\_-  
     INCLUDEDd, [739](#)  
 OSCL\_SOCKET\_LISTEN\_H\_INCLUDEDd  
   oscl\_socket\_listen.h, [739](#)  
 oscl\_socket\_method.h, [740](#)  
   MSEC\_TO\_MICROSEC, [740](#)  
 oscl\_socket\_recv.h, [741](#)  
 oscl\_socket\_recv\_from.h, [742](#)  
 oscl\_socket\_request.h, [743](#)  
 oscl\_socket\_send.h, [744](#)  
 oscl\_socket\_send\_to.h, [745](#)  
 oscl\_socket\_serv\_imp.h, [746](#)  
 oscl\_socket\_serv\_imp\_base.h, [747](#)  
 oscl\_socket\_serv\_imp\_pv.h, [748](#)  
   OSCL\_EXCEPTSET\_FLAG, [748](#)  
   OSCL\_READSET\_FLAG, [748](#)  
   OSCL\_WRITESET\_FLAG, [748](#)  
 oscl\_socket\_serv\_imp\_reqlist.h, [749](#)  
 oscl\_socket\_shutdown.h, [750](#)  
 oscl\_socket\_stats.h  
   EOsclSocket\_DataRecv, [752](#)  
   EOsclSocket\_DataSent, [752](#)  
   EOsclSocket\_Except, [751](#)  
   EOsclSocket\_OS, [751](#)  
   EOsclSocket\_Readable, [751](#)  
   EOsclSocket\_RequestAO\_Canceled, [751](#)  
   EOsclSocket\_RequestAO\_Error, [751](#)  
   EOsclSocket\_RequestAO\_Success, [751](#)  
   EOsclSocket\_RequestAO\_Timeout, [751](#)  
   EOsclSocket\_ServPoll, [751](#)

EOscI Socket\_ServRequestCancelIssued, [752](#)  
 EOscI Socket\_ServRequestComplete, [752](#)  
 EOscI Socket\_ServRequestIssued, [751](#)  
 EOscI Socket\_Writable, [751](#)  
 EOscI SocketServ\_LastEvent, [751](#)  
 EOscI SocketServ\_LoopsckError, [752](#)  
 EOscI SocketServ\_LoopsckOk, [752](#)  
 EOscI SocketServ\_SelectActivity, [751](#)  
 EOscI SocketServ\_SelectNoActivity, [751](#)  
 EOscI SocketServ\_SelectRescheduleAsap, [751](#)  
 EOscI SocketServ\_SelectReschedulePoll, [751](#)  
 oscl\_socket\_stats.h, [751](#)  
 TOscI SocketServStatEvent, [751](#)  
 TOscI SocketStatEvent, [751](#)  
 oscl\_socket\_tuneables.h, [753](#)  
 PV\_OSCL\_SOCKET\_1MB\_RECV\_BUF, [753](#)  
 PV\_OSCL\_SOCKET\_SERVER\_-  
     LOGGER\_OUTPUT, [753](#)  
 PV\_OSCL\_SOCKET\_STATS\_LOGGING, [753](#)  
 PV\_SOCKET\_REQUEST\_AO\_-  
     PRIORITY, [753](#)  
 PV\_SOCKET\_SERVER, [753](#)  
 PV\_SOCKET\_SERVER\_AO\_-  
     INTERVAL\_MSEC, [754](#)  
 PV\_SOCKET\_SERVER\_AO\_PRIORITY, [754](#)  
 PV\_SOCKET\_SERVER\_IS\_THREAD, [754](#)  
 PV\_SOCKET\_SERVER\_SELECT, [754](#)  
 PV\_SOCKET\_SERVER\_SELECT\_-  
     LOOPBACK\_SOCKET, [754](#)  
 PV\_SOCKET\_SERVER\_SELECT\_-  
     TIMEOUT\_MSEC, [754](#)  
 PV\_SOCKET\_SERVER\_THREAD\_-  
     PRIORITY, [754](#)  
 PV\_SOCKET\_SERVI\_STATS, [754](#)  
 oscl\_socket\_types.h  
     EPVSocket\_Last, [756](#)  
     EPVSocketAccept, [756](#)  
     EPVSocketBind, [756](#)  
     EPVSocketBothShutdown, [756](#)  
     EPVSocketCancel, [755](#)  
     EPVSocketConnect, [756](#)  
     EPVSocketFailure, [755](#)  
     EPVSocketListen, [756](#)  
     EPVSocketPending, [755](#)  
     EPVSocketRecv, [756](#)  
     EPVSocketRecvFrom, [756](#)  
     EPVSocketRecvShutdown, [756](#)  
 EPVSocketSend, [756](#)  
 EPVSocketSendShutdown, [756](#)  
 EPVSocketSendTo, [756](#)  
 EPVSocketShutdown, [756](#)  
 EPVSocketSuccess, [755](#)  
 EPVSocketTimeout, [755](#)  
 oscl\_socket\_types.h, [755](#)  
 PVNETWORKADDRESS\_LEN, [755](#)  
 TPVSocketEvent, [755](#)  
 TPVSocketFxn, [755](#)  
 TPVSocketShutdown, [756](#)  
 oscl\_sqrt  
     osclutil, [72](#)  
 OSCL\_StackString, [248](#)  
     osclutil, [72, 73](#)  
 OSCL\_StackString  
     chartype, [249](#)  
     OSCL\_String, [249](#)  
 oscl\_stat  
     osclio, [97, 98](#)  
 OSCL\_STAT\_BUF  
     osclio, [93](#)  
 oscl\_stat\_buf, [250](#)  
     mode, [250](#)  
     perms, [250](#)  
 oscl\_statfs  
     osclio, [98](#)  
 OSCL\_STATIC\_CAST  
     osclbase, [31](#)  
 oscl\_stdstring.h, [757](#)  
 oscl\_str\_escape\_xml  
     osclutil, [73](#)  
 oscl\_str\_is\_valid\_utf8  
     osclutil, [74](#)  
 oscl\_str\_need\_escape\_xml  
     osclutil, [74](#)  
 oscl\_str\_ptr\_len.h, [759](#)  
 oscl\_str\_truncate\_utf8  
     osclutil, [74](#)  
 oscl\_str\_unescape\_uri  
     osclutil, [75](#)  
 oscl\_streat  
     osclbase, [36](#)  
 oscl\_strchr  
     osclbase, [36, 37](#)  
 oscl\_strcmp  
     osclbase, [37](#)  
 OSCL\_StrError  
     osclerror, [89](#)  
 OSCL\_String, [251](#)  
     ~OSCL\_String, [252](#)  
     append\_rep, [252](#)  
     chartype, [252](#)  
     get\_cstr, [252](#)

---

get\_maxsize, 252  
 get\_size, 252  
 get\_str, 253  
 hash, 253  
 is\_writable, 253  
 operator!=, 253  
 operator+=, 253  
 operator<, 253  
 operator<=, 253  
 operator=, 253, 254  
 operator==, 254  
 operator>, 254  
 operator>=, 254  
 operator[], 254  
 OSCL\_FastString, 172  
 OSCL\_HeapString, 192  
 OSCL\_HeapStringA, 196  
 OSCL\_StackString, 249  
 OSCL\_String, 252  
 read, 254  
 set\_len, 254  
 set\_rep, 254, 255  
 write, 255  
 oscl\_string.h, 760  
 oscl\_string\_containers.h, 761  
 oscl\_string\_rep.h, 762  
 oscl\_string\_uri.h, 763  
 oscl\_string\_utf8.h, 764  
 oscl\_string\_utils.h, 765  
 oscl\_string\_xml.h, 766  
 oscl\_strlen  
     osclbase, 37  
 oscl\_strncat  
     osclbase, 38  
 oscl\_strcmp  
     osclbase, 38, 39  
 oscl\_strncpy  
     osclbase, 39  
 oscl strrchr  
     osclbase, 40  
 oscl\_strset  
     osclbase, 40  
 oscl strstr  
     osclbase, 40, 41  
 Oscl\_Tag, 256  
     ~Oscl\_Tag, 256  
     operator<, 256  
     Oscl\_Tag, 256  
     tag, 256  
     tagAllocator, 256  
 Oscl\_Tag\_Base, 258  
     operator(), 259  
     size\_type, 259  
     tag\_ancestor, 259  
                 tag\_base\_type, 259  
                 tag\_base\_unit, 259  
                 tag\_cmp, 259  
                 tag\_copy, 259  
                 tag\_depth, 259  
                 tag\_len, 259  
                 Oscl\_TagTree, 260  
                 Oscl\_TagTree, 261  
                 Oscl\_TagTree  
                     ~Oscl\_TagTree, 261  
                     begin, 261  
                     children\_type, 261  
                     clear, 262  
                     count, 262  
                     empty, 262  
                     end, 262  
                     erase, 262  
                     find, 262  
                     insert, 263  
                     map\_type, 261  
                     node\_ptr, 261  
                     node\_type, 261  
                     operator=, 263  
                     operator[], 263  
                     Oscl\_TagTree, 261  
                     pair\_iterator\_bool, 261  
                     size, 263  
                     size\_type, 261  
                     tag\_base\_type, 261  
                     tag\_type, 261  
                     value\_type, 261  
                 oscl\_tagtree.h, 767  
                     OSCL\_DISABLE\_WARNING\_-  
                         TRUNCATE\_DEBUG\_MESSAGE,  
                         767  
                 Oscl\_TagTree::const\_iterator, 264  
                 Oscl\_TagTree::const\_iterator  
                     const\_iterator, 265  
                     mapit, 265  
                     mapiter, 265  
                     operator \*, 265  
                     operator!=, 265  
                     operator++, 265  
                     operator-, 265  
                     operator->, 265  
                     operator==, 265  
                     pointer, 265  
                     reference, 265  
                     self, 265  
                 Oscl\_TagTree::iterator, 267  
                 Oscl\_TagTree::iterator  
                     iterator, 268  
                     mapit, 268  
                     mapiter, 268

operator \*, 268  
 operator!=, 268  
 operator++, 268  
 operator-, 268  
 operator->, 268  
 operator==, 268  
 pointer, 268  
 reference, 268  
 self, 268  
**Oscl\_TagTree::Node**, 270  
**Oscl\_TagTree::Node**  
   children, 271  
   children\_type, 271  
   depth, 271  
   Node, 271  
   parent, 271  
   sort\_children, 271  
   tag, 271  
   value, 271  
**Oscl\_TAlloc**, 272  
   ~Oscl\_TAlloc, 273  
   address, 273  
   alloc\_and\_construct, 273  
   alloc\_and\_construct\_fl, 273  
   allocate, 273  
   allocate\_fl, 273  
   const\_pointer, 273  
   const\_reference, 273  
   construct, 273  
   deallocate, 273  
   destroy, 273  
   destruct\_and\_dealloc, 273  
   pointer, 273  
   reference, 273  
   size\_type, 273  
   value\_type, 273  
**Oscl\_TAlloc::rebind**, 275  
   other, 275  
**oscl\_tan**  
   osclutil, 76  
**OSCL\_TCHAR**  
   osclbase, 32  
**oscl\_tcp\_socket.h**, 768  
**OSCL\_TEMPLATED\_DESTRUCTOR\_CALL**  
   osclbase, 31  
   osclconfig.h, 783  
**oscl\_thread.h**  
   Start\_on\_creation, 769  
   Suspend\_on\_creation, 769  
   ThreadPriorityAboveNormal, 770  
   ThreadPriorityBelowNormal, 769  
   ThreadPriorityHighest, 770  
   ThreadPriorityLow, 769  
   ThreadPriorityLowest, 769  
   ThreadPriorityNormal, 769  
   ThreadPriorityTimeCritical, 770  
**oscl\_thread.h**, 769  
   OsclThread\_State, 769  
   OsclThreadPriority, 769  
   TOsclThreadFuncPtr, 769  
**OSCL\_THREAD DECL**  
   osclconfig\_proc\_unix\_android.h, 810  
   osclconfig\_proc\_unix\_common.h, 812  
**oscl\_tickcount.h**, 771  
**oscl\_time.h**, 772  
**oscl\_timer.h**, 774  
**oscl\_tls.h**, 775  
**OSCL\_TLS\_BASE\_SLOTS**  
   osclbase, 31  
**OSCL\_TLS\_EXTERNAL\_SLOTS**  
   osclbase, 31  
**OSCL\_TLS\_GET\_FUNC**  
   osclconfig\_unix\_android.h, 818  
   osclconfig\_unix\_common.h, 822  
**OSCL\_TLS\_ID\_BASE\_LAST**  
   osclbase, 43  
**OSCL\_TLS\_ID\_ERRORHOOK**  
   osclbase, 43  
**OSCL\_TLS\_ID\_MAGICNUM**  
   osclbase, 43  
**OSCL\_TLS\_ID\_OSCLREGISTRY**  
   osclbase, 43  
**OSCL\_TLS\_ID\_PAYLOADPARSER**  
   osclbase, 43  
**OSCL\_TLS\_ID\_PVERRORTRAP**  
   osclbase, 43  
**OSCL\_TLS\_ID\_PVLOGGER**  
   osclbase, 43  
**OSCL\_TLS\_ID\_PVMFRECOGNIZER**  
   osclbase, 43  
**OSCL\_TLS\_ID\_PVSCHEDULER**  
   osclbase, 43  
**OSCL\_TLS\_ID\_SDPMEDIAPARSER**  
   osclbase, 43  
**OSCL\_TLS\_ID\_SQLITE3**  
   osclbase, 43  
**OSCL\_TLS\_ID\_TEST**  
   osclbase, 43  
**OSCL\_TLS\_ID\_WMDRM**  
   osclbase, 43  
**OSCL\_TLS\_IS\_KEYED**  
   osclconfig\_unix\_android.h, 818  
   osclconfig\_unix\_common.h, 822  
**OSCL\_TLS\_KEY\_CREATE\_FUNC**  
   osclconfig\_unix\_android.h, 818  
   osclconfig\_unix\_common.h, 822  
**OSCL\_TLS\_KEY\_DELETE\_FUNC**  
   osclconfig\_unix\_android.h, 818

osclconfig\_unix\_common.h, 822  
**OSCL\_TLS\_MAX\_SLOTS**  
 osclbase, 31  
**OSCL\_TLS\_STORE\_FUNC**  
 osclconfig\_unix\_android.h, 818  
 osclconfig\_unix\_common.h, 822  
**oscl\_tolower**  
 osclbase, 41  
**OSCL\_TRAP\_ALLOC\_NEW**  
 osclmemory, 54  
**OSCL\_TRAP\_AUDIT\_NEW**  
 osclmemory, 55  
**OSCL\_TRAP\_NEW**  
 osclmemory, 55  
**OSCL\_TRAPSTACK\_POP**  
 osclerror, 86  
**OSCL\_TRAPSTACK\_POPDEALLOC**  
 osclerror, 86  
**OSCL\_TRAPSTACK\_PUSH**  
 osclerror, 86  
**oscl\_tree.h**, 776  
 OSCL\_DISABLE\_WARNING\_-  
     TRUNCATE\_DEBUG\_MESSAGE,  
     776  
**OSCL\_TRY**  
 osclerror, 86  
**OSCL\_TRY\_NO\_TLS**  
 osclerror, 86  
**OSCL\_TStrPtrLen**  
 osclutil, 66  
**oscl\_types.h**, 777  
**oscl\_udp\_socket.h**, 778  
**oscl\_UncodeToUTF8**  
 osclutil, 76  
**OSCL\_UNSIGNED\_CONST**  
 osclbase, 31  
 osclconfig.h, 783  
**OSCL\_UNUSED\_ARG**  
 osclbase, 31  
**OSCL\_UNUSED\_RETURN**  
 osclbase, 32  
**oscl\_utf8conv.h**, 779  
**oscl\_UTF8ToUnicode**  
 osclutil, 76  
**oscl\_uuid.h**, 780  
 BYTES\_IN\_UUID\_ARRAY, 780  
 OsclUid32, 780  
 PV\_CHAR\_CLOSE\_BRACKET, 780  
 PV\_CHAR\_COMMA, 780  
**Oscl\_Vector**, 276  
 ~Oscl\_Vector, 277  
 back, 278  
 begin, 278  
 clear, 278  
 const\_iterator, 277  
 const\_reference, 277  
 destroy, 278  
 end, 278  
 erase, 278  
 front, 279  
 insert, 279  
 iterator, 277  
 operator=, 279  
 operator[], 279  
 Oscl\_Vector, 277  
 pointer, 277  
 pop\_back, 279  
 push\_back, 280  
 push\_front, 280  
 reference, 277  
 value\_type, 277  
**oscl\_vector.h**, 781  
**Oscl\_Vector\_Base**, 281  
 ~Oscl\_Vector\_Base, 282  
 assign\_vector, 282  
 bufsize, 284  
 capacity, 282  
 construct, 282  
 destroy, 282  
 elems, 284  
 empty, 282  
 erase, 282, 283  
 insert, 283  
 numelems, 284  
 OsclPriorityQueueBase, 284  
 pop\_back, 283  
 push\_back, 283  
 push\_front, 284  
 reserve, 284  
 size, 284  
 sizeof\_T, 284  
 oscl\_vsnprintf  
 osclutil, 77, 79  
**oscl\_wchar**  
 osclbase, 32  
**OSCL\_wFastString**, 285  
 OSCL\_wFastString, 286  
**OSCL\_wFastString**  
 ~OSCL\_wFastString, 286  
 chartype, 285  
 get\_cstr, 286  
 get\_maxsize, 286  
 get\_size, 286  
 get\_str, 286  
 operator=, 286  
 OSCL\_wFastString, 286  
 OSCL\_wString, 286  
 set, 286

set\_length, 286  
**OSCL\_wHeapString**, 288  
     osclutil, 79  
**OSCL\_wHeapString**  
     chartype, 288  
     **OSCL\_wString**, 289  
**OSCL\_wHeapStringA**, 290  
     **OSCL\_wHeapStringA**, 291  
**OSCL\_wHeapStringA**  
     ~**OSCL\_wHeapStringA**, 291  
     chartype, 291  
     get\_cstr, 291  
     get\_maxsize, 291  
     get\_size, 291  
     get\_str, 291  
     operator=, 291, 292  
     **OSCL\_wHeapStringA**, 291  
     **OSCL\_wString**, 292  
     set, 292  
**OSCL\_WRITESET\_FLAG**  
     oscl\_socket\_serv\_imp\_pv.h, 748  
**OSCL\_wStackString**, 293  
     osclutil, 79  
**OSCL\_wStackString**  
     chartype, 294  
     **OSCL\_wString**, 294  
**OSCL\_wString**, 295  
     **OSCL\_wFastString**, 286  
     **OSCL\_wHeapString**, 289  
     **OSCL\_wHeapStringA**, 292  
     **OSCL\_wStackString**, 294  
     **OSCL\_wString**, 296  
**OSCL\_wString**  
     ~**OSCL\_wString**, 296  
     append\_rep, 296  
     chartype, 296  
     get\_cstr, 296  
     get\_maxsize, 296  
     get\_size, 296  
     get\_str, 296  
     hash, 296  
     is\_writable, 297  
     operator!=, 297  
     operator+=, 297  
     operator<, 297  
     operator<=, 297  
     operator=, 297  
     operator==, 297  
     operator>, 297  
     operator>=, 297  
     operator[], 297  
     **OSCL\_wString**, 296  
     read, 297  
     set\_len, 298  
     set\_rep, 298  
     write, 298  
**OSCL\_ZEROIZE**  
     osclproc, 101  
**OsclAccept**  
     osclconfig\_io.h, 793  
**OsclAcceptMethod**, 299  
**OsclAcceptMethod**  
     ~**OsclAcceptMethod**, 299  
     Accept, 299  
     AcceptRequest, 299  
     DiscardAcceptedSocket, 299  
     GetAcceptedSocket, 299  
     NewL, 299  
**OsclAcceptRequest**, 300  
     OsclAcceptRequest, 300  
     OsclSocketI, 524  
**OsclAcceptRequest**  
     Accept, 300  
     OsclAcceptRequest, 300  
**OsclActiveObject**, 301  
     EPriorityHigh, 302  
     EPriorityHighest, 302  
     EPriorityIdle, 302  
     EPriorityLow, 302  
     EPriorityNominal, 302  
     OsclActiveObject, 302  
     OsclExecSchedulerCommonBase, 388  
     PVActiveBase, 596  
     PVActiveStats, 597  
     PVThreadContext, 616  
**OsclActiveObject**  
     ~**OsclActiveObject**, 302  
     AddToScheduler, 302  
     Cancel, 302  
     DoCancel, 303  
     IsBusy, 303  
     OsclActiveObject, 302  
     OsclActivePriority, 302  
     PendComplete, 303  
     PendForExec, 303  
     Priority, 303  
     RemoveFromScheduler, 303  
     RunError, 303  
     RunIfNotReady, 304  
     SetBusy, 304  
     SetStatus, 304  
     Status, 304  
     StatusRef, 304  
**OsclActivePriority**  
     OsclActiveObject, 302  
**OsclAllocDestructDealloc**, 305  
**OsclAllocDestructDealloc**  
     ~**OsclAllocDestructDealloc**, 305

OsclAny  
    osclbase, 32  
OsclAOStatus, 306  
    OsclAOStatus, 306  
OsclAOStatus  
    operator!=, 306  
    operator<, 306  
    operator<=, 306  
    operator=, 306  
    operator==, 306  
    operator>, 306  
    operator>=, 306  
    OsclAOStatus, 306  
    Value, 306  
OsclAsyncFile, 307  
OsclAsyncFile  
    ~OsclAsyncFile, 308  
    Close, 308  
    Delete, 308  
    EndOfFile, 308  
    Flush, 308  
    iNumOfRun, 309  
    iNumOfRunErr, 309  
    NewL, 308  
    Open, 308, 309  
    Read, 309  
    Seek, 309  
    Size, 309  
    Tell, 309  
    Write, 309  
OsclAsyncFileBuffer, 310  
OsclAsyncFileBuffer  
    ~OsclAsyncFileBuffer, 311  
    Buffer, 311  
    CleanInUse, 311  
    HasThisOffset, 311  
    Id, 311  
    IsInUse, 311  
    IsValid, 311  
    Length, 311  
    NewL, 311  
    Offset, 311  
    SetInUse, 311  
    SetOffset, 311  
    StartAsyncRead, 311  
    UpdateData, 311  
OsclAuditCB, 312  
    OsclAuditCB, 312  
OsclAuditCB  
    OsclAuditCB, 312  
    pAudit, 312  
    pStatsNode, 312  
OsclBase  
    OsclSingletonRegistry, 520  
OsclTLSRegistry, 579  
osclbase  
    \_OSCL\_Abort, 33  
    ALLOC\_AND\_CONSTRUCT, 30  
    ALLOCATE, 30  
    big\_endian\_to\_host, 33  
    Bind, 33  
    c\_bool, 32  
    CTIME\_BUFFER\_SIZE, 43  
    CtimeStrBuf, 32  
    host\_to\_big\_endian, 33  
    host\_to\_little\_endian, 34  
    int64, 32  
    little\_endian\_to\_host, 34  
    mbchar, 32  
    MICROSECONDS, 33  
    MILLISECONDS, 33  
    MSEC\_PER\_SEC, 43  
    NULL, 30  
    NULL\_TERM\_CHAR, 30  
    octet, 32  
    operator-, 34  
    operator==, 34  
    OSCL\_ABS, 30  
    OSCL\_ASSERT, 30  
    OSCL\_Assert, 34  
    OSCL\_BEGIN\_PACKED, 30  
    oscl\_CIstrcmp, 34, 35  
    oscl\_CIstrncmp, 35  
    OSCL\_COND\_EXPORT\_REF, 30  
    OSCL\_COND\_IMPORT\_REF, 30  
    OSCL\_CONST\_CAST, 30  
    OSCL\_DISABLE\_WARNING\_-  
        RETURN\_TYPE\_NOT\_UDT, 30  
    OSCL\_DISABLE\_WARNING\_-  
        TRUNCATE\_DEBUG\_MESSAGE,  
            30  
    OSCL\_DLL\_ENTRY\_POINT, 30  
    OSCL\_DLL\_ENTRY\_POINT\_DEFAULT,  
        31  
    OSCL\_DYNAMIC\_CAST, 31  
    OSCL\_END\_PACKED, 31  
    OSCL\_HAS\_SINGLETON\_SUPPORT, 31  
    OSCL\_INLINE, 31  
    oscl\_isLetter, 35  
    OSCL\_MAX, 31  
    OSCL\_MIN, 31  
    OSCL\_PACKED\_VAR, 31  
    OSCL\_REINTERPRET\_CAST, 31  
    OSCL\_STATIC\_CAST, 31  
    oscl\_strcat, 36  
    oscl\_strchr, 36, 37  
    oscl\_stremp, 37  
    oscl\_strlen, 37

oscl\_strncat, 38  
 oscl\_strcmp, 38, 39  
 oscl\_memcpy, 39  
 oscl strrchr, 40  
 oscl\_strset, 40  
 oscl strstr, 40, 41  
 OSCL\_TCHAR, 32  
 OSCL\_TEMPLATED\_DESTRUCTOR\_-  
     CALL, 31  
 OSCL\_TLS\_BASE\_SLOTS, 31  
 OSCL\_TLS\_EXTERNAL\_SLOTS, 31  
 OSCL\_TLS\_ID\_BASE\_LAST, 43  
 OSCL\_TLS\_ID\_ERRORHOOK, 43  
 OSCL\_TLS\_ID\_MAGICNUM, 43  
 OSCL\_TLS\_ID\_OSCLREGISTRY, 43  
 OSCL\_TLS\_ID\_PAYLOADPARSER, 43  
 OSCL\_TLS\_ID\_PVERRORTRAP, 43  
 OSCL\_TLS\_ID\_PVLOGGER, 43  
 OSCL\_TLS\_ID\_PVMFRECOGNIZER, 43  
 OSCL\_TLS\_ID\_PVSCHEDULER, 43  
 OSCL\_TLS\_ID\_SDPMEDIAPARSER, 43  
 OSCL\_TLS\_ID\_SQLITE3, 43  
 OSCL\_TLS\_ID\_TEST, 43  
 OSCL\_TLS\_ID\_WMDRM, 43  
 OSCL\_TLS\_MAX\_SLOTS, 31  
 oscl\_tolower, 41  
 OSCL\_UNSIGNED\_CONST, 31  
 OSCL\_UNUSED\_ARG, 31  
 OSCL\_UNUSED\_RETURN, 32  
 oscl\_wchar, 32  
 OsclAny, 32  
 OsclFloat, 32  
 PV8601TIME\_BUFFER\_SIZE, 43  
 PV8601timeStrBuf, 32  
 PV8601ToRFC822, 41  
 PVMEM\_INST\_LEVEL, 32  
 PVOsclBase\_Cleanup, 42  
 PVOsclBase\_Init, 42  
 RFC822ToPV8601, 42  
 SECONDS, 33  
 TimeUnits, 33  
 TOscITlsKey, 33  
 uint, 33  
 uint64, 33  
 unix\_ntp\_offset, 43  
 USEC\_PER\_SEC, 43  
 OsclBasicDateTimeStruct  
     osclconfig\_time.h, 813  
 OsclBasicTimeStruct  
     osclconfig\_time.h, 813  
 OsclBind  
     osclconfig\_io.h, 793  
 OsclBindMethod, 313  
 OsclBindMethod  
     ~OsclBindMethod, 313  
     Bind, 313  
     BindRequest, 313  
     NewL, 313  
 OsclBindRequest, 314  
     OsclBindRequest, 314  
 OsclBindRequest  
     Bind, 314  
     OsclBindRequest, 314  
 OsclBinIStream, 315  
     OsclBinIStream, 315  
 OsclBinIStream  
     ~OsclBinIStream, 315  
     get, 315  
     OsclBinIStream, 315  
     Read\_uint8, 315  
 OsclBinIStreamBigEndian, 317  
     OsclBinIStreamBigEndian, 318  
 OsclBinIStreamBigEndian  
     operator>>, 318  
     OsclBinIStreamBigEndian, 318  
     Read, 318  
     Read\_uint16, 318  
     Read\_uint32, 318  
 OsclBinIStreamLittleEndian, 320  
     OsclBinIStreamLittleEndian, 321  
 OsclBinIStreamLittleEndian  
     operator>>, 321  
     OsclBinIStreamLittleEndian, 321  
     Read\_uint16, 321  
     Read\_uint32, 321  
 OsclBinOStream, 322  
     OsclBinOStream, 322  
 OsclBinOStream  
     ~OsclBinOStream, 322  
     OsclBinOStream, 322  
     write, 322  
 OsclBinOStreamBigEndian, 323  
     OsclBinOStreamBigEndian, 324  
 OsclBinOStreamBigEndian  
     operator<<, 324  
     OsclBinOStreamBigEndian, 324  
     WriteUnsignedLong, 324  
     WriteUnsignedShort, 324  
 OsclBinOStreamLittleEndian, 325  
     OsclBinOStreamLittleEndian, 326  
 OsclBinOStreamLittleEndian  
     operator<<, 326  
     OsclBinOStreamLittleEndian, 326  
     WriteUnsignedLong, 326  
     WriteUnsignedShort, 326  
 OsclBinStream, 327  
     EOF\_STATE, 328  
     FAIL\_STATE, 328

GOOD\_STATE, 328  
 OsclBinStream, 328  
**OsclBinStream**  
 Attach, 328  
 eof, 328  
 fail, 329  
 firstFragPtr, 330  
 fragsLeft, 330  
 good, 329  
 HaveRoomInCurrentBlock, 329  
 length, 330  
 nextFragPtr, 330  
 numFrags, 330  
 OsclBinStream, 328  
 pBasePosition, 330  
 PositionInBlock, 329  
 pPosition, 330  
 ReserveSpace, 329  
 Seek, 329  
 seekFromCurrentPosition, 329  
 specialFragBuffer, 330  
 state, 330  
 state\_t, 328  
 tellg, 329  
**OsclBuf**, 331  
 OsclBuf, 332  
**OsclBuf**  
 Delete, 332  
 Des, 332  
 Desc, 332  
 iBuffer, 332  
 iLength, 332  
 iMaxLength, 332  
 Length, 332  
 NewL, 332  
 OsclBuf, 332  
**OsclCloseSocket**  
 osclconfig\_io.h, 794  
**OsclCoeActiveScheduler**  
 OsclExecSchedulerBase, 382  
 OsclExecSchedulerCommonBase, 388  
 PVThreadContext, 616  
**OsclCoeActiveSchedulerBase**  
 PVThreadContext, 616  
**OsclCompareLess**, 333  
**OsclCompareLess**  
 compare, 333  
**OsclComponentFactory**  
 osclutil, 66  
**OsclComponentRegistry**, 334  
 OsclComponentRegistry, 335  
**OsclComponentRegistry**  
 ~OsclComponentRegistry, 335  
 CloseSession, 335  
 FindExact, 335  
 FindHierarchical, 335  
 iComponentIdCounter, 335  
 iData, 335  
 iMutex, 335  
 iNumSessions, 335  
 OpenSession, 335  
 OsclComponentRegistry, 335  
 Register, 335  
 Unregister, 335  
 OsclComponentRegistryData, 336  
**OsclComponentRegistryData**  
 Find, 336  
 iVec, 336  
**OsclComponentRegistryElement**, 337  
**OsclComponentRegistryElement**, 337  
**OsclComponentRegistryElement**  
 ~OsclComponentRegistryElement, 337  
 iComponentId, 337  
 iFactory, 337  
 iId, 337  
 Match, 337  
 operator=, 337  
 OsclComponentRegistryElement, 337  
**osclconfig**  
 \_\_int16\_check\_\_, 22  
 \_\_int32\_check\_\_, 22  
 \_\_int8\_check\_\_, 22  
 \_\_uint16\_check\_\_, 22  
 \_\_uint32\_check\_\_, 22  
 \_\_uint8\_check\_\_, 22  
 OSCL\_ASSERT\_ALWAYS, 20  
 OSCL\_BYTE\_ORDER\_BIG\_ENDIAN, 20  
 OSCL\_BYTE\_ORDER\_LITTLE\_ENDIAN, 20  
 OSCL\_HAS\_BERKELEY\_SOCKETS, 20  
 OSCL\_HAS\_MSWIN\_PARTIAL\_SUPPORT, 21  
 OSCL\_HAS\_MSWIN\_SUPPORT, 21  
 OSCL\_HAS\_PTHREAD\_SUPPORT, 21  
 OSCL\_HAS\_PV\_C\_OS\_API\_MEMORY\_FUNCS, 21  
 OSCL\_HAS\_PV\_C\_OS\_SUPPORT, 21  
 OSCL\_HAS\_PV\_C\_OS\_TIME\_FUNCS, 21  
 OSCL\_HAS\_SAVAJE\_IO\_SUPPORT, 21  
 OSCL\_HAS\_SAVAJE\_SUPPORT, 21  
 OSCL\_HAS\_SEM\_TIMEDWAIT\_SUPPORT, 21  
 OSCL\_HAS\_SYMBIAN\_COMPATIBLE\_IO\_FUNCTION, 21

OSCL\_HAS\_SYMBIAN\_DNS\_SERVER,  
   21  
 OSCL\_HAS\_SYMBIAN\_ERRORTRAP,  
   21  
 OSCL\_HAS\_SYMBIAN\_MATH, 21  
 OSCL\_HAS\_SYMBIAN\_MEMORY\_-  
   FUNCS, 21  
 OSCL\_HAS\_SYMBIAN\_SCHEDULER,  
   21  
 OSCL\_HAS\_SYMBIAN\_SOCKET\_-  
   SERVER, 21  
 OSCL\_HAS\_SYMBIAN\_SUPPORT, 21  
 OSCL\_HAS\_SYMBIAN\_TIMERS, 21  
 OSCL\_HAS\_UNIX\_SUPPORT, 21  
 OSCL\_HAS\_UNIX\_TIME\_FUNCS, 21  
 OSCL\_INTEGERS\_WORD\_ALIGNED,  
   21  
 osclconfig.h, 782  
   \_\_TFS\_\_, 783  
 OSCL\_BEGIN\_PACKED, 783  
 OSCL\_END\_PACKED, 783  
 OSCL\_HAS\_ANDROID\_FILE\_IO\_-  
   SUPPORT, 783  
 OSCL\_HAS\_ANDROID\_SUPPORT, 783  
 OSCL\_NATIVE\_UINT64\_TYPE, 783  
 OSCL\_PACKED\_STRUCT\_BEGIN, 783  
 OSCL\_PACKED\_STRUCT\_END, 783  
 OSCL\_PACKED\_VAR, 783  
 OSCL\_RELEASE\_BUILD, 783  
 OSCL\_TEMPLATED\_DESTRUCTOR\_-  
   CALL, 783  
 OSCL\_UNSIGNED\_CONST, 783  
 osclconfig\_ansi\_memory.h, 784  
   OSCL\_HAS\_ANSI\_MEMORY\_FUNCS,  
     784  
   oscl\_memsize\_t, 784  
 osclconfig\_check.h, 785  
 osclconfig\_compiler\_warnings.h, 786  
   OSCL\_FUNCTION\_PTR, 786  
 osclconfig\_error.h, 787  
   OSCL\_HAS\_ERRNO\_H, 787  
   OSCL\_HAS\_EXCEPTIONS, 787  
   OSCL\_HAS\_SETJMP\_H, 787  
   OSCL\_HAS\_SYMBIAN\_ERRORTRAP,  
     787  
 osclconfig\_error\_check.h, 788  
 osclconfig\_global\_new\_delete.h, 789  
 osclconfig\_global\_placement\_new.h, 790  
   operator new, 790  
 osclconfig\_io.h, 791  
   \_\_FILE\_OFFSET\_BITS, 793  
   OSCL\_AF\_INET, 793  
   OSCL\_FILE\_BUFFER\_MAX\_SIZE, 793  
 OSCL\_HAS\_ANSI\_64BIT\_FILE\_IO\_-  
   SUPPORT, 793  
 OSCL\_HAS\_ANSI\_FILE\_IO\_SUPPORT,  
   793  
 OSCL\_HAS\_BERKELEY\_SOCKETS,  
   793  
 OSCL\_HAS\_GLOB, 793  
 OSCL\_HAS\_LARGE\_FILE\_SUPPORT,  
   793  
 OSCL\_HAS\_MSWIN\_FILE\_IO\_-  
   SUPPORT, 793  
 OSCL\_HAS\_NATIVE\_FILE\_CACHE\_-  
   ENABLE, 793  
 OSCL\_HAS\_PV\_FILE\_CACHE, 793  
 OSCL\_HAS\_SOCKET\_SUPPORT, 793  
 OSCL\_HAS\_SYMBIAN\_-  
   COMPATIBLE\_IO\_FUNCTION,  
   793  
 OSCL\_HAS\_SYMBIAN\_DNS\_SERVER,  
   793  
 OSCL\_HAS\_SYMBIAN\_SOCKET\_-  
   SERVER, 793  
 OSCL IPPROTO\_TCP, 793  
 OSCL IPPROTO\_UDP, 793  
 OSCL\_SD\_BOTH, 793  
 OSCL\_SD\_RECEIVE, 793  
 OSCL\_SD\_SEND, 793  
 OSCL SOCK\_DGRAM, 793  
 OSCL SOCK\_STREAM, 793  
 OsclAccept, 793  
 OsclBind, 793  
 OsclCloseSocket, 794  
 OsclConnect, 794  
 OsclConnectComplete, 794  
 OsclGetAsyncSockErr, 794  
 OsclGetDottedAddr, 794  
 OsclGethostname, 794  
 OsclJoin, 795  
 OsclListen, 795  
 OsclMakeSockAddr, 795  
 OsclRecv, 795  
 OsclRecvFrom, 795  
 OsclSend, 796  
 OsclSendTo, 796  
 OsclSetNonBlocking, 796  
 OsclSetRecvBufferSize, 796  
 OsclShutdown, 796  
 OsclSocket, 796  
 OsclSocketCleanup, 797  
 OsclSocketSelect, 797  
 OsclSocketStartup, 797  
 OsclUnmakeSockAddr, 797  
 OsclValidInetAddr, 797  
 TOsclFileOffset, 797

TOsclHostent, 797  
 TOsclSockAddr, 797  
 TOsclSockAddrLen, 797  
 TOsclSocket, 797  
 osclconfig\_io\_check.h, 798  
   \_\_verify\_TOsclFileOffset\_defined\_\_,  
   798  
 osclconfig\_ix86.h, 799  
 osclconfig\_lib.h, 800  
   OSCL\_HAS\_RUNTIME\_LIB\_-  
     LOADING\_SUPPORT, 800  
   OSCL\_LIB\_READ\_DEBUG\_LIBS, 800  
   PV\_DYNAMIC\_LOADING\_CONFIG\_-  
     FILE\_PATH, 800  
   PV\_RUNTIME\_LIB\_FILENAME\_-  
     EXTENSION, 800  
 osclconfig\_lib\_check.h, 801  
 osclconfig\_limits\_typedefs.h, 802  
   OSCL\_CHAR\_IS\_SIGNED, 802  
   OSCL\_CHAR\_IS\_UNSIGNED, 802  
 osclconfig\_memory.h, 803  
   OSCL\_BYPASS\_MEMMGT, 803  
   OSCL\_HAS\_GLOBAL\_NEW\_DELETE,  
     803  
   OSCL\_HAS\_HEAP\_BASE\_SUPPORT,  
     803  
   OSCL\_HAS\_SYMBIAN\_MEMORY\_-  
     FUNCS, 803  
   PVMEM\_INST\_LEVEL, 803  
 osclconfig\_memory\_check.h, 804  
 osclconfig\_no\_os.h, 805  
 osclconfig\_proc.h, 806  
 osclconfig\_proc\_check.h, 807  
   \_\_verify\_TOsclConditionObject\_\_-  
     defined\_\_, 807  
   \_\_verify\_TOsclMutexObject\_\_defined\_\_,  
     807  
   \_\_verify\_TOsclSemaphoreObject\_\_-  
     defined\_\_, 807  
   \_\_verify\_TOsclThreadFuncArg\_\_-  
     defined\_\_, 807  
   \_\_verify\_TOsclThreadFuncRet\_\_-  
     defined\_\_, 807  
   \_\_verify\_TOsclThreadId\_\_defined\_\_, 807  
   \_\_verify\_TOsclThreadObject\_\_defined\_\_-,  
     807  
 osclconfig\_proc\_unix\_android.h, 809  
   OSCL\_HAS\_NON\_PREEMPTIVE\_-  
     THREAD\_SUPPORT, 810  
   OSCL\_HAS\_PTHREAD\_SUPPORT, 810  
   OSCL\_HAS\_SEM\_TIMEDWAIT\_-  
     SUPPORT, 810  
   OSCL\_HAS\_SYMBIAN\_SCHEDULER,  
     810  
   OSCL\_HAS\_THREAD\_SUPPORT, 810  
   OSCL\_THREAD\_DECL, 810  
   TOsclConditionObject, 810  
   TOsclMutexObject, 810  
   TOsclSemaphoreObject, 810  
   TOsclThreadFuncArg, 810  
   TOsclThreadFuncRet, 810  
   TOsclThreadId, 810  
   TOsclThreadObject, 810  
   osclconfig\_proc\_unix\_common.h, 811  
     OSCL\_HAS\_NON\_PREEMPTIVE\_-  
       THREAD\_SUPPORT, 812  
   OSCL\_HAS\_PTHREAD\_SUPPORT, 812  
   OSCL\_HAS\_SEM\_TIMEDWAIT\_-  
     SUPPORT, 812  
   OSCL\_HAS\_SYMBIAN\_SCHEDULER,  
     812  
   OSCL\_HAS\_THREAD\_SUPPORT, 812  
   OSCL\_THREAD\_DECL, 812  
   TOsclConditionObject, 812  
   TOsclMutexObject, 812  
   TOsclSemaphoreObject, 812  
   TOsclThreadFuncArg, 812  
   TOsclThreadFuncRet, 812  
   TOsclThreadId, 812  
   TOsclThreadObject, 812  
 osclconfig\_time.h, 813  
   OSCL\_HAS\_UNIX\_TIME\_FUNCS, 813  
   OsclBasicDateStruct, 813  
   OsclBasicTimeStruct, 813  
 osclconfig\_time\_check.h, 814  
   \_\_Validate\_BasicTimeDateStruct\_\_, 814  
   \_\_Validate\_BasicTimeStruct\_\_, 814  
 osclconfig\_unix\_android.h, 815  
   \_STRLIT, 818  
   \_STRLIT\_CHAR, 818  
   \_STRLIT\_WCHAR, 818  
   INT64, 818  
   INT64\_HILO, 818  
   OSCL\_DISABLE\_INLINES, 818  
   OSCL\_HAS\_ANSI\_MATH\_SUPPORT,  
     818  
   OSCL\_HAS\_ANSI\_STDIO\_SUPPORT,  
     818  
   OSCL\_HAS\_ANSI\_STDLIB\_SUPPORT,  
     818  
   OSCL\_HAS\_ANSI\_STRING\_SUPPORT,  
     818  
   OSCL\_HAS\_ANSI\_WIDE\_STRING\_-  
     SUPPORT, 818  
   OSCL\_HAS\_BASIC\_LOCK, 818  
   OSCL\_HAS\_GLOBAL\_VARIABLE\_-  
     SUPPORT, 818  
   OSCL\_HAS\_MSWIN\_SUPPORT, 818

OSCL\_HAS\_NATIVE\_INT64\_TYPE, 818  
 OSCL\_HAS\_NATIVE\_UINT64\_TYPE,  
   818  
 OSCL\_HAS\_SYMBIAN\_SUPPORT, 818  
 OSCL\_HAS\_TLS\_SUPPORT, 818  
 OSCL\_HAS\_UNICODE\_SUPPORT, 818  
 OSCL\_HAS\_UNIX\_SUPPORT, 818  
 OSCL\_MEMFRAG\_PTR\_BEFORE\_LEN,  
   818  
 OSCL\_NATIVE\_INT64\_TYPE, 818  
 OSCL\_NATIVE\_UINT64\_TYPE, 818  
 OSCL\_NATIVE\_WCHAR\_TYPE, 818  
 OSCL\_TLS\_GET\_FUNC, 818  
 OSCL\_TLS\_IS\_KEYED, 818  
 OSCL\_TLS\_KEY\_CREATE\_FUNC, 818  
 OSCL\_TLS\_KEY\_DELETE\_FUNC, 818  
 OSCL\_TLS\_STORE\_FUNC, 818  
 TOsclBasicLockObject, 818  
 TOsclTlsKey, 818  
 UINT64, 818  
 UINT64\_HILO, 818  
 osclconfig\_unix\_common.h, 819  
   \_STRLIT, 822  
   \_STRLIT\_CHAR, 822  
   \_STRLIT\_WCHAR, 822  
 INT64, 822  
 INT64\_HILO, 822  
 OSCL\_DISABLE\_INLINES, 822  
 OSCL\_HAS\_ANSI\_MATH\_SUPPORT,  
   822  
 OSCL\_HAS\_ANSI\_STDIO\_SUPPORT,  
   822  
 OSCL\_HAS\_ANSI\_STDLIB\_SUPPORT,  
   822  
 OSCL\_HAS\_ANSI\_STRING\_SUPPORT,  
   822  
 OSCL\_HAS\_ANSI\_WIDE\_STRING\_-  
   SUPPORT, 822  
 OSCL\_HAS\_BASIC\_LOCK, 822  
 OSCL\_HAS\_GLOBAL\_VARIABLE\_-  
   SUPPORT, 822  
 OSCL\_HAS\_MSWIN\_SUPPORT, 822  
 OSCL\_HAS\_NATIVE\_INT64\_TYPE, 822  
 OSCL\_HAS\_NATIVE\_UINT64\_TYPE,  
   822  
 OSCL\_HAS\_SYMBIAN\_SUPPORT, 822  
 OSCL\_HAS\_TLS\_SUPPORT, 822  
 OSCL\_HAS\_UNICODE\_SUPPORT, 822  
 OSCL\_HAS\_UNIX\_SUPPORT, 822  
 OSCL\_MEMFRAG\_PTR\_BEFORE\_LEN,  
   822  
 OSCL\_NATIVE\_INT64\_TYPE, 822  
 OSCL\_NATIVE\_UINT64\_TYPE, 822  
 OSCL\_NATIVE\_WCHAR\_TYPE, 822  
 OSCL\_TLS\_GET\_FUNC, 822  
 OSCL\_TLS\_IS\_KEYED, 822  
 OSCL\_TLS\_KEY\_CREATE\_FUNC, 822  
 OSCL\_TLS\_KEY\_DELETE\_FUNC, 822  
 OSCL\_TLS\_STORE\_FUNC, 822  
 TOsclBasicLockObject, 822  
 TOsclTlsKey, 822  
 UINT64, 822  
 UINT64\_HILO, 822  
 osclconfig\_util.h, 823  
   OSCL\_CLOCK\_HAS\_DRIFT\_-  
     CORRECTION, 823  
 OSCL\_HAS\_SYMBIAN\_MATH, 823  
 OSCL\_HAS\_SYMBIAN\_TIMERS, 823  
 OSCL RAND\_MAX, 823  
 SLEEP\_ONE\_SEC, 823  
 osclconfig\_util\_check.h, 824  
 OsclConnect  
   osclconfig\_io.h, 794  
 OsclConnectComplete  
   osclconfig\_io.h, 794  
 OsclConnectMethod, 339  
 OsclConnectMethod  
   ~OsclConnectMethod, 339  
   Connect, 339  
   ConnectRequest, 339  
   NewL, 339  
 OsclConnectRequest, 340  
   OsclConnectRequest, 340  
   OsclSocketI, 524  
 OsclConnectRequest  
   Connect, 340  
   OsclConnectRequest, 340  
 OsclDestructDealloc, 341  
 OsclDestructDealloc  
   destruct\_and\_dealloc, 341  
 OsclDNS, 342  
   OsclSocketServ, 541  
 OsclDNS  
   ~OsclDNS, 342  
   CancelGetHostByName, 342  
   GetHostByName, 343  
   NewL, 343  
   OsclDNSRequestAO, 343  
 OsclDNSI, 344  
   OsclDNSRequestAO, 356  
   OsclSocketServI, 543  
 OsclDNSI  
   ~OsclDNSI, 344  
   Close, 344  
   DNSRequestParam, 345  
   GetHostByName, 344  
   GetHostByNameSuccess, 344  
   NewL, 345

Open, 345  
 OsclDNSRequest, 345  
**OsclDNSIBase**, 346  
 OsclDNSIBase, 347  
**OsclDNSIBase**  
 ~OsclDNSIBase, 347  
 CancelFxn, 347  
 CancelGetHostByName, 347  
 Close, 347  
 GetHostName, 347  
 GetHostNameSuccess, 347  
 iAlloc, 347  
 iSocketServ, 347  
 IsReady, 347  
 Open, 347  
 OsclDNSIBase, 347  
 OsclDNSRequest, 347  
 OsclGetHostByNameRequest, 347  
**OsclDNSMethod**, 349  
 OsclDNSMethod, 350  
 OsclDNSRequestAO, 356  
**OsclDNSMethod**  
 Abort, 350  
 AbortAll, 350  
 CancelMethod, 350  
 ConstructL, 350  
 iAlloc, 351  
 iDNSFxn, 351  
 iDNSObserver, 351  
 iDNSRequestAO, 351  
 iId, 351  
 iLogger, 351  
 MethodDone, 350  
 OsclDNSMethod, 350  
 Run, 350  
 StartMethod, 350  
**OsclDNSObserver**, 352  
**OsclDNSObserver**  
 ~OsclDNSObserver, 352  
 HandleDNSEvent, 352  
**OsclDNSRequest**, 353  
 OsclDNSI, 345  
 OsclDNSIBase, 347  
 OsclDNSRequest, 353  
 OsclDNSRequestAO, 356  
**OsclDNSRequest**  
 ~OsclDNSRequest, 353  
 Activate, 353  
 CancelRequest, 353  
 Complete, 353  
 iActive, 353  
 iDNSRequestAO, 353  
 iDNSRequestParam, 353  
 OsclDNSRequest, 353  
 OsclDNSRequestAO, 354  
 OsclDNS, 343  
 OsclDNSRequestAO, 355  
**OsclDNSRequestAO**  
 Abort, 355  
 ConstructL, 355  
 DNSRequestParam, 356  
 DoCancel, 355  
 GetSocketError, 355  
 iDNSI, 356  
 iDNSMethod, 356  
 iLogger, 356  
 iSocketError, 356  
 NewRequest, 355  
 OsclDNSI, 356  
 OsclDNSMethod, 356  
 OsclDNSRequest, 356  
 OsclDNSRequestAO, 355  
 RequestDone, 355  
 Run, 355  
 Serv, 355  
 Success, 356  
**OsclDoubleLink**, 357  
 OsclDoubleLink, 357  
**OsclDoubleLink**  
 iNext, 357  
 InsertAfter, 357  
 InsertBefore, 357  
 iPrev, 357  
 OsclDoubleLink, 357  
 Remove, 357  
**OsclDoubleList**, 358  
 OsclDoubleList, 358  
**OsclDoubleList**  
 Head, 358  
 InsertHead, 358  
 InsertTail, 358  
 IsHead, 358  
 IsTail, 358  
 OsclDoubleList, 358  
 Tail, 358  
**OsclDoubleListBase**, 359  
 OsclDoubleListBase, 360  
**OsclDoubleListBase**  
 getHead, 360  
 getOffset, 360  
 iHead, 360  
 Insert, 360  
 InsertHead, 360  
 InsertTail, 360  
 iOffset, 360  
 IsEmpty, 360  
 OsclDoubleListBase, 360  
 Reset, 360

SetOffset, 360  
**OsclDoubleRunner**, 361  
   OsclDoubleRunner, 361  
**OsclDoubleRunner**  
   iHead, 361  
   iNext, 361  
   iOffset, 361  
   operator T \*, 361  
   operator++, 361  
   operator--, 361  
   OsclDoubleRunner, 361  
   Set, 361  
   SetToHead, 361  
   SetToTail, 361  
**OsclErrAlreadyExists**  
   osclerror, 88  
**OsclErrAlreadyInstalled**  
   osclerror, 88  
**OsclErrArgument**  
   osclerror, 88  
**OsclErrBadHandle**  
   osclerror, 88  
**OsclErrBusy**  
   osclerror, 88  
**OsclErrCancelled**  
   osclerror, 88  
**OsclErrCorrupt**  
   osclerror, 88  
**OsclErrGeneral**  
   osclerror, 88  
**OsclErrInvalidState**  
   osclerror, 88  
**OsclErrNoHandler**  
   osclerror, 88  
**OsclErrNoMemory**  
   osclerror, 88  
**OsclErrNone**  
   osclerror, 88  
**OsclErrNoResources**  
   osclerror, 88  
**OsclErrNotInstalled**  
   osclerror, 88  
**OsclErrNotReady**  
   osclerror, 88  
**OsclErrNotSupported**  
   osclerror, 88  
**OsclError**, 363  
   OsclErrorTrapImp, 369  
   OsclExecSchedulerCommonBase, 388  
   OsclTrapStack, 582  
**OsclError**  
   Leave, 363  
   LeaveIfError, 363  
   LeaveIfNull, 363  
   Pop, 363  
   PopDealloc, 363, 364  
   PushL, 364  
**osclerror**  
   \_PV\_TRAP, 84  
   \_PV\_TRAP\_NO\_TLS, 84  
   internalLeave, 84  
   OSCL\_BAD\_ALLOC\_EXCEPTION\_-  
    CODE, 84  
   OSCL\_CATCH, 84  
   OSCL\_CATCH\_ANY, 85  
   OSCL\_ERR\_NONE, 85  
   OSCL\_FIRST\_CATCH, 85  
   OSCL\_FIRST\_CATCH\_ANY, 85  
   OSCL\_GetLastError, 89  
   OSCL\_IsErrnoSupported, 89  
   OSCL\_JUMP\_MAX\_JUMP\_MARKS, 85  
   OSCL\_LAST\_CATCH, 85  
   OSCL\_LEAVE, 86  
   OSCL\_MAX\_TRAP\_LEVELS, 86  
   OSCL\_SetLastError, 89  
   OSCL\_StrError, 89  
   OSCL\_TRAPSTACK\_POP, 86  
   OSCL\_TRAPSTACK\_POPDEALLOC, 86  
   OSCL\_TRAPSTACK\_PUSH, 86  
   OSCL\_TRY, 86  
   OSCL\_TRY\_NO\_TLS, 86  
   OsclErrAlreadyExists, 88  
   OsclErrAlreadyInstalled, 88  
   OsclErrArgument, 88  
   OsclErrBadHandle, 88  
   OsclErrBusy, 88  
   OsclErrCancelled, 88  
   OsclErrCorrupt, 88  
   OsclErrGeneral, 88  
   OsclErrInvalidState, 88  
   OsclErrNoHandler, 88  
   OsclErrNoMemory, 88  
   OsclErrNone, 88  
   OsclErrNoResources, 88  
   OsclErrNotInstalled, 88  
   OsclErrNotReady, 88  
   OsclErrNotSupported, 88  
   OsclErrOverflow, 88  
   OsclErrSystemCallFailed, 88  
   OsclErrThreadContextIncorrect, 88  
   OsclErrTimeout, 88  
   OsclErrUnderflow, 88  
   OsclFailure, 88  
   OsclLeaveCode, 89  
   OsclPending, 88  
   OsclReturnCode, 89  
   OsclSuccess, 88  
   OsclTrapOperation, 89

PVError\_DoLeave, 88  
 PVERROR\_IMP\_JUMPS, 88  
 PVERRORTRAP\_REGISTRY, 88  
 PVERRORTRAP\_REGISTRY\_ID, 89  
**OsclErrorAllocator**, 365  
     OsclErrorAllocator, 365  
**OsclErrorAllocator**  
     allocate, 365  
     deallocate, 365  
     operator delete, 366  
     operator new, 366  
     OsclErrorAllocator, 365  
**OsclErrorTrap**, 367  
     OsclErrorTrapImp, 369  
     OsclTrapStack, 582  
**OsclErrorTrap**  
     Cleanup, 367  
     GetErrorTrapImp, 367  
     Init, 367  
**OsclErrorTrapImp**, 368  
     OsclJump, 406  
     OsclTrapStack, 582  
**OsclErrorTrapImp**  
     CPVInterfaceProxy, 369  
     iJumpData, 369  
     iLeave, 369  
     iTrapStack, 369  
     OsclError, 369  
     OsclErrorTrap, 369  
     OsclExecScheduler, 369  
     OsclExecSchedulerCommonBase, 369  
     OsclJump, 369  
     OsclJumpMark, 369  
     OsclScheduler, 369  
     OsclTrapStack, 369  
     Trap, 368  
     TrapNoTls, 368  
     UnTrap, 368  
**OsclErrOverflow**  
     osclerror, 88  
**OsclErrSystemCallFailed**  
     osclerror, 88  
**OsclErrThreadContextIncorrect**  
     osclerror, 88  
**OsclErrTimeout**  
     osclerror, 88  
**OsclErrUnderflow**  
     osclerror, 88  
**OsclException**, 370  
     OsclException, 370  
**OsclException**  
     getLeaveCode, 370  
     OsclException, 370  
**OsclExclusiveArrayPtr**, 371  
  
     OsclExclusiveArrayPtr, 372  
     ~OsclExclusiveArrayPtr, 372  
     \_Ptr, 373  
     get, 372  
     operator \*, 372  
     operator->, 372  
     operator=, 372  
     OsclExclusiveArrayPtr, 372  
     release, 373  
     set, 373  
**OsclExclusivePtr**, 374  
     OsclExclusivePtr, 375  
**OsclExclusivePtr**  
     ~OsclExclusivePtr, 375  
     \_Ptr, 376  
     get, 375  
     operator \*, 375  
     operator->, 375  
     operator=, 375  
     OsclExclusivePtr, 375  
     release, 376  
     set, 376  
**OsclExclusivePtrA**, 377  
     OsclExclusivePtrA, 378  
**OsclExclusivePtrA**  
     ~OsclExclusivePtrA, 378  
     \_Ptr, 379  
     get, 378  
     operator \*, 378  
     operator->, 378  
     operator=, 378  
     OsclExclusivePtrA, 378  
     release, 379  
     set, 379  
**OsclExecScheduler**, 380  
     OsclErrorTrapImp, 369  
     OsclExecSchedulerBase, 382  
     OsclExecSchedulerCommonBase, 388  
     PVActiveBase, 596  
     PVActiveStats, 597  
     PVThreadContext, 616  
**OsclExecScheduler**  
     Current, 380  
     OsclScheduler, 381  
     RegisterForCallback, 380  
     RunSchedulerNonBlocking, 380  
**OsclExecSchedulerBase**, 382  
     PVThreadContext, 616  
**OsclExecSchedulerBase**  
     OsclCoeActiveScheduler, 382  
     OsclExecScheduler, 382  
     PVActiveBase, 382  
     OsclExecSchedulerCommonBase, 383

EOOtherExecStats\_Last, 385  
 EOOtherExecStats\_NativeOS, 385  
 EOOtherExecStats\_QueueTime, 385  
 EOOtherExecStats\_ReleaseTime, 385  
 EOOtherExecStats\_WaitTime, 385  
 OsclErrorTrapImp, 369  
 OsclExecSchedulerCommonBase, 386  
 PVActiveStats, 597  
 PVThreadContext, 616  
**OsclExecSchedulerCommonBase**  
   ~OsclExecSchedulerCommonBase, 386  
   AddToExecTimerQ, 386  
   BeginScheduling, 386  
   BeginStats, 386  
   BlockingLoopL, 386  
   CallRunExec, 386  
   CleanupExecQ, 386  
   CleanupStatQ, 386  
   ConstructL, 386  
   ConstructStatQ, 386  
   EndScheduling, 386  
   EndStats, 386  
   Error, 386  
   FindPVBase, 386  
   GetId, 386  
   GetName, 386  
   GetScheduler, 386  
   iAlloc, 390  
   iBlockingMode, 390  
   iDebugLogger, 390  
   iDefAlloc, 390  
   iDelta, 390  
   iDoStop, 390  
   iDoSuspend, 390  
   iErrorTrapImp, 390  
   iExecTimerQ, 390  
   iGrandTotalTicks, 390  
   iLogger, 390  
   iLogPerfIndentStr, 390  
   iLogPerfIndentStrLen, 390  
   iLogPerfTotal, 390  
   iName, 390  
   iNativeMode, 390  
   IncLogPerf, 387  
   InitExecQ, 387  
   InstallScheduler, 387  
   iNumAOAdded, 390  
   iOtherExecStats, 390  
   iPVStatQ, 390  
   iPVStats, 390  
   iReadyQ, 390  
   iResumeSem, 390  
   IsInstalled, 387  
   IsStarted, 387  
   iStopper, 390  
   iStopperCrit, 390  
   iSuspended, 390  
   iThreadContext, 390  
   iTime, 390  
   iTimeCompareThreshold, 390  
   iTTotalPercent, 390  
   iTTotalTicksTemp, 390  
   OsclActiveObject, 388  
   OsclCoeActiveScheduler, 388  
   OsclError, 388  
   OsclExecScheduler, 388  
   OsclExecSchedulerCommonBase, 386  
   OsclReadyQ, 388  
   OsclScheduler, 388  
   OsclTimerCompare, 388  
   OsclTimerObject, 390  
   PendComplete, 387  
   PVActiveBase, 390  
   PVActiveStats, 390  
   PVSchedulerStopper, 390  
   PVThreadContext, 390  
   RequestCanceled, 387  
   ResetLogPerf, 387  
   ResumeScheduler, 387  
   SetScheduler, 387  
   ShowStats, 387  
   ShowSummaryStats, 387  
   StartNativeScheduler, 387  
   StartScheduler, 387  
   StopScheduler, 387  
   SuspendScheduler, 388  
   TOtherExecStats, 385  
   UninstallScheduler, 388  
   UpdateTimers, 388  
   UpdateTimersMsec, 388  
   WaitForReadyAO, 388  
**OsclFailure**  
   osclerror, 88  
**OsclFileCache**, 392  
   **Oscl\_File**, 181  
   OsclFileCache, 393  
**OsclFileCache**  
   ~OsclFileCache, 393  
   \_fixedCaches, 393  
   \_movableCache, 393  
   AddFixedCache, 393  
   Close, 393  
   EndOfFile, 393  
   FileSize, 393  
   Flush, 393  
   Open, 393  
   OsclFileCache, 393  
   OsclFileCacheBuffer, 393

Read, 393  
 Seek, 393  
 Tell, 393  
 Write, 393  
**OsclFileCacheBuffer**, 394  
 Oscl\_File, 181  
**OsclFileCache**, 393  
**OsclFileCacheBuffer**, 395  
**OsclFileCacheBuffer**  
 capacity, 395  
 Contains, 395  
 currentPos, 395  
 endPos, 395  
 filePosition, 395  
 FillFromFile, 395  
 iContainer, 395  
 isFixed, 395  
 IsUpdated, 395  
**OsclFileCacheBuffer**, 395  
 pBuffer, 395  
 Preceeds, 395  
 PrepRead, 395  
 PrepWrite, 395  
 SetPosition, 395  
 updateEnd, 395  
 updateStart, 395  
 usableSize, 395  
 WriteUpdatesToFile, 395  
**OsclFileHandle**, 396  
 OsclFileHandle, 396  
**OsclFileHandle**  
 Handle, 396  
 Oscl\_File, 396  
 OsclFileHandle, 396  
**OsclFileStats**, 397  
 OsclFileStats, 397  
**OsclFileStats**  
 End, 397  
 Log, 397  
 LogAll, 397  
 OsclFileStats, 397  
 Start, 397  
**OsclFileStatsItem**, 398  
**OsclFileStatsItem**  
 iOpCount, 398  
 iParam, 398  
 iParam2, 398  
 iStartTick, 398  
 iTicks, 398  
**OsclFloat**  
 osclbase, 32  
**OsclGetAsyncSockErr**  
 osclconfig\_io.h, 794  
**OsclGetDottedAddr**  
 osclconfig\_io.h, 794  
**OsclGetHostbyname**  
 osclconfig\_io.h, 794  
**OsclGetHostNameMethod**, 399  
 OsclGetHostNameRequest, 400  
**OsclGetHostNameMethod**  
 ~OsclGetHostNameMethod, 399  
 GetHostName, 399  
 NewL, 399  
**OsclGetHostNameRequest**, 400  
 OsclDNSIBase, 347  
**OsclGetHostNameRequest**  
 OsclGetHostNameMethod, 400  
**OsclInit**, 401  
**OsclInit**  
 Cleanup, 401  
 Init, 401  
**OsclInteger64Transport**, 402  
**OsclInteger64Transport**  
 iHigh, 402  
 iLow, 402  
**osclio**  
 EOscFileOp\_Close, 94  
 EOscFileOp\_EndOfFile, 94  
 EOscFileOp\_Flush, 94  
 EOscFileOp\_Last, 95  
 EOscFileOp\_NativeClose, 94  
 EOscFileOp\_NativeEndOfFile, 95  
 EOscFileOp\_NativeFlush, 95  
 EOscFileOp\_NativeOpen, 94  
 EOscFileOp\_NativeRead, 94  
 EOscFileOp\_NativeSeek, 95  
 EOscFileOp\_NativeSize, 95  
 EOscFileOp\_NativeTell, 95  
 EOscFileOp\_NativeWrite, 94  
 EOscFileOp\_Open, 94  
 EOscFileOp\_Read, 94  
 EOscFileOp\_Seek, 94  
 EOscFileOp\_Size, 94  
 EOscFileOp\_Tell, 94  
 EOscFileOp\_Write, 94  
 EPVDNSCancel, 95  
 EPVDNSFailure, 95  
 EPVDNSGetHostName, 95  
 EPVDNSPending, 95  
 EPVDNSSuccess, 95  
 EPVDNSTimeout, 95  
 oscl\_chdir, 95  
 OSCL\_FILE\_CHAR\_PATH\_-  
 DELIMITER, 93  
 OSCL\_FILE\_STATS\_LOGGER\_NODE,  
 93  
 OSCL\_FILE\_WCHAR\_PATH\_-  
 DELIMITER, 93

OSCL\_FILEMGMT\_E\_ALREADY\_EXISTS, 94  
 OSCL\_FILEMGMT\_E\_NO\_MATCH, 94  
 OSCL\_FILEMGMT\_E\_NOT\_EMPTY, 94  
 OSCL\_FILEMGMT\_E\_NOT\_IMPLEMENTED, 94  
 OSCL\_FILEMGMT\_E\_OK, 94  
 OSCL\_FILEMGMT\_E\_PATH\_NOT\_FOUND, 94  
 OSCL\_FILEMGMT\_E\_PATH\_TOO\_LONG, 94  
 OSCL\_FILEMGMT\_E\_PERMISSION\_DENIED, 94  
 OSCL\_FILEMGMT\_E\_SYS\_SPECIFIC, 94  
 OSCL\_FILEMGMT\_E\_UNKNOWN, 94  
 OSCL\_FILEMGMT\_ERR\_TYPE, 94  
 OSCL\_FILEMGMT\_MODE\_DIR, 94  
 OSCL\_FILEMGMT\_MODES, 94  
 OSCL\_FILEMGMT\_PERMS, 94  
 OSCL\_FILEMGMT\_PERMS\_EXECUTE, 94  
 OSCL\_FILEMGMT\_PERMS\_READ, 94  
 OSCL\_FILEMGMT\_PERMS\_WRITE, 94  
 OSCL\_FSSTAT, 93  
 oscl\_getcwd, 95, 96  
 OSCL\_IO\_EXTENSION\_MAXLEN, 93  
 OSCL\_IO\_FILENAME\_MAXLEN, 93  
 oscl\_mkdir, 96  
 oscl\_rename, 96, 97  
 oscl\_rmdir, 97  
 oscl\_stat, 97, 98  
 OSCL\_STAT\_BUF, 93  
 oscl\_statfs, 98  
 TOsclFileHandle, 93  
 TOsclFileOffsetInt32, 93  
 TOsclFileOp, 94  
 TPVDNSEvent, 95  
 TPVDNSFxN, 95  
 OsclIPSocketI, 403  
   OsclIPSocketI, 404  
 OsclIPSocketI  
   ~OsclIPSocketI, 404  
   Alloc, 404  
   Bind, 404  
   Close, 404  
   ConstructL, 404  
   GetRecvData, 404  
   GetSendData, 404  
   iAddress, 405  
   iAlloc, 405  
   iId, 405  
   iLogger, 405  
   iObserver, 405  
 iSocket, 405  
 iSocketServ, 405  
 Join, 404  
 OsclIPSocketI, 404  
 OsclSocketMethod, 405  
 OsclSocketRequestAO, 405  
 SetRecvBufferSize, 405  
 SocketServ, 405  
 OsclJoin  
   osclconfig\_io.h, 795  
 OsclJump, 406  
   OsclErrorTrapImp, 369  
 OsclJump  
   ~OsclJump, 406  
   Jump, 406  
   OsclErrorTrapImp, 406  
   StaticJump, 406  
   Top, 406  
 OsclJumpMark  
   OsclErrorTrapImp, 369  
 OsclLeaveCode  
   osclerror, 89  
 OsclListen  
   osclconfig\_io.h, 795  
 OsclListenMethod, 407  
 OsclListenMethod  
   ~OsclListenMethod, 407  
   Listen, 407  
   ListenRequest, 407  
   NewL, 407  
 OsclListenRequest, 408  
   OsclListenRequest, 408  
 OsclListenRequest  
   Listen, 408  
   OsclListenRequest, 408  
 OsclLockBase, 409  
 OsclLockBase  
   ~OsclLockBase, 409  
   Lock, 409  
   Unlock, 409  
 OsclMakeSockAddr  
   osclconfig\_io.h, 795  
 OsclMem, 410  
   OsclMemGlobalAuditObject, 426  
 OsclMem  
   Cleanup, 410  
   Init, 410  
 OsclMemAllocator, 411  
 OsclMemAllocator  
   allocate, 411  
   allocate\_fl, 411  
   deallocate, 411  
 OsclMemAllocDestructDealloc, 412  
 OsclMemAllocDestructDealloc

---

```

allocate, 412
allocate_fl, 412
deallocate, 412
destruct_and_dealloc, 412
OsclMemAudit, 414
  OsclMemAudit, 414
OsclMemAudit
  ~OsclMemAudit, 414
  GetLock, 415
  MM_AddTag, 415
  MM_allocate, 415
  MM_CreateAllocNodeInfo, 415
  MM_deallocate, 415
  MM_GetAllocNo, 415
  MM_GetAllocNodeInfo, 415
  MM_GetExistingTag, 416
  MM_GetMode, 416
  MM_GetNumAllocNodes, 416
  MM_GetOverheadStats, 416
  MM_GetPostfillPattern, 416
  MM_GetPrefillPattern, 416
  MM_GetRefCount, 416
  MM_GetRootNode, 417
  MM_GetStats, 417
  MM_GetStatsInDepth, 417
  MM_GetTagName, 417
  MM_GetTreeNodes, 417
  MM_ReleaseAllocNodeInfo, 417
  MM_SetFailurePoint, 417
  MM_SetMode, 418
  MM_SetPostfillPattern, 418
  MM_SetPrefillPattern, 418
  MM_SetTagLevel, 418
  MM_UnsetFailurePoint, 418
  MM_Validate, 418
  OsclMemAudit, 414
  OsclMemGlobalAuditObject, 419
OSCLMemAutoPtr, 420
  OSCLMemAutoPtr, 421
OSCLMemAutoPtr
  ~OSCLMemAutoPtr, 421
  _Ownership, 423
  allocate, 422
  deallocate, 422
  get, 422
  operator *, 422
  operator->, 422
  operator=, 422
  OSCLMemAutoPtr, 421
  release, 422
  setWithoutOwnership, 422
  takeOwnership, 423
OsclMemBasicAllocator, 424
OsclMemBasicAllocator
  allocate, 424
  deallocate, 424
  OsclMemBasicAllocDestructDealloc, 425
  OsclMemBasicAllocDestructDealloc
    allocate, 425
    deallocate, 425
    destruct_and_dealloc, 425
  OsclMemGlobalAuditObject, 426
    OsclMemAudit, 419
  OsclMemGlobalAuditObject
    audit_type, 426
    getGlobalMemAuditObject, 426
    OsclMem, 426
  OsclMemInit
    osclmemory, 59
  osclmemory
    _OSCL_CLEANUP_BASE_CLASS, 47
    _OSCL_TRAP_NEW, 47
    _oscl_audit_calloc, 56
    _oscl_audit_free, 56
    _oscl_audit_malloc, 56
    _oscl_audit_new, 56
    _oscl_audit_realloc, 57
    _oscl_calloc, 57
    _oscl_default_audit_calloc, 57
    _oscl_default_audit_malloc, 57
    _oscl_default_audit_new, 57
    _oscl_default_audit_realloc, 57
    _oscl_free, 57
    _oscl_malloc, 57
    _oscl_realloc, 57
    ALLOC_NODE_FLAG, 59
    COMPUTE_MEM_ALIGN_SIZE, 48
    DEFAULT_MM_AUDIT_MODE, 49
    DEFAULT_POSTFILL_PATTERN, 49
    DEFAULT_PREFILL_PATTERN, 49
    FENCE_PATTERN, 49
    MEM_ALIGN_SIZE, 49
    MIN_FENCE_SIZE, 49
    MM_ALLOC_MAX_QUERY_-
      FILENAME_LEN, 49
    MM_ALLOC_MAX_QUERY_TAG_LEN,
      49
    MM_AllocNodeAutoPtr, 56
    MM_AUDIT_ALLOC_NODE_-
      ENABLE_FLAG, 49
    MM_AUDIT_ALLOC_NODE_-
      SUPPORT, 49
    MM_AUDIT_FAILURE_SIMULATION_-
      SUPPORT, 49
    MM_AUDIT_FENCE_SUPPORT, 49
    MM_AUDIT_FILL_SUPPORT, 49
    MM_AUDIT_INCLUDE_ALL_HEAP_-
      VALIDATION, 49

```

MM\_AUDIT\_POSTFILL\_FLAG, 49  
 MM\_AUDIT\_PREFILL\_FLAG, 49  
 MM\_AUDIT\_SUPPRESS\_FILENAME\_-  
   FLAG, 49  
 MM\_AUDIT\_VALIDATE\_ALL\_HEAP\_-  
   FLAG, 49  
 MM\_AUDIT\_VALIDATE\_BLOCK, 49  
 MM\_AUDIT\_VALIDATE\_ON\_FREE\_-  
   FLAG, 49  
 MM\_StatsNodeTagTreeType, 56  
 MMAuditCharAutoPtr, 56  
 MMAuditUint8AutoPtr, 56  
 operator delete, 57  
 operator delete[], 57  
 operator new, 57  
 operator new[], 57  
 OSCL\_ALLOC\_DELETE, 49  
 OSCL\_ALLOC\_NEW, 50  
 OSCL\_ARRAY\_DELETE, 50  
 OSCL\_ARRAY\_NEW, 50  
 OSCL\_AUDIT\_ARRAY\_NEW, 50  
 OSCL\_AUDIT\_CALLOC, 51  
 OSCL\_AUDIT\_MALLOC, 51  
 OSCL\_AUDIT\_NEW, 51  
 OSCL\_AUDIT\_REALLOC, 52  
 OSCL\_CALLOC, 52  
 oscl\_calloc, 52  
 OSCL\_CLEANUP\_BASE\_CLASS, 52  
 OSCL\_DEFAULT\_FREE, 53  
 OSCL\_DEFAULT\_MALLOC, 53  
 OSCL\_DELETE, 53  
 OSCL\_DISABLE\_WARNING\_-  
   RETURN\_TYPE\_NOT\_UDT, 53  
 OSCL\_DISABLE\_WARNING\_-  
   TRUNCATE\_DEBUG\_MESSAGE,  
   53  
 OSCL\_FREE, 53  
 oscl\_free, 53  
 OSCL\_HAS\_GLOBAL\_NEW\_DELETE,  
   53  
 OSCL\_MALLOC, 54  
 oscl\_malloc, 54  
 oscl\_mem\_aligned\_size, 57  
 oscl\_memcmp, 58  
 oscl\_memcpy, 58  
 oscl\_memmove, 58  
 oscl\_memmove32, 58  
 oscl\_memset, 59  
 OSCL\_NEW, 54  
 OSCL\_PLACEMENT\_NEW, 54  
 OSCL\_REALLOC, 54  
 oscl\_realloc, 54  
 OSCL\_TRAP\_ALLOC\_NEW, 54  
 OSCL\_TRAP\_AUDIT\_NEW, 55  
 OSCL\_TRAP\_NEW, 55  
 OsclMemInit, 59  
 OsclMemStatsNodeAutoPtr, 56  
 OsclTagTreeType, 56  
 TagTree\_Allocator, 56  
 OsclMemoryFragment, 427  
 OsclMemoryFragment  
   len, 427  
   ptr, 427  
 OsclMemPoolAllocator, 428  
   OsclMemPoolAllocator, 428  
 OsclMemPoolAllocator  
   ~OsclMemPoolAllocator, 428  
   CreateMemPool, 428  
   DestroyMemPool, 428  
   oscl\_mem\_aligned\_size, 428  
   OsclMemPoolAllocator, 428  
 OsclMemPoolFixedChunkAllocator, 429  
   OsclMemPoolFixedChunkAllocator, 430  
 OsclMemPoolFixedChunkAllocator  
   ~OsclMemPoolFixedChunkAllocator, 430  
   addRef, 430  
   allocate, 430  
   CancelFreeChunkAvailableCallback, 430  
   createmempool, 430  
   deallocate, 431  
   destroymempool, 431  
   enablenullpointerreturn, 431  
   iCheckNextAvailableFreeChunk, 432  
   iChunkSize, 432  
   iChunkSizeMemAligned, 432  
   iEnableNullPtrReturn, 432  
   iFreeMemChunkList, 432  
   iMemPool, 432  
   iMemPoolAllocator, 432  
   iNextAvailableContextData, 432  
   iNumChunk, 432  
   iObserver, 432  
   iRefCount, 432  
   notifyfreechunkavailable, 431  
   OsclMemPoolFixedChunkAllocator, 430  
   removeRef, 431  
 OsclMemPoolFixedChunkAllocatorObserver,  
   433  
 OsclMemPoolFixedChunkAllocatorObserver  
   ~OsclMemPoolFixedChunkAllocatorObserver,  
   433  
   freechunkavailable, 433  
 OsclMemPoolResizableAllocator, 434  
   OsclMemPoolResizableAllocator, 435  
 OsclMemPoolResizableAllocator  
   ~OsclMemPoolResizableAllocator, 435  
   addnewmempoolbuffer, 435  
   addRef, 435

allocate, 436  
 allocateblock, 436  
 CancelFreeChunkAvailableCallback, 436  
 CancelFreeMemoryAvailableCallback, 436  
 deallocate, 436  
 deallocateblock, 436  
 destroyallmempoolbuffers, 436  
 enablenullpointerreturn, 436  
 findfreeblock, 437  
 getAllocatedSize, 437  
 getAvailableSize, 437  
 getBufferSize, 437  
 getLargestContiguousFreeBlockSize, 437  
 getMemPoolBufferAllocatedSize, 437  
 getMemPoolBufferSize, 437  
 iBlockInfoAlignedSize, 439  
 iBufferInfoAlignedSize, 439  
 iCheckFreeMemoryAvailable, 439  
 iCheckNextAvailable, 439  
 iEnableNullPtrReturn, 439  
 iExpectedNumBlocksPerBuffer, 439  
 iFreeMemContextData, 439  
 iFreeMemPoolObserver, 439  
 iMaxNewMemPoolBufferSz, 439  
 iMemPoolBufferAllocator, 439  
 iMemPoolBufferList, 439  
 iMemPoolBufferNumLimit, 439  
 iMemPoolBufferSize, 439  
 iNextAvailableContextData, 439  
 iObserver, 439  
 iRefCount, 439  
 iRequestedAvailableFreeMemSize, 439  
 iRequestedNextAvailableSize, 439  
 memoryPoolBufferMgmtOverhead, 437  
 notifyfreeblockavailable, 437  
 notifyfreememoryavailable, 437  
 OsclMemPoolResizableAllocator, 435  
 removeRef, 438  
 setMaxSzForNewMemPoolBuffer, 438  
 trim, 438  
 validateblock, 438  
 OsclMemPoolResizableAllocator::MemPoolBlockInfo, 440  
 OsclMemPoolResizableAllocator::MemPoolBlockInfo  
   iBlockBuffer, 440  
   iBlockPostFence, 440  
   iBlockPreFence, 440  
   iBlockSize, 440  
   iNextFreeBlock, 440  
   iParentBuffer, 440  
   iPrevFreeBlock, 440  
 OsclMemPoolResizableAllocator::MemPoolBufferInfo, 441  
   OsclNativeFile, 448  
   Oscl\_FileServer, 189

OsclMemPoolResizableAllocator::MemPool-  
   BufferInfo  
   iAllocatedSz, 441  
   iBufferPostFence, 441  
   iBufferPreFence, 441  
   iBufferSize, 441  
   iEndAddr, 441  
   iNextFreeBlock, 441  
   iNumOutstanding, 441  
   iStartAddr, 441  
 OsclMemPoolResizableAllocatorMemoryObserver, 442  
 OsclMemPoolResizableAllocatorMemory-  
   Observer  
   ~OsclMemPoolResizableAllocatorMemoryObserver, 442  
   freememoryavailable, 442  
 OsclMemPoolResizableAllocatorObserver, 443  
 OsclMemPoolResizableAllocatorObserver  
   ~OsclMemPoolResizableAllocatorObserver, 443  
   freeblockavailable, 443  
 OsclMemStatsNode, 444  
   OsclMemStatsNode, 444  
 OsclMemStatsNode  
   ~OsclMemStatsNode, 444  
   operator delete, 444  
   operator new, 444  
   OsclMemStatsNode, 444  
   pMMFIParam, 444  
   pMMStats, 444  
   reset, 444  
   tag, 444  
 OsclMemStatsNodeAutoPtr  
   osclmemory, 56  
 OsclMutex, 445  
   OsclMutex, 445  
 OsclMutex  
   ~OsclMutex, 445  
   Close, 445  
   Create, 445  
   Lock, 446  
   OsclMutex, 445  
   TryLock, 446  
   Unlock, 446  
 OsclNameString, 447  
   OsclNameString, 447  
 OsclNameString  
   MaxLen, 447  
   OsclNameString, 447  
   Set, 447  
   Str, 447

OsclNativeFile, 449  
 OsclNativeFile  
   ~OsclNativeFile, 449  
   Close, 449  
   EndOfFile, 449  
   Flush, 449  
   GetError, 449  
   GetReadAsyncNumElements, 449  
   HasAsyncRead, 449  
   Mode, 449  
   Open, 449  
   OsclNativeFile, 449  
   Read, 449  
   ReadAsync, 449  
   ReadAsyncCancel, 449  
   Seek, 450  
   Size, 450  
   Tell, 450  
   Write, 450  
 OsclNativeFileParams, 451  
   OsclNativeFileParams, 451  
 OsclNativeFileParams  
   iAsyncReadBufferSize, 451  
   iNativeAccessMode, 451  
   iNativeBufferSize, 451  
   OsclNativeFileParams, 451  
 OsclNetworkAddress, 452  
   OsclNetworkAddress, 452  
 OsclNetworkAddress  
   ipAddr, 452  
   operator==, 452  
   OsclNetworkAddress, 452  
   port, 452  
 OsclNoYieldMutex  
   oscl\_mutex.h, 704  
 OsclNullLock, 453  
 OsclNullLock  
   ~OsclNullLock, 453  
   Lock, 453  
   Unlock, 453  
 OsclPending  
   osclerror, 88  
 OsclPriorityLink, 454  
 OsclPriorityLink  
   iPriority, 454  
 OsclPriorityList, 455  
   OsclPriorityList, 455  
 OsclPriorityList  
   Head, 455  
   Insert, 455  
   IsHead, 455  
   IsTail, 455  
   OsclPriorityList, 455  
   Tail, 455  
 OsclPriorityQueue, 456  
 OsclPriorityQueue, 457  
 OsclPriorityQueue  
   ~OsclPriorityQueue, 457  
   c, 459  
   comp, 459  
   compare\_EQ, 457  
   compare\_LT, 457  
   const\_reference, 457  
   container\_type, 457  
   empty, 458  
   find\_heap, 458  
   iterator, 457  
   oscl\_priqueue\_test, 459  
   OsclPriorityQueue, 457  
   pop, 458  
   pop\_heap, 458  
   push, 458  
   push\_heap, 458  
   remove, 458  
   reserve, 458  
   size, 458  
   swap, 458  
   top, 458  
   validate, 459  
   value\_type, 457  
   vec, 459  
 OsclPriorityQueueBase, 460  
   Oscl\_Vector\_Base, 284  
 OsclPriorityQueueBase  
   ~OsclPriorityQueueBase, 460  
   construct, 460  
   find\_heap, 460  
   pop\_heap, 460  
   push\_heap, 460  
   remove, 460  
 osclproc  
   EPVThreadContext\_InThread, 102  
   EPVThreadContext\_NonOsclThread, 102  
   EPVThreadContext\_OsclThread, 102  
   EPVThreadContext\_Undetermined, 102  
   OSCL\_PERF\_SUMMARY\_LOGGING, 101  
   OSCL\_REQUEST\_ERR\_CANCEL, 102  
   OSCL\_REQUEST\_ERR\_GENERAL, 102  
   OSCL\_REQUEST\_ERR\_NONE, 102  
   OSCL\_REQUEST\_PENDING, 102  
   OSCL\_ZEROIZE, 101  
   OsclPtrAdd, 102  
   OsclPtrSub, 102  
   PV\_SCHED\_CHECK\_Q, 101  
   PV\_SCHED\_ENABLE\_AO\_STATS, 101  
   PV\_SCHED\_ENABLE\_LOOP\_STATS, 101

PV\_SCHED\_ENABLE\_PERF\_-  
     LOGGING, 101  
 PV\_SCHED\_ENABLE\_THREAD\_-  
     CONTEXT\_CHECKS, 101  
 PV\_SCHED\_FAIR\_SCHEDULING, 101  
 PV\_SCHED\_LOG\_Q, 101  
 PVEEXECNAMELEN, 101  
 PVSCHEDNAMELEN, 101  
 QUE\_ITER\_BEGIN, 101  
 QUE\_ITER\_END, 101  
 TOsclReady, 102  
 TPVThreadContext, 102  
**OsclProcStatus**, 461  
     ALREADY\_SUSPENDED\_ERROR, 461  
     BAD\_THREADID\_ADDR\_ERROR, 461  
     EXCEED\_MAX\_COUNT\_VARIABLE\_-  
         ERROR, 462  
     EXCEED\_MAX\_SEM\_COUNT\_ERROR,  
         462  
     INVALID\_ACCESS\_ERROR, 462  
     INVALID\_ARGUMENT\_ERROR, 462  
     INVALID\_FUNCTION\_ERROR, 462  
     INVALID\_HANDLE\_ERROR, 462  
     INVALID\_OPERATION\_ERROR, 462  
     INVALID\_PARAM\_ERROR, 461  
     INVALID\_POINTER\_ERROR, 462  
     INVALID\_PRIORITY\_ERROR, 461  
     INVALID\_THREAD\_ERROR, 461  
     INVALID\_THREAD\_ID\_ERROR, 461  
     MAX\_THRDS\_REACHED\_ERROR, 461  
     MUTEX\_LOCKED\_ERROR, 462  
     NO\_PERMISSION\_ERROR, 461  
     NOT\_ENOUGH\_MEMORY\_ERROR, 461  
     NOT\_ENOUGH\_RESOURCES\_ERROR,  
         461  
     NOT\_IMPLEMENTED, 462  
     NOT\_SUSPENDED\_ERROR, 461  
     OTHER\_ERROR, 461  
     OUTOFMEMORY\_ERROR, 461  
     PSHARED\_ATTRIBUTE\_SETTING\_-  
         ERROR, 462  
     PSHARED\_NOT\_ZERO\_ERROR, 462  
     RELOCK\_MUTEX\_ERROR, 462  
     SEM\_NOT\_SIGNALLED\_ERROR, 462  
     SUCCESS\_ERROR, 461  
     SYSTEM\_RESOURCES\_-  
         UNAVAILABLE\_ERROR, 462  
     THREAD\_1\_INACTIVE\_ERROR, 461  
     THREAD\_BLOCK\_ERROR, 462  
     THREAD\_NOT\_OWN\_MUTEX\_-  
         ERROR, 462  
     TOO\_MANY\_THREADS\_ERROR, 461  
     WAIT\_ABANDONED\_ERROR, 462  
     WAIT\_TIMEOUT\_ERROR, 462  
  
**OsclProcStatus**  
     eOsclProcError, 461  
**OsclPtr**, 463  
     OsclPtr, 463  
**OsclPtr**  
     Append, 463  
     Length, 463  
     OsclPtr, 463  
     Ptr, 463  
     Set, 463  
     SetLength, 463  
     Zero, 463  
**OsclPtrAdd**  
     osclproc, 102  
**OsclPtrC**, 465  
     OsclPtrC, 466  
**OsclPtrC**  
     Left, 466  
     Length, 466  
     OsclPtrC, 466  
     Ptr, 466  
     Right, 466  
     Set, 466  
     SetLength, 466  
     Zero, 466  
**OsclPtrSub**  
     osclproc, 102  
**OsclRand**, 467  
**OsclRand**  
     Rand, 467  
     Seed, 467  
**OsclReadyAlloc**, 468  
**OsclReadyAlloc**  
     allocate, 468  
     allocate\_fl, 468  
     deallocate, 468  
**OsclReadyCompare**, 469  
     PVActiveBase, 596  
**OsclReadyCompare**  
     compare, 469  
**OsclReadyQ**, 470  
     OsclExecSchedulerCommonBase, 388  
     PVActiveBase, 596  
     PVActiveStats, 597  
**OsclReadyQ**  
     Callback, 471  
     Construct, 471  
     Depth, 471  
     IsIn, 471  
     PendComplete, 471  
     PopTop, 471  
     RegisterForCallback, 471  
     Remove, 471  
     ThreadLogoff, 471

ThreadLogon, 471  
 TimerCallback, 471  
 Top, 471  
 WaitAndPopTop, 471  
 WaitForRequestComplete, 471  
**OsclReadySetPosition**  
 PVActiveBase, 596  
**OsclRecv**  
 osclconfig\_io.h, 795  
**OsclRecvFrom**  
 osclconfig\_io.h, 795  
**OsclRecvFromMethod**, 472  
**OsclRecvFromMethod**  
 ~OsclRecvFromMethod, 472  
 GetRecvData, 472  
 NewL, 472  
 RecvFrom, 472  
 RecvFromRequest, 472  
**OsclRecvFromRequest**, 474  
 OsclRecvFromRequest, 474  
 OsclSocketI, 524  
**OsclRecvFromRequest**  
 GetRecvData, 474  
 OsclRecvFromRequest, 474  
 RecvFrom, 474  
 Success, 474  
**OsclRecvMethod**, 476  
**OsclRecvMethod**  
 ~OsclRecvMethod, 476  
 GetRecvData, 476  
 NewL, 476  
 Recv, 476  
 RecvRequest, 476  
**OsclRecvRequest**, 477  
 OsclRecvRequest, 477  
 OsclSocketI, 524  
**OsclRecvRequest**  
 GetRecvData, 477  
 OsclRecvRequest, 477  
 Recv, 477  
 Success, 477  
**OsclRefCounter**, 478  
**OsclRefCounter**  
 ~OsclRefCounter, 478  
 addRef, 478  
 getCount, 478  
 removeRef, 478  
**OsclRefCounterDA**, 480  
 OsclRefCounterDA, 480  
**OsclRefCounterDA**  
 ~OsclRefCounterDA, 480  
 addRef, 481  
 getCount, 481  
 OsclRefCounterDA, 480  
 removeRef, 481  
 getCount, 482  
 OsclRefCounterDA, 482  
 addRef, 482  
 getCount, 482  
 removeRef, 482  
 OsclRefCounterMemFrag, 482  
 ~OsclRefCounterMemFrag, 482  
 OsclRefCounterMemFrag  
 ~OsclRefCounterMemFrag, 482  
 getCapacity, 483  
 getCount, 483  
 getMemFrag, 483  
 getMemFragPtr, 483  
 getMemFragSize, 483  
 getRefCounter, 483  
 operator=, 483  
 OsclRefCounterMemFrag, 482  
**OsclRefCounterMTDA**, 484  
 OsclRefCounterMTDA, 484  
**OsclRefCounterMTDA**  
 ~OsclRefCounterMTDA, 484  
 addRef, 485  
 getCount, 485  
 OsclRefCounterMTDA, 484  
 removeRef, 485  
**OsclRefCounterMTSA**, 486  
 OsclRefCounterMTSA, 486  
**OsclRefCounterMTSA**  
 ~OsclRefCounterMTSA, 486  
 addRef, 487  
 getCount, 487  
 OsclRefCounterMTSA, 486  
 removeRef, 487  
**OsclRefCounterSA**, 488  
 OsclRefCounterSA, 488  
**OsclRefCounterSA**  
 ~OsclRefCounterSA, 488  
 addRef, 489  
 getCount, 489  
 OsclRefCounterSA, 488  
 removeRef, 489  
**OsclRegistryAccessClient**, 490  
 OsclRegistryAccessClient, 490  
 OsclRegistryClientImpl, 498  
 OsclRegistryServTlsImpl, 501  
**OsclRegistryAccessClient**  
 ~OsclRegistryAccessClient, 490  
 Close, 490  
 Connect, 490  
 GetFactories, 490  
 GetFactory, 490  
 OsclRegistryAccessClient, 490  
**OsclRegistryAccessClientImpl**, 492  
**OsclRegistryAccessClientTlsImpl**, 493  
**OsclRegistryAccessElement**, 494  
**OsclRegistryAccessElement**  
 iFactory, 494  
 iMimeType, 494

**OsclRegistryClient**, 495  
 OsclRegistryClient, 495  
 OsclRegistryClientImpl, 498  
 OsclRegistryServTlsImpl, 501  
**OsclRegistryClient**  
 ~OsclRegistryClient, 495  
 Close, 495  
 Connect, 495  
 OsclRegistryClient, 495  
 Register, 495  
 UnRegister, 496  
**OsclRegistryClientImpl**, 497  
**OsclRegistryClientImpl**  
 Close, 498  
 Connect, 498  
 GetFactories, 498  
 GetFactory, 498  
 OsclRegistryAccessClient, 498  
 OsclRegistryClient, 498  
 Register, 498  
 UnRegister, 498  
**OsclRegistryClientTlsImpl**, 499  
**OsclRegistryServTlsImpl**, 500  
 OsclRegistryServTlsImpl, 501  
**OsclRegistryServTlsImpl**  
 ~OsclRegistryServTlsImpl, 501  
 Close, 501  
 Connect, 501  
 GetFactories, 501  
 GetFactory, 501  
 OsclRegistryAccessClient, 501  
 OsclRegistryClient, 501  
 OsclRegistryServTlsImpl, 501  
 Register, 501  
 UnRegister, 501  
**OsclReturnCode**  
 oscrror, 89  
**OsclScheduler**, 502  
 OsclErrorTrapImp, 369  
 OsclExecScheduler, 381  
 OsclExecSchedulerCommonBase, 388  
**OsclScheduler**  
 Cleanup, 502  
 Init, 502  
**OsclSchedulerCommonBase**  
 PVActiveBase, 596  
**OsclSchedulerObserver**, 503  
**OsclSchedulerObserver**  
 ~OsclSchedulerObserver, 503  
 OsclSchedulerReadyCallback, 503  
 OsclSchedulerTimerCallback, 503  
**OsclSchedulerReadyCallback**  
 OsclSchedulerObserver, 503  
**OsclSchedulerTimerCallback**

OsclSchedulerObserver, 503  
 OsclScopedLock, 504  
 OsclScopedLock, 504  
**OsclScopedLock**  
 ~OsclScopedLock, 504  
 OsclScopedLock, 504  
**OsclSelect**, 505  
 OsclSelect, 506  
**OsclSelect**  
 iErrAlloc, 506  
 iHeapCheck, 506  
 iOsclBase, 506  
 iOsclErrorTrap, 506  
 iOsclLogger, 506  
 iOsclMemory, 506  
 iOsclScheduler, 506  
 iOutputFile, 506  
 iSchedulerAlloc, 506  
 iSchedulerName, 506  
 iSchedulerReserve, 506  
 OsclSelect, 506  
**OsclSemaphore**, 507  
 OsclSemaphore, 507  
**OsclSemaphore**  
 ~OsclSemaphore, 507  
 Close, 507  
 Create, 507  
 OsclSemaphore, 507  
 Signal, 508  
 TryWait, 508  
 Wait, 508  
**OsclSend**  
 osclconfig\_io.h, 796  
**OsclSendMethod**, 509  
**OsclSendMethod**  
 ~OsclSendMethod, 509  
 GetSendData, 509  
 NewL, 509  
 Send, 509  
 SendRequest, 509  
**OsclSendRequest**, 510  
 OsclSendRequest, 510  
 OsclSocketI, 524  
**OsclSendRequest**  
 GetSendData, 510  
 OsclSendRequest, 510  
 Send, 510  
 Success, 510  
**OsclSendTo**  
 osclconfig\_io.h, 796  
**OsclSendToMethod**, 511  
**OsclSendToMethod**  
 ~OsclSendToMethod, 511  
 GetSendData, 511

NewL, [511](#)  
 SendTo, [511](#)  
 SendToRequest, [511](#)  
**OsclSendToRequest**, [512](#)  
 OsclSendToRequest, [512](#)  
 OsclSocketI, [524](#)  
**OsclSendToRequest**  
 GetSendData, [512](#)  
 OsclSendToRequest, [512](#)  
 SendTo, [512](#)  
 Success, [512](#)  
**OsclSetNonBlocking**  
 osclconfig\_io.h, [796](#)  
**OsclSetRecvBufferSize**  
 osclconfig\_io.h, [796](#)  
**OsclSharedPtr**, [513](#)  
 OsclSharedPtr, [514](#)  
**OsclSharedPtr**  
 ~OsclSharedPtr, [514](#)  
 get\_count, [514](#)  
 GetRefCounter, [514](#)  
 GetRep, [514](#)  
 operator \*, [514](#)  
 operator TheClass \*, [515](#)  
 operator->, [515](#)  
 operator=, [515](#)  
 OsclSharedPtr, [514](#)  
 Unbind, [515](#)  
**OsclShutdown**  
 osclconfig\_io.h, [796](#)  
**OsclShutdownMethod**, [516](#)  
**OsclShutdownMethod**  
 ~OsclShutdownMethod, [516](#)  
 NewL, [516](#)  
 Shutdown, [516](#)  
 ShutdownRequest, [516](#)  
**OsclShutdownRequest**, [517](#)  
 OsclShutdownRequest, [517](#)  
 OsclSocketI, [524](#)  
**OsclShutdownRequest**  
 OsclShutdownRequest, [517](#)  
 Shutdown, [517](#)  
**OsclSingleton**, [518](#)  
 OsclSingleton, [518](#)  
**OsclSingleton**  
 ~OsclSingleton, [518](#)  
 \_Ptr, [519](#)  
 operator \*, [518](#)  
 operator->, [518](#)  
 OsclSingleton, [518](#)  
 set, [518](#)  
**OsclSingletonRegistry**, [520](#)  
**OsclSingletonRegistry**  
 getInstance, [520](#)  
 lockAndGetInstance, [520](#)  
**OsclBase**, [520](#)  
 registerInstance, [520](#)  
 registerInstanceAndUnlock, [520](#)  
**OsclSocket**  
 osclconfig\_io.h, [796](#)  
**OsclSocketCleanup**  
 osclconfig\_io.h, [797](#)  
**OsclSocketI**, [521](#)  
 OsclSocketRequestAO, [539](#)  
 OsclSocketServI, [543](#)  
**OsclSocketI**  
 ~OsclSocketI, [522](#)  
 Accept, [522](#)  
 Bind, [522](#)  
 Close, [522](#)  
 Connect, [522](#)  
 Join, [522](#)  
 Listen, [522](#)  
 Logger, [522](#)  
 MakeAddr, [523](#)  
 NewL, [523](#)  
 Open, [523](#)  
 OsclAcceptRequest, [524](#)  
 OsclConnectRequest, [524](#)  
 OsclRecvFromRequest, [524](#)  
 OsclRecvRequest, [524](#)  
 OsclSendRequest, [524](#)  
 OsclSendToRequest, [524](#)  
 OsclShutdownRequest, [524](#)  
 OsclTCPSocket, [524](#)  
 OsclUDPSocket, [524](#)  
 ProcessAccept, [523](#)  
 ProcessConnect, [523](#)  
 ProcessRecv, [523](#)  
 ProcessRecvFrom, [523](#)  
 ProcessSend, [523](#)  
 ProcessSendTo, [523](#)  
 ProcessShutdown, [523](#)  
 Recv, [523](#)  
 RecvFrom, [523](#)  
 RecvFromSuccess, [523](#)  
 RecvSuccess, [523](#)  
 Send, [523](#)  
 SendSuccess, [524](#)  
 SendTo, [524](#)  
 SendToSuccess, [524](#)  
 SetRecvBufferSize, [524](#)  
 Shutdown, [524](#)  
 Socket, [524](#)  
**OsclSocketIBase**, [526](#)  
 OsclSocketIBase, [527](#)  
**OsclSocketIBase**  
 ~OsclSocketIBase, [527](#)

Accept, [527](#)  
 Bind, [527](#)  
 BindAsync, [527](#)  
 CancelAccept, [528](#)  
 CancelBind, [528](#)  
 CancelConnect, [528](#)  
 CancelFxn, [528](#)  
 CancelListen, [528](#)  
 CancelRecv, [528](#)  
 CancelRecvFrom, [528](#)  
 CancelSend, [528](#)  
 CancelSendTo, [528](#)  
 CancelShutdown, [528](#)  
 Close, [528](#)  
 Connect, [528](#)  
 GetShutdown, [528](#)  
 HasAsyncBind, [528](#)  
 HasAsyncListen, [528](#)  
 iAlloc, [530](#)  
 iSocketServ, [530](#)  
 IsOpen, [528](#)  
 Join, [528](#)  
 Listen, [528](#)  
 ListenAsync, [528](#)  
 Open, [529](#)  
 OsclSocketIBase, [527](#)  
 OsclSocketMethod, [530](#)  
 OsclSocketRequest, [530](#)  
 OsclSocketRequestAO, [530](#)  
 OsclTCPSocket, [530](#)  
 OsclUDPSocket, [530](#)  
 Recv, [529](#)  
 RecvFrom, [529](#)  
 RecvFromSuccess, [529](#)  
 RecvSuccess, [529](#)  
 Send, [529](#)  
 SendSuccess, [529](#)  
 SendTo, [529](#)  
 SendToSuccess, [529](#)  
 Shutdown, [530](#)  
 OsclSocketMethod, [531](#)  
 OsclIPSocketI, [405](#)  
 OsclSocketIBase, [530](#)  
 OsclSocketMethod, [532](#)  
 OsclSocketRequestAO, [539](#)  
 OsclSocketMethod  
   ~OsclSocketMethod, [532](#)  
 Abort, [532](#)  
 AbortAll, [532](#)  
 Alloc, [532](#)  
 CancelMethod, [532](#)  
 ConstructL, [532](#)  
 iContainer, [533](#)  
 iSocketFxn, [533](#)  
 iSocketRequestAO, [533](#)  
 MethodDone, [532](#)  
 OsclSocketMethod, [532](#)  
 Run, [532](#)  
 StartMethod, [533](#)  
 OsclSocketObserver, [534](#)  
 OsclSocketObserver  
   ~OsclSocketObserver, [534](#)  
 HandleSocketEvent, [534](#)  
 OsclSocketRequest, [535](#)  
 OsclSocketIBase, [530](#)  
 OsclSocketRequest, [535](#)  
 OsclSocketRequestAO, [539](#)  
 OsclSocketServI, [543](#)  
 OsclSocketRequest  
   Activate, [535](#)  
   CancelRequest, [535](#)  
   Complete, [535](#)  
   Fxn, [535](#)  
   iParam, [535](#)  
   iSocketI, [535](#)  
   iSocketRequestAO, [535](#)  
   OsclSocketRequest, [535](#)  
 OsclSocketRequestAO, [536](#)  
 OsclIPSocketI, [405](#)  
 OsclSocketIBase, [530](#)  
 OsclSocketRequestAO, [537](#)  
 OsclSocketRequestAO  
   ~OsclSocketRequestAO, [537](#)  
   Abort, [537](#)  
   Alloc, [537](#)  
   CleanupParam, [537](#)  
   ConstructL, [537](#)  
   DoCancel, [537](#)  
   GetSocketError, [537](#)  
   iContainer, [539](#)  
   Id, [538](#)  
   iParam, [539](#)  
   iParamSize, [539](#)  
   iSocketError, [539](#)  
   NewRequest, [538](#)  
   OsclSocketI, [539](#)  
   OsclSocketMethod, [539](#)  
   OsclSocketRequest, [539](#)  
   OsclSocketRequestAO, [537](#)  
   RequestDone, [538](#)  
   Run, [538](#)  
   SocketI, [538](#)  
   SocketObserver, [538](#)  
   Success, [538](#)  
 OsclSocketSelect  
   osclconfig\_io.h, [797](#)  
 OsclSocketServ, [540](#)  
 OsclSocketServI, [543](#)

**OsclSocketServ**  
 ~OsclSocketServ, [540](#)  
 Close, [540](#)  
 Connect, [540](#)  
 NewL, [541](#)  
 OsclDNS, [541](#)  
 OsclTCPSocket, [541](#)  
 OsclUDPSocket, [541](#)  
**OsclSocketServI**, [542](#)  
 OsclSocketServRequestList, [546](#)  
**OsclSocketServI**  
 Close, [542](#)  
 Connect, [542](#)  
 IsServerThread, [543](#)  
 LoopbackSocket, [543](#)  
 NewL, [543](#)  
 OsclDNSI, [543](#)  
 OsclSocketI, [543](#)  
 OsclSocketRequest, [543](#)  
 OsclSocketServ, [543](#)  
 OsclSocketServRequestList, [543](#)  
 OsclTCPSocketI, [543](#)  
 OsclUDPSocketI, [543](#)  
**OsclSocketServIBase**, [544](#)  
 ESocketServ\_Connected, [544](#)  
 ESocketServ\_Error, [545](#)  
 ESocketServ\_Idle, [544](#)  
 OsclSocketServIBase, [545](#)  
**OsclSocketServIBase**  
 ~OsclSocketServIBase, [545](#)  
 Close, [545](#)  
 Connect, [545](#)  
 iAlloc, [545](#)  
 iLogger, [545](#)  
 iServError, [545](#)  
 iServState, [545](#)  
 IsServConnected, [545](#)  
 OsclSocketServIBase, [545](#)  
 State, [545](#)  
 TSocketServState, [544](#)  
**OsclSocketServRequestList**, [546](#)  
 OsclSocketServI, [543](#)  
 OsclSocketServRequestList, [546](#)  
**OsclSocketServRequestList**  
 Add, [546](#)  
 Close, [546](#)  
 Open, [546](#)  
 OsclSocketServI, [546](#)  
 OsclSocketServRequestList, [546](#)  
 Remove, [546](#)  
 StartCancel, [546](#)  
 WaitOnRequests, [546](#)  
 Wakeup, [546](#)  
**OsclSocketServRequestQElem**, [548](#)  
**OsclSocketServRequestQElem**  
 iCancel, [548](#)  
 iSelect, [548](#)  
 iSocketRequest, [548](#)  
**OsclSocketServRequestQElem**, [548](#)  
**OsclSocketStartup**  
 osclconfig\_io.h, [797](#)  
**OsclSuccess**  
 osclerror, [88](#)  
**OsclTagTreeType**  
 osclmemory, [56](#)  
**OsclTCPSocket**, [549](#)  
 OsclSocketI, [524](#)  
 OsclSocketIBase, [530](#)  
 OsclSocketServ, [541](#)  
**OsclTCPSocket**  
 ~OsclTCPSocket, [550](#)  
 Accept, [550](#)  
 Bind, [550](#)  
 BindAsync, [550](#)  
 CancelAccept, [550](#)  
 CancelBind, [551](#)  
 CancelConnect, [551](#)  
 CancelListen, [551](#)  
 CancelRecv, [551](#)  
 CancelSend, [551](#)  
 CancelShutdown, [551](#)  
 Close, [551](#)  
 Connect, [552](#)  
 GetAcceptedSocketL, [552](#)  
 GetRecvData, [552](#)  
 GetSendData, [552](#)  
 Listen, [553](#)  
 ListenAsync, [553](#)  
 NewL, [553](#)  
 Recv, [553](#)  
 Send, [554](#)  
 Shutdown, [554](#)  
**OsclTCPSocketI**, [555](#)  
 OsclSocketServI, [543](#)  
**OsclTCPSocketI**  
 ~OsclTCPSocketI, [556](#)  
 Accept, [556](#)  
 BindAsync, [556](#)  
 CancelAccept, [556](#)  
 CancelBind, [556](#)  
 CancelConnect, [556](#)  
 CancelListen, [556](#)  
 CancelRecv, [556](#)  
 CancelSend, [556](#)  
 CancelShutdown, [556](#)  
 Close, [556](#)  
 Connect, [556](#)

GetAcceptedSocketL, 556  
 GetRecvData, 556  
 GetSendData, 556  
 Listen, 556  
 ListenAsync, 557  
 NewL, 557  
 Recv, 557  
 Send, 557  
 Shutdown, 557  
 OsclThread, 558  
     OsclThread, 558  
 OsclThread  
     ~OsclThread, 558  
     CompareId, 558  
     Create, 559  
     EnableKill, 559  
     Exit, 559  
     GetId, 559  
     GetPriority, 560  
     OsclThread, 558  
     Resume, 560  
     SetPriority, 560  
     SleepMillsec, 560  
     Suspend, 561  
     Terminate, 561  
 OsclThread\_State  
     oscl\_thread.h, 769  
 OsclThreadLock, 562  
     OsclThreadLock, 562  
 OsclThreadLock  
     ~OsclThreadLock, 562  
     Lock, 562  
     OsclThreadLock, 562  
     Unlock, 562  
 OsclThreadPriority  
     oscl\_thread.h, 769  
 OsclTickCount, 563  
 OsclTickCount  
     MsecToTicks, 563  
     TickCount, 563  
     TickCountFrequency, 563  
     TickCountPeriod, 563  
     TicksToMsec, 563  
 OSCLTICKCOUNT\_MAX\_TICKS  
     osclutil, 66  
 OsclTimer, 565  
     OsclTimer, 566  
 OsclTimer  
     ~OsclTimer, 566  
     callback\_timer\_type, 566  
     CallbackTimer< Alloc >, 567  
     Cancel, 566  
     Clear, 566  
     OsclTimer, 566  
     Request, 566  
     SetExactFrequency, 566  
     SetFrequency, 567  
     SetObserver, 567  
     TimerBaseElapsed, 567  
 OsclTimerCompare, 568  
     OsclExecSchedulerCommonBase, 388  
 OsclTimerCompare  
     compare, 568  
 OsclTimerObject, 569  
     OsclExecSchedulerCommonBase, 390  
     OsclTimerObject, 570  
     PVActiveBase, 596  
     PVActiveStats, 597  
     PVThreadContext, 616  
 OsclTimerObject  
     ~OsclTimerObject, 570  
     AddToScheduler, 570  
     After, 570  
     Cancel, 570  
     DoCancel, 570  
     IsBusy, 571  
     OsclTimerObject, 570  
     Priority, 571  
     RemoveFromScheduler, 571  
     RunError, 571  
     RunIfNotReady, 571  
     SetBusy, 571  
     SetStatus, 571  
     Status, 572  
     StatusRef, 572  
 OsclTimerObserver, 573  
 OsclTimerObserver  
     ~OsclTimerObserver, 573  
     TimeoutOccurred, 573  
 OsclTimerQ, 574  
 OsclTimerQ  
     Add, 574  
     Construct, 574  
     IsIn, 574  
     Pop, 574  
     PopTop, 574  
     Remove, 574  
     Top, 574  
 OsclTLS, 575  
     OsclTLS, 575  
 OsclTLS  
     ~OsclTLS, 575  
     \_Ptr, 576  
     operator \*, 575  
     operator->, 575  
     OsclTLS, 575  
     set, 575  
 OsclTLSEx, 577

OsclTLSEEx, [577](#)  
 OsclTLSEEx  
   ~OsclTLSEEx, [577](#)  
   \_Ptr, [578](#)  
   operator \*, [577](#)  
   operator->, [577](#)  
   OsclTLSEEx, [577](#)  
   set, [577](#)  
 OsclTLSRegistry, [579](#)  
 OsclTLSRegistry  
   getInstance, [579](#)  
   OsclBase, [579](#)  
   registerInstance, [579](#)  
 OsclTLSRegistryEx, [580](#)  
 OsclTLSRegistryEx  
   getInstance, [580](#)  
   registerInstance, [580](#)  
 OsclTrapItem, [581](#)  
   OsclTrapItem, [581](#)  
 OsclTrapItem  
   OsclTrapItem, [581](#)  
   OsclTrapStack, [581](#)  
   OsclTrapStackItem, [581](#)  
 OsclTrapOperation  
   osclerror, [89](#)  
 OsclTrapStack, [582](#)  
   OsclErrorTrapImp, [369](#)  
   OsclTrapItem, [581](#)  
 OsclTrapStack  
   OsclError, [582](#)  
   OsclErrorTrap, [582](#)  
   OsclErrorTrapImp, [582](#)  
 OsclTrapStackItem, [583](#)  
   OsclTrapItem, [581](#)  
   OsclTrapStackItem, [583](#)  
 OsclTrapStackItem  
   iCBase, [583](#)  
   iNext, [583](#)  
   iTAny, [583](#)  
   iTrapOperation, [583](#)  
   OsclTrapStackItem, [583](#)  
 OsclUDPSocket, [584](#)  
   OsclSocketI, [524](#)  
   OsclSocketIBase, [530](#)  
   OsclSocketServ, [541](#)  
 OsclUDPSocket  
   ~OsclUDPSocket, [584](#)  
   Bind, [585](#)  
   BindAsync, [585](#)  
   CancelBind, [585](#)  
   CancelRecvFrom, [585](#)  
   CancelSendTo, [585](#)  
   Close, [585](#)  
   GetRecvData, [586](#)  
     GetSendData, [586](#)  
     Join, [586](#)  
     NewL, [586](#)  
     RecvFrom, [587](#)  
     SendTo, [587](#)  
     SetRecvBufferSize, [587](#)  
 OsclUDPSocketI, [589](#)  
   OsclSocketServI, [543](#)  
 OsclUDPSocketI  
   ~OsclUDPSocketI, [590](#)  
   BindAsync, [590](#)  
   CancelBind, [590](#)  
   CancelRecvFrom, [590](#)  
   CancelSendTo, [590](#)  
   Close, [590](#)  
   GetRecvData, [590](#)  
   GetSendData, [590](#)  
   NewL, [590](#)  
   RecvFrom, [590](#)  
   SendTo, [590](#)  
 OsclUid32  
   oscl\_uuid.h, [780](#)  
 OsclUnMakeSockAddr  
   osclconfig\_io.h, [797](#)  
 osclutil  
   ~OSCL\_HeapString, [80](#)  
   ~OSCL\_StackString, [80](#)  
   ~OSCL\_wHeapString, [80](#)  
   ~OSCL\_wStackString, [80](#)  
   APPEND\_MEDIA\_AT\_END, [80](#)  
   BufferFreeFuncPtr, [66](#)  
   extract\_string, [66](#)  
   get\_cstr, [66, 67](#)  
   get\_maxsize, [67](#)  
   get\_size, [67, 68](#)  
   get\_str, [68](#)  
   GetBufferState, [68](#)  
   GetFragment, [69](#)  
   MediaTimestamp, [66](#)  
   operator=, [69, 70](#)  
   oscl\_abs, [70](#)  
   OSCL\_ASCII\_CASE\_MAGIC\_BIT, [80](#)  
   oscl\_asin, [70](#)  
   oscl\_atan, [70](#)  
   oscl\_cos, [70](#)  
   oscl\_exp, [71](#)  
   oscl\_floor, [71](#)  
   OSCL\_HeapString, [71](#)  
   oscl\_isdigit, [66](#)  
   oscl\_log, [72](#)  
   oscl\_log10, [72](#)  
   oscl\_pow, [72](#)  
   oscl\_sin, [72](#)  
   oscl\_snprintf, [72](#)



---

pop  
   Oscl\_Queue, 229  
   Oscl\_Queue\_Base, 231  
   OsclPriorityQueue, 458  
 pop\_back  
   Oscl\_Vector, 279  
   Oscl\_Vector\_Base, 283  
 pop\_heap  
   OsclPriorityQueue, 458  
   OsclPriorityQueueBase, 460  
 PopDealloc  
   OsclError, 363, 364  
 PopTop  
   OsclReadyQ, 471  
   OsclTimerQ, 574  
 port  
   OsclNetworkAddress, 452  
 PositionInBlock  
   OsclBinStream, 329  
 pPosition  
   OsclBinStream, 330  
 pPrev  
   MM\_AllocNode, 146  
 Preceeds  
   OsclFileCacheBuffer, 395  
 PrepRead  
   OsclFileCacheBuffer, 395  
 PrepWrite  
   OsclFileCacheBuffer, 395  
 Priority  
   OsclActiveObject, 303  
   OsclTimerObject, 571  
 ProcessAccept  
   OsclSocketI, 523  
 ProcessConnect  
   OsclSocketI, 523  
 ProcessRecv  
   OsclSocketI, 523  
 ProcessRecvFrom  
   OsclSocketI, 523  
 ProcessSend  
   OsclSocketI, 523  
 ProcessSendTo  
   OsclSocketI, 523  
 ProcessShutdown  
   OsclSocketI, 523  
 pRootNode  
   MM\_AllocBlockHdr, 143  
 pruneSubtree  
   MM\_Audit\_Imp, 154  
 PSHARED\_ATTRIBUTE\_SETTING\_ERROR  
   OsclProcStatus, 462  
 PSHARED\_NOT\_ZERO\_ERROR  
   OsclProcStatus, 462

pStats  
   MM\_Stats\_CB, 158  
 pStatsNode  
   MM\_AllocInfo, 145  
   OsclAuditCB, 312  
 Ptr  
   OsclPtr, 463  
   OsclPtrC, 466  
 ptr  
   OsclMemoryFragment, 427  
   StrPtrLen, 629  
   WStrPtrLen, 639  
 push  
   Oscl\_Queue, 229  
   Oscl\_Queue\_Base, 231  
   OsclPriorityQueue, 458  
 push\_back  
   Oscl\_Vector, 280  
   Oscl\_Vector\_Base, 283  
 push\_front  
   Oscl\_Vector, 280  
   Oscl\_Vector\_Base, 284  
 push\_heap  
   OsclPriorityQueue, 458  
   OsclPriorityQueueBase, 460  
 PushL  
   OsclError, 364  
 PV8601TIME\_BUFFER\_SIZE  
   osclbase, 43  
 PV8601timeStrBuf  
   osclbase, 32  
 PV8601ToRFC822  
   osclbase, 41  
 PV\_atof  
   osclutil, 79  
 PV\_atoi  
   osclutil, 79  
 PV\_CHAR\_CLOSE\_BRACKET  
   oscl\_uuid.h, 780  
 PV\_CHAR\_COMMA  
   oscl\_uuid.h, 780  
 PV\_DNS\_IS\_THREAD  
   oscl\_dns\_tuneables.h, 657  
 PV\_DNS\_SERVER  
   oscl\_dns\_tuneables.h, 657  
 PV\_DYNAMIC\_LOADING\_CONFIG\_FILE\_PATH  
   osclconfig\_lib.h, 800  
 PV\_OSCL\_SOCKET\_1MB\_RECV\_BUF  
   oscl\_socket\_tuneables.h, 753  
 PV\_OSCL\_SOCKET\_SERVER\_LOGGER\_OUTPUT  
   oscl\_socket\_tuneables.h, 753  
 PV\_OSCL\_SOCKET\_STATS\_LOGGING

oscl\_socket\_tuneables.h, 753  
**PV\_RUNTIME\_LIB\_FILENAME\_-  
EXTENSION**  
 osclconfig\_lib.h, 800  
**PV\_SCHED\_CHECK\_Q**  
 osclproc, 101  
**PV\_SCHED\_ENABLE\_AO\_STATS**  
 osclproc, 101  
**PV\_SCHED\_ENABLE\_LOOP\_STATS**  
 osclproc, 101  
**PV\_SCHED\_ENABLE\_PERF\_LOGGING**  
 osclproc, 101  
**PV\_SCHED\_ENABLE\_THREAD\_-  
CONTEXT\_CHECKS**  
 osclproc, 101  
**PV\_SCHED\_FAIR\_SCHEDULING**  
 osclproc, 101  
**PV\_SCHED\_LOG\_Q**  
 osclproc, 101  
**PV\_SOCKET\_REQUEST\_AO\_PRIORITY**  
 oscl\_socket\_tuneables.h, 753  
**PV\_SOCKET\_SERVER**  
 oscl\_socket\_tuneables.h, 753  
**PV\_SOCKET\_SERVER\_AO\_INTERVAL\_-  
MSEC**  
 oscl\_socket\_tuneables.h, 754  
**PV\_SOCKET\_SERVER\_AO\_PRIORITY**  
 oscl\_socket\_tuneables.h, 754  
**PV\_SOCKET\_SERVER\_IS\_THREAD**  
 oscl\_socket\_tuneables.h, 754  
**PV\_SOCKET\_SERVER\_SELECT**  
 oscl\_socket\_tuneables.h, 754  
**PV\_SOCKET\_SERVER\_SELECT\_-  
LOOPBACK\_SOCKET**  
 oscl\_socket\_tuneables.h, 754  
**PV\_SOCKET\_SERVER\_SELECT\_-  
TIMEOUT\_MSEC**  
 oscl\_socket\_tuneables.h, 754  
**PV\_SOCKET\_SERVER\_THREAD\_-  
PRIORITY**  
 oscl\_socket\_tuneables.h, 754  
**PV\_SOCKET\_SERVI\_STATS**  
 oscl\_socket\_tuneables.h, 754  
**PVActiveBase**, 593  
 OsclExecSchedulerBase, 382  
 OsclExecSchedulerCommonBase, 390  
**PVActiveBase**, 594  
**PVActiveStats**, 597  
**PVThreadContext**, 616  
**PVActiveBase**  
 ~PVActiveBase, 594  
**Activate**, 594  
**AddToScheduler**, 594  
**Cancel**, 594  
 Destroy, 594  
**DoCancel**, 594  
**iAddedNum**, 596  
**iBusy**, 596  
**iName**, 596  
**iPVActiveStats**, 596  
**iPVReadyQLink**, 596  
**IsAdded**, 594  
**IsInAnyQ**, 595  
**iStatus**, 596  
**iThreadContext**, 596  
**OsclActiveObject**, 596  
**OsclExecScheduler**, 596  
**OsclReadyCompare**, 596  
**OsclReadyQ**, 596  
**OsclReadySetPosition**, 596  
**OsclSchedulerCommonBase**, 596  
**OsclTimerObject**, 596  
**PVActiveBase**, 594  
**PVActiveStats**, 596  
**RemoveFromScheduler**, 595  
**Run**, 595  
**RunError**, 595  
**PVActiveStats**, 597  
 OsclExecSchedulerCommonBase, 390  
**PVActiveBase**, 596  
**PVActiveStats**  
 OsclActiveObject, 597  
 OsclExecScheduler, 597  
 OsclExecSchedulerCommonBase, 597  
 OsclReadyQ, 597  
 OsclTimerObject, 597  
**PVActiveBase**, 597  
**PVCleanupStack**  
 \_OsclHeapBase, 107  
**PVError\_DoLeave**  
 oscl\_error\_imp\_fatalerror.h, 665  
 oscl\_error\_imp\_jumps.h, 667  
 osclerror, 88  
**PVERROR\_IMP\_JUMPS**  
 osclerror, 88  
**PVERRORTRAP\_REGISTRY**  
 osclerror, 88  
**PVERRORTRAP\_REGISTRY\_ID**  
 osclerror, 89  
**PVEXECNAMELEN**  
 osclproc, 101  
**PVLogger**, 598  
 ~PVLogger, 599  
**AddAppender**, 599  
**AddFilter**, 599  
**alloc\_type**, 599  
**Cleanup**, 600  
**DisableAppenderInheritance**, 600

filter\_status\_type, 599  
 GetLoggerObject, 600  
 GetLogLevel, 600  
 GetNumAppenders, 600  
 GetParent, 601  
 Init, 601  
 IsActive, 601  
 log\_level\_type, 599  
 LogMsgBuffers, 601  
 LogMsgBuffersV, 601  
 LogMsgString, 602  
 LogMsgStringV, 602  
 message\_id\_type, 599  
 PVLogger, 599  
 PVLoggerRegistry, 603  
 RemoveAppender, 602  
 SetLogLevel, 602  
 SetLogLevelAndPropagate, 603  
 SetParent, 603  
**pvlogger.h, 825**  
   \_PVLOGGER\_LOGBIN, 827  
   \_PVLOGGER\_LOGBIN\_V, 827  
   \_PVLOGGER\_LOGMSG, 827  
   \_PVLOGGER\_LOGMSG\_V, 827  
   PVLOGGER\_ENABLE, 827  
   PVLOGGER\_INST\_LEVEL, 828  
   PVLOGGER\_INST\_LEVEL\_SUPPORT, 828  
   PVLOGGER\_LEVEL\_UNINITIALIZED, 831  
   PVLOGGER\_LOG\_USE\_ONLY, 828  
   PVLOGGER\_LOGBIN, 828  
   PVLOGGER\_LOGBIN\_PVLOGMSG\_-  
     INST\_HLDBG, 828  
   PVLOGGER\_LOGBIN\_PVLOGMSG\_-  
     INST\_LLDBG, 829  
   PVLOGGER\_LOGBIN\_PVLOGMSG\_-  
     INST\_MLDBG, 829  
   PVLOGGER\_LOGBIN\_PVLOGMSG\_-  
     INST\_PROF, 829  
   PVLOGGER\_LOGBIN\_PVLOGMSG\_-  
     INST\_REL, 829  
   PVLOGGER\_LOGBIN\_V, 829  
   PVLOGGER\_LOGBIN\_V\_-  
     PVLOGMSG\_INST\_HLDBG, 829  
   PVLOGGER\_LOGBIN\_V\_-  
     PVLOGMSG\_INST\_LLDBG, 829  
   PVLOGGER\_LOGBIN\_V\_-  
     PVLOGMSG\_INST\_PROF, 829  
   PVLOGGER\_LOGBIN\_V\_-  
     PVLOGMSG\_INST\_REL, 829  
   PVLOGGER\_LOGBIN\_V\_-  
     PVLOGMSG\_V\_INST\_MLDBG,  
       829  
           PVLOGGER\_LOGMSG, 829  
           PVLOGGER\_LOGMSG\_PVLOGMSG\_-  
             INST\_HLDBG, 829  
           PVLOGGER\_LOGMSG\_PVLOGMSG\_-  
             INST\_LLDBG, 830  
           PVLOGGER\_LOGMSG\_PVLOGMSG\_-  
             INST\_MLDBG, 830  
           PVLOGGER\_LOGMSG\_PVLOGMSG\_-  
             INST\_PROF, 830  
           PVLOGGER\_LOGMSG\_PVLOGMSG\_-  
             INST\_REL, 830  
           PVLOGGER\_LOGMSG\_V, 830  
           PVLOGGER\_LOGMSG\_V\_-  
             PVLOGMSG\_INST\_HLDBG, 830  
           PVLOGGER\_LOGMSG\_V\_-  
             PVLOGMSG\_INST\_LLDBG, 830  
           PVLOGGER\_LOGMSG\_V\_-  
             PVLOGMSG\_INST\_MLDBG,  
               830  
           PVLOGGER\_LOGMSG\_V\_-  
             PVLOGMSG\_INST\_PROF, 830  
           PVLOGGER\_LOGMSG\_V\_-  
             PVLOGMSG\_INST\_REL, 830  
           PVLOGMSG\_ALERT, 831  
           PVLOGMSG\_CRIT, 831  
           PVLOGMSG\_DEBUG, 831  
           PVLOGMSG\_EMERG, 831  
           PVLOGMSG\_ERR, 831  
           PVLOGMSG\_FATAL\_ERROR, 831  
           PVLOGMSG\_INFO, 832  
           PVLOGMSG\_INST\_HLDBG, 830  
           PVLOGMSG\_INST\_LLDBG, 830  
           PVLOGMSG\_INST\_MLDBG, 830  
           PVLOGMSG\_INST\_PROF, 831  
           PVLOGMSG\_INST\_REL, 831  
           PVLOGMSG\_NONFATAL\_ERROR, 832  
           PVLOGMSG\_NOTICE, 832  
           PVLOGMSG\_STACK\_TRACE, 832  
           PVLOGMSG\_STATISTIC, 832  
           PVLOGMSG\_VERBOSE, 832  
           PVLOGMSG\_WARNING, 832  
**pvlogger\_accessories.h, 833**  
   PVLOGGER\_FILTER\_ACCEPT, 833  
   PVLOGGER\_FILTER\_NEUTRAL, 833  
   PVLOGGER\_FILTER\_REJECT, 833  
**pvlogger\_c.h, 834**  
   PVLOGGER\_C\_INST\_LEVEL, 835  
   pvLogger\_GetLoggerObject, 835  
   pvLogger\_IsActive, 835  
   pvLogger\_LogMsgString, 835  
   PVLOGMSG\_C\_ALERT, 835  
   PVLOGMSG\_C\_CRIT, 835  
   PVLOGMSG\_C\_EMERG, 835  
   PVLOGMSG\_C\_ERR, 835

PVLOGMSG\_C\_INFO, 835  
 PVLOGMSG\_C\_INST\_LLDBG, 835  
 PVLOGMSG\_C\_INST\_LLDBG, 835  
 PVLOGMSG\_C\_INST\_LLDBG, 835  
 PVLOGMSG\_C\_INST\_PROF, 835  
 PVLOGMSG\_C\_INST\_REL, 835  
 PVLOGMSG\_C\_NOTICE, 835  
 PVLOGMSG\_C\_STACK\_DEBUG, 835  
 PVLOGMSG\_C\_STACK\_TRACE, 835  
 PVLOGMSG\_C\_WARNING, 835  
**PVLOGGER\_C\_INST\_LEVEL**  
 pvlogger\_c.h, 835  
**PVLOGGER\_ENABLE**  
 pvlogger.h, 827  
**PVLOGGER\_FILTER\_ACCEPT**  
 pvlogger\_accessories.h, 833  
**PVLOGGER\_FILTER\_NEUTRAL**  
 pvlogger\_accessories.h, 833  
**PVLOGGER\_FILTER\_REJECT**  
 pvlogger\_accessories.h, 833  
**pvLogger\_GetLoggerObject**  
 pvlogger\_c.h, 835  
**PVLOGGER\_INST\_LEVEL**  
 pvlogger.h, 828  
**PVLOGGER\_INST\_LEVEL\_SUPPORT**  
 pvlogger.h, 828  
**pvLogger\_IsActive**  
 pvlogger\_c.h, 835  
**PVLOGGER\_LEVEL\_UNINITIALIZED**  
 pvlogger.h, 831  
**PVLOGGER\_LOG\_USE\_ONLY**  
 pvlogger.h, 828  
**PVLOGGER\_LOGBIN**  
 pvlogger.h, 828  
**PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_LLDBG**  
 pvlogger.h, 828  
**PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_LLDBG**  
 pvlogger.h, 829  
**PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_MLDBG**  
 pvlogger.h, 829  
**PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_PROF**  
 pvlogger.h, 829  
**PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_REL**  
 pvlogger.h, 829  
**PVLOGGER\_LOGBIN\_V**  
 pvlogger.h, 829  
**PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_LLDBG**  
 pvlogger.h, 829

PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_LLDBG  
 pvlogger.h, 829  
 pvlogger.h, 829  
 PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_PROF  
 pvlogger.h, 829  
 PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_REL  
 pvlogger.h, 829  
 PVLOGGER\_LOGMSG  
 pvlogger.h, 829  
**PVLOGGER\_LOGMSG\_PVLOGMSG\_INST\_LLDBG**  
 pvlogger.h, 829  
**PVLOGGER\_LOGMSG\_PVLOGMSG\_INST\_LLDBG**  
 pvlogger.h, 830  
**PVLOGGER\_LOGMSG\_PVLOGMSG\_INST\_MLDBG**  
 pvlogger.h, 830  
**PVLOGGER\_LOGMSG\_PVLOGMSG\_INST\_PROF**  
 pvlogger.h, 830  
**PVLOGGER\_LOGMSG\_PVLOGMSG\_INST\_REL**  
 pvlogger.h, 830  
**PVLOGGER\_LOGMSG\_V**  
 pvlogger.h, 830  
**PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_INST\_LLDBG**  
 pvlogger.h, 830  
**PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_INST\_LLDBG**  
 pvlogger.h, 830  
**PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_INST\_MLDBG**  
 pvlogger.h, 830  
**PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_INST\_PROF**  
 pvlogger.h, 830  
**PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_INST\_REL**  
 pvlogger.h, 830  
**pvLogger\_LogMsgString**  
 pvlogger\_c.h, 835  
**pvlogger\_registry.h**, 836  
**PVLoggerAppender**, 604  
**PVLoggerAppender**  
 ~PVLoggerAppender, 604  
**AppendBuffers**, 604  
**AppendString**, 604

message\_id\_type, 604  
**PVLoggerFilter**, 605  
**PVLoggerFilter**  
 ~PVLoggerFilter, 606  
 filter\_status\_type, 605  
 FilterOpaqueMessge, 606  
 FilterString, 606  
 log\_level\_type, 605  
 message\_id\_type, 605  
**PVLoggerLayout**, 607  
**PVLoggerLayout**  
 ~PVLoggerLayout, 607  
 FormatOpaqueMessage, 607  
 FormatString, 607  
 message\_id\_type, 607  
**PVLoggerRegistry**, 609  
 PVLogger, 603  
 PVLoggerRegistry, 609  
**PVLoggerRegistry**  
 ~PVLoggerRegistry, 609  
 alloc\_type, 609  
 CreatePVLogger, 610  
 GetPVLoggerObject, 610  
 GetPVLoggerRegistry, 610  
 log\_level\_type, 609  
 PVLoggerRegistry, 609  
 SetNodeLogLevelExplicit, 610  
**PVLOGMSG\_ALERT**  
 pvlogger.h, 831  
**PVLOGMSG\_C\_ALERT**  
 pvlogger\_c.h, 835  
**PVLOGMSG\_C\_CRIT**  
 pvlogger\_c.h, 835  
**PVLOGMSG\_C\_EMERG**  
 pvlogger\_c.h, 835  
**PVLOGMSG\_C\_ERR**  
 pvlogger\_c.h, 835  
**PVLOGMSG\_C\_INFO**  
 pvlogger\_c.h, 835  
**PVLOGMSG\_C\_INST\_HLDBG**  
 pvlogger\_c.h, 835  
**PVLOGMSG\_C\_INST\_LLDBG**  
 pvlogger\_c.h, 835  
**PVLOGMSG\_C\_INST\_MLDBG**  
 pvlogger\_c.h, 835  
**PVLOGMSG\_C\_INST\_PROF**  
 pvlogger\_c.h, 835  
**PVLOGMSG\_C\_INST\_REL**  
 pvlogger\_c.h, 835  
**PVLOGMSG\_C\_NOTICE**  
 pvlogger\_c.h, 835  
**PVLOGMSG\_C\_STACK\_DEBUG**  
 pvlogger\_c.h, 835  
**PVLOGMSG\_C\_STACK\_TRACE**  
 pvlogger\_c.h, 835  
**pvlogger\_c.h**, 835  
**PVLOGMSG\_C\_WARNING**  
 pvlogger\_c.h, 835  
**PVLOGMSG\_CRIT**  
 pvlogger.h, 831  
**PVLOGMSG\_DEBUG**  
 pvlogger.h, 831  
**PVLOGMSG\_EMERG**  
 pvlogger.h, 831  
**PVLOGMSG\_ERR**  
 pvlogger.h, 831  
**PVLOGMSG\_FATAL\_ERROR**  
 pvlogger.h, 831  
**PVLOGMSG\_INFO**  
 pvlogger.h, 832  
**PVLOGMSG\_INST\_HLDBG**  
 pvlogger.h, 830  
**PVLOGMSG\_INST\_LLDBG**  
 pvlogger.h, 830  
**PVLOGMSG\_INST\_MLDBG**  
 pvlogger.h, 830  
**PVLOGMSG\_INST\_PROF**  
 pvlogger.h, 831  
**PVLOGMSG\_INST\_REL**  
 pvlogger.h, 831  
**PVLOGMSG\_NONFATAL\_ERROR**  
 pvlogger.h, 832  
**PVLOGMSG\_NOTICE**  
 pvlogger.h, 832  
**PVLOGMSG\_STACK\_TRACE**  
 pvlogger.h, 832  
**PVLOGMSG\_STATISTIC**  
 pvlogger.h, 832  
**PVLOGMSG\_VERBOSE**  
 pvlogger.h, 832  
**PVLOGMSG\_WARNING**  
 pvlogger.h, 832  
**PVMEM\_INST\_LEVEL**  
 osclbase, 32  
 osclconfig\_memory.h, 803  
**PVNETWORKADDRESS\_LEN**  
 oscl\_socket\_types.h, 755  
**PVOscBase\_Cleanup**  
 osclbase, 42  
**PVOscBase\_Init**  
 osclbase, 42  
**PVSCHEDNAMELEN**  
 osclproc, 101  
**PVSchedulerStopper**, 612  
 OsclExecSchedulerCommonBase, 390  
**PVSchedulerStopper**, 612  
**PVSchedulerStopper**  
 ~PVSchedulerStopper, 612  
**PVSchedulerStopper**, 612

PVSOCK\_ERR\_BAD\_PARAM  
     oscl\_socket\_imp\_pv.h, 738  
 PVSOCK\_ERR\_NOT\_IMPLEMENTED  
     oscl\_socket\_imp\_pv.h, 738  
 PVSOCK\_ERR\_SERV\_NOT\_CONNECTED  
     oscl\_socket\_imp\_pv.h, 738  
 PVSOCK\_ERR SOCK\_NO\_SERV  
     oscl\_socket\_imp\_pv.h, 738  
 PVSOCK\_ERR SOCK\_NOT\_CONNECTED  
     oscl\_socket\_imp\_pv.h, 738  
 PVSOCK\_ERR SOCK\_NOT\_OPEN  
     oscl\_socket\_imp\_pv.h, 738  
 PVSockBufRecv, 613  
     PVSockBufRecv, 613  
 PVSockBufRecv  
     iLen, 613  
     iMaxLen, 613  
     iPtr, 613  
     PVSockBufRecv, 613  
 PVSockBufSend, 614  
     PVSockBufSend, 614  
 PVSockBufSend  
     iLen, 614  
     iPtr, 614  
     PVSockBufSend, 614  
 PVThreadContext, 615  
     OsclExecSchedulerCommonBase, 390  
     PVThreadContext, 615  
 PVThreadContext  
     ~PVThreadContext, 615  
     EnterThreadContext, 615  
     ExitThreadContext, 615  
     Id, 615  
     IsSameThreadContext, 615  
     OsclActiveObject, 616  
     OsclCoeActiveScheduler, 616  
     OsclCoeActiveSchedulerBase, 616  
     OsclExecScheduler, 616  
     OsclExecSchedulerBase, 616  
     OsclExecSchedulerCommonBase, 616  
     OsclTimerObject, 616  
     PVActiveBase, 616  
     PVThreadContext, 615  
     ThreadHasScheduler, 616  
  
 QUE\_ITER\_BEGIN  
     osclproc, 101  
 QUE\_ITER\_END  
     osclproc, 101  
  
 Rand  
     OsclRand, 467  
 Read  
     Oscl\_File, 178  
  
 OsclAsyncFile, 309  
 OsclBinIStreamBigEndian, 318  
 OsclFileCache, 393  
 OsclNativeFile, 449  
 read  
     OSCL\_String, 254  
     OSCL\_wString, 297  
 Read\_uint16  
     OsclBinIStreamBigEndian, 318  
     OsclBinIStreamLittleEndian, 321  
 Read\_uint32  
     OsclBinIStreamBigEndian, 318  
     OsclBinIStreamLittleEndian, 321  
 Read\_uint8  
     OsclBinIStream, 315  
 ReadAsync  
     OsclNativeFile, 449  
 ReadAsyncCancel  
     OsclNativeFile, 449  
 rebalance  
     Oscl\_Rb\_Tree\_Base, 237  
 rebalance\_for\_erase  
     Oscl\_Rb\_Tree\_Base, 237  
 Recv  
     OsclRecvMethod, 476  
     OsclRecvRequest, 477  
     OsclSocketI, 523  
     OsclSocketIBase, 529  
     OsclTCPSocket, 553  
     OsclTCPSocketI, 557  
 RecvFrom  
     OsclRecvFromMethod, 472  
     OsclRecvFromRequest, 474  
     OsclSocketI, 523  
     OsclSocketIBase, 529  
     OsclUDPSocket, 587  
     OsclUDPSocketI, 590  
 RecvFromParam, 617  
     RecvFromParam, 617  
 RecvFromParam  
     iAddr, 617  
     iBufRecv, 617  
     iFlags, 617  
     iMultiMaxLen, 617  
     iPacketLen, 617  
     iPacketSource, 617  
     RecvFromParam, 617  
 RecvFromRequest  
     OsclRecvFromMethod, 472  
 RecvFromSuccess  
     OsclSocketI, 523  
     OsclSocketIBase, 529  
 RecvParam, 619  
     RecvParam, 619

RecvParam  
   *iBufRecv*, 619  
   *iFlags*, 619  
   *RecvParam*, 619  
 RecvRequest  
   *OsclRecvMethod*, 476  
 RecvSuccess  
   *OsclSocketI*, 523  
   *OsclSocketIBase*, 529  
 red  
   *Oscl\_Rb\_Tree\_Node\_Base*, 245  
 RedBl  
   *Oscl\_Rb\_Tree\_Node\_Base*, 245  
 refcount  
   *CHeapRep*, 126  
 reference  
   *Oscl\_Map*, 210  
   *Oscl\_Queue*, 228  
   *Oscl\_Rb\_Tree*, 235  
   *Oscl\_Rb\_Tree\_Const\_Iterator*, 239  
   *Oscl\_Rb\_Tree\_Iterator*, 242  
   *Oscl\_TagTree::const\_iterator*, 265  
   *Oscl\_TagTree::iterator*, 268  
   *Oscl\_TAlloc*, 273  
   *Oscl\_Vector*, 277  
 Register  
   *OsclComponentRegistry*, 335  
   *OsclRegistryClient*, 495  
   *OsclRegistryClientImpl*, 498  
   *OsclRegistryServTlsImpl*, 501  
 RegisterForCallback  
   *OsclExecScheduler*, 380  
   *OsclReadyQ*, 471  
 registerInstance  
   *OsclSingletonRegistry*, 520  
   *OsclTLSRegistry*, 579  
   *OsclTLSRegistryEx*, 580  
 registerInstanceAndUnlock  
   *OsclSingletonRegistry*, 520  
 release  
   *OsclExclusiveArrayPtr*, 373  
   *OsclExclusivePtr*, 376  
   *OsclExclusivePtrA*, 379  
   *OSCLMemAutoPtr*, 422  
 RELOCK\_MUTEX\_ERROR  
   *OsclProcStatus*, 462  
 Remove  
   *OsclDoubleLink*, 357  
   *OsclReadyQ*, 471  
   *OsclSocketServRequestList*, 546  
   *OsclTimerQ*, 574  
 remove  
   *OsclPriorityQueue*, 458  
   *OsclPriorityQueueBase*, 460  
   *remove\_element*  
     *Oscl\_Linked\_List*, 203  
     *Oscl\_Linked\_List\_Base*, 207  
     *Oscl\_MTLINKED\_LIST*, 219  
   *remove\_ref*  
     *CHeapRep*, 126  
   *removeALLAllocNodes*  
     *MM\_Audit\_Imp*, 154  
   *removeAllocNode*  
     *MM\_Audit\_Imp*, 154  
 RemoveAppender  
   *PVLogger*, 602  
 RemoveFixedCache  
   *Oscl\_File*, 178  
 RemoveFromScheduler  
   *OsclActiveObject*, 303  
   *OsclTimerObject*, 571  
   *PVActiveBase*, 595  
 RemoveRef  
   *DNSRequestParam*, 130  
 removeRef  
   *Oscl\_DefAllocWithRefCounter*, 169  
   *OsclMemPoolFixedChunkAllocator*, 431  
   *OsclMemPoolResizableAllocator*, 438  
   *OsclRefCounter*, 478  
   *OsclRefCounterDA*, 481  
   *OsclRefCounterMTDA*, 485  
   *OsclRefCounterMTSA*, 487  
   *OsclRefCounterSA*, 489  
 Request  
   *OsclTimer*, 566  
 RequestCanceled  
   *OsclExecSchedulerCommonBase*, 387  
 RequestDone  
   *OsclDNSRequestAO*, 355  
   *OsclSocketRequestAO*, 538  
 reserve  
   *Oscl\_Queue\_Base*, 231  
   *Oscl\_Vector\_Base*, 284  
   *OsclPriorityQueue*, 458  
 ReserveSpace  
   *OsclBinStream*, 329  
 Reset  
   *OsclDoubleListBase*, 360  
 reset  
   *BufferState*, 115  
   *MM\_FailInsertParam*, 157  
   *MM\_Stats\_t*, 160  
   *OsclMemStatsNode*, 444  
 ResetLogPerf  
   *OsclExecSchedulerCommonBase*, 387  
 Resume  
   *OsclThread*, 560  
 ResumeScheduler

OsclExecSchedulerCommonBase, 387  
 retrieveParentTag  
   MM\_Audit\_Imp, 154  
 retrieveParentTagLength  
   MM\_Audit\_Imp, 154  
 RFC822ToPV8601  
   osclbase, 42  
 Right  
   OsclPtrC, 466  
 right  
   Oscl\_Rb\_Tree\_Node\_Base, 246  
 rotate\_left  
   Oscl\_Rb\_Tree\_Base, 237  
 rotate\_right  
   Oscl\_Rb\_Tree\_Base, 237  
 Run  
   CallbackTimer, 120  
   OsclDNSMethod, 350  
   OsclDNSRequestAO, 355  
   OsclSocketMethod, 532  
   OsclSocketRequestAO, 538  
   PVActiveBase, 595  
 RunError  
   OsclActiveObject, 303  
   OsclTimerObject, 571  
   PVActiveBase, 595  
 RunIfNotReady  
   OsclActiveObject, 304  
   OsclTimerObject, 571  
 RunSchedulerNonBlocking  
   OsclExecScheduler, 380  
  
 save\_registry  
   TLSStorageOps, 636  
 second  
   Oscl\_Pair, 226  
 SECONDS  
   osclbase, 33  
 Seed  
   OsclRand, 467  
 Seek  
   Oscl\_File, 178  
   OsclAsyncFile, 309  
   OsclBinStream, 329  
   OsclFileCache, 393  
   OsclNativeFile, 450  
 seek\_type  
   Oscl\_File, 175  
 SEEKCUR  
   Oscl\_File, 175  
 SEEKEND  
   Oscl\_File, 175  
 seekFromCurrentPosition  
   OsclBinStream, 329  
  
 SEEKSET  
   Oscl\_File, 175  
 self  
   Oscl\_Map, 210  
   Oscl\_Rb\_Tree\_Const\_Iterator, 239  
   Oscl\_Rb\_Tree\_Iterator, 242  
   Oscl\_TagTree::const\_iterator, 265  
   Oscl\_TagTree::iterator, 268  
 SEM\_NOT\_SIGNALLED\_ERROR  
   OsclProcStatus, 462  
 Send  
   OsclSendMethod, 509  
   OsclSendRequest, 510  
   OsclSocketI, 523  
   OsclSocketIBase, 529  
   OsclTCPSocket, 554  
   OsclTCPSocketI, 557  
 SendParam, 620  
   SendParam, 620  
 SendParam  
   iBufSend, 620  
   iFlags, 620  
   iXferLen, 620  
   SendParam, 620  
 SendRequest  
   OsclSendMethod, 509  
 SendSuccess  
   OsclSocketI, 524  
   OsclSocketIBase, 529  
 SendTo  
   OsclSendToMethod, 511  
   OsclSendToRequest, 512  
   OsclSocketI, 524  
   OsclSocketIBase, 529  
   OsclUDPSocket, 587  
   OsclUDPSocketI, 590  
 SendToParam, 621  
   SendToParam, 621  
 SendToParam  
   ~SendToParam, 621  
   iAddr, 621  
   iBufSend, 621  
   iFlags, 621  
   iXferLen, 621  
   SendToParam, 621  
 SendToRequest  
   OsclSendToMethod, 511  
 SendToSuccess  
   OsclSocketI, 524  
   OsclSocketIBase, 529  
 Serv  
   OsclDNSRequestAO, 355  
 Set  
   OsclDoubleRunner, 361

OsclNameString, 447  
 OsclPtr, 463  
 OsclPtrC, 466  
 set  
   CHHeapRep, 126  
   CStackRep, 128  
   OSCL\_FastString, 172  
   OSCL\_HeapStringA, 196  
   OSCL\_wFastString, 286  
   OSCL\_wHeapStringA, 292  
   OsclExclusiveArrayPtr, 373  
   OsclExclusivePtr, 376  
   OsclExclusivePtrA, 379  
   OsclSingleton, 518  
   OsclTLS, 575  
   OsclTLSEx, 577  
   osclutil, 79, 80  
 set\_from\_ntp\_time  
   TimeValue, 634  
 set\_from\_system\_time  
   NTPTime, 164  
 set\_int64  
   Oscl\_Int64\_Utils, 198  
 set\_len  
   OSCL\_String, 254  
   OSCL\_wString, 298  
 set\_length  
   OSCL\_FastString, 172  
   OSCL\_wFastString, 286  
 set\_next  
   Oscl\_Opaque\_Type\_Alloc\_LL, 223  
 set\_r  
   CFastRep, 124  
 set\_rep  
   CHHeapRep, 126  
   OSCL\_String, 254, 255  
   OSCL\_wString, 298  
 set\_to\_current\_time  
   NTPTime, 164  
   TimeValue, 634  
 set\_to\_zero  
   TimeValue, 634  
 set\_uint64  
   Oscl\_Int64\_Utils, 198  
 set\_w  
   CFastRep, 124  
 setAllocNodeFlag  
   MM\_AllocBlockHdr, 143  
 SetAsyncReadBufferSize  
   Oscl\_File, 178  
 SetBusy  
   OsclActiveObject, 304  
   OsclTimerObject, 571  
 SetCacheObserver  
   Oscl\_File, 178  
 setCheckSum  
   StrCSumPtrLen, 626  
 SetExactFrequency  
   OsclTimer, 566  
 SetFileHandle  
   Oscl\_File, 179  
 SetFrequency  
   OsclTimer, 567  
 SetInUse  
   OsclAsyncFileBuffer, 311  
 SetLength  
   OsclPtr, 463  
   OsclPtrC, 466  
 SetLoggingEnable  
   Oscl\_File, 179  
 SetLogLevel  
   PVLogger, 602  
 SetLogLevelAndPropagate  
   PVLogger, 603  
 setMaxSzForNewMemPoolBuffer  
   OsclMemPoolResizableAllocator, 438  
 SetNativeAccessMode  
   Oscl\_File, 179  
 SetNativeBufferSize  
   Oscl\_File, 179  
 SetNodeLogLevelExplicit  
   PVLoggerRegistry, 610  
 SetObserver  
   OsclTimer, 567  
 SetOffset  
   OsclAsyncFileBuffer, 311  
   OsclDoubleListBase, 360  
 SetParent  
   PVLogger, 603  
 SetPosition  
   OsclFileCacheBuffer, 395  
 SetPriority  
   OsclThread, 560  
 setPtrLen  
   StrCSumPtrLen, 626  
   StrPtrLen, 629  
   WStrPtrLen, 639  
 SetPVCacheSize  
   Oscl\_File, 180  
 SetRecvBufferSize  
   OsclIPSocketI, 405  
   OsclSocketI, 524  
   OsclUDPSocket, 587  
 SetScheduler  
   OsclExecSchedulerCommonBase, 387  
 SetStatus  
   OsclActiveObject, 304  
   OsclTimerObject, 571

SetSummaryStatsLoggingEnable  
     Oscl\_File, 180

SetTimestamp  
     MediaData, 139

SetToHead  
     OsclDoubleRunner, 361

SetToTail  
     OsclDoubleRunner, 361

setWithoutOwnership  
     OSCLMemAutoPtr, 422

ShowStats  
     OsclExecSchedulerCommonBase, 387

ShowSummaryStats  
     OsclExecSchedulerCommonBase, 387

Shutdown  
     OsclShutdownMethod, 516

OsclShutdownRequest, 517

OsclSocketI, 524

OsclSocketIBase, 530

OsclTCPSocket, 554

OsclTCPSocketI, 557

ShutdownParam, 622

    ShutdownParam, 622

ShutdownParam  
     iHow, 622

    ShutdownParam, 622

ShutdownRequest  
     OsclShutdownMethod, 516

Signal  
     OsclSemaphore, 508

Size  
     Oscl\_File, 180

OsclAsyncFile, 309

OsclNativeFile, 450

size  
     CFastRep, 124

CHheapRep, 126

CStackRep, 128

MM\_AllocBlockHdr, 143

MM\_AllocInfo, 145

MM\_AllocQueryInfo, 147

Oscl\_Map, 213

Oscl\_Queue\_Base, 231

Oscl\_Rb\_Tree, 235

Oscl\_TagTree, 263

Oscl\_Vector\_Base, 284

OsclPriorityQueue, 458

StrPtrLen, 629

WStrPtrLen, 639

size\_type  
     Oscl\_Map, 210

Oscl\_Queue, 228

Oscl\_Rb\_Tree, 235

Oscl\_Tag\_Base, 259

Oscl\_TagTree, 261

Oscl\_TAlloc, 273

sizeof\_T  
     Oscl\_Linked\_List\_Base, 207

Oscl\_Queue\_Base, 232

Oscl\_Vector\_Base, 284

skip\_to\_line\_term  
     osclutil, 80

skip\_to\_whitespace  
     osclutil, 80

skip\_whitespace  
     osclutil, 80

skip\_whitespace\_and\_line\_term  
     osclutil, 80

SLEEP\_ONE\_SEC  
     osclconfig\_util.h, 823

SleepMillisec  
     OsclThread, 560

Socket  
     OsclSocketI, 524

SocketI  
     OsclSocketRequestAO, 538

SocketObserver  
     OsclSocketRequestAO, 538

SocketRequestParam, 623

    SocketRequestParam, 624

SocketRequestParam  
     iFxn, 624

    SocketRequestParam, 624

SocketServ  
     OsclIPSocketI, 405

sort\_children  
     Oscl\_TagTree::Node, 271

specialFragBuffer  
     OsclBinStream, 330

Start  
     OsclFileStats, 397

Start\_on\_creation  
     oscl\_thread.h, 769

StartAsyncRead  
     OsclAsyncFileBuffer, 311

StartCancel  
     OsclSocketServRequestList, 546

StartMethod  
     OsclDNSMethod, 350

OsclSocketMethod, 533

StartNativeScheduler  
     OsclExecSchedulerCommonBase, 387

StartScheduler  
     OsclExecSchedulerCommonBase, 387

State  
     OsclSocketServIBase, 545

state  
     OsclBinStream, 330

state\_t  
     OsclBinStream, 328  
 StaticJump  
     OsclJump, 406  
 stats\_overhead  
     MM\_AuditOverheadStats, 156  
 Status  
     OsclActiveObject, 304  
     OsclTimerObject, 572  
 status\_t  
     BufFragStatusClass, 119  
 StatusRef  
     OsclActiveObject, 304  
     OsclTimerObject, 572  
 StopScheduler  
     OsclExecSchedulerCommonBase, 387  
 Str  
     OsclNameString, 447  
 StrCSumPtrLen, 625  
     osclutil, 66  
 StrCSumPtrLen, 626  
 StrCSumPtrLen  
     checkSum, 626  
     CheckSumType, 626  
     getCheckSum, 626  
     isCIEquivalentTo, 626  
     operator!=, 626  
     operator=, 626  
     operator==, 626  
     setCheckSum, 626  
     setPtrLen, 626  
     StrCSumPtrLen, 626  
 StrPtrLen, 628  
     osclutil, 66  
     StrPtrLen, 629  
 StrPtrLen  
     c\_str, 629  
     isCIEquivalentTo, 629  
     isCIPrefixOf, 629  
     isLetter, 629  
     len, 629  
     length, 629  
     operator!=, 629  
     operator=, 629  
     operator==, 629  
     ptr, 629  
     setPtrLen, 629  
     size, 629  
     StrPtrLen, 629  
 Success  
     OsclDNSRequestAO, 356  
     OsclRecvFromRequest, 474  
     OsclRecvRequest, 477  
     OsclSendRequest, 510  
     OsclSendToRequest, 512  
     OsclSocketRequestAO, 538  
 SUCCESS\_ERROR  
     OsclProcStatus, 461  
 Suspend  
     OsclThread, 561  
 Suspend\_on\_creation  
     oscl\_thread.h, 769  
 SuspendScheduler  
     OsclExecSchedulerCommonBase, 388  
 swap  
     Oscl\_Opaque\_Type\_Compare, 224  
     OsclPriorityQueue, 458  
 SYSTEM\_RESOURCES\_UNAVAILABLE\_-  
     ERROR  
     OsclProcStatus, 462  
 tag  
     MM\_AllocQueryInfo, 147  
     MM\_Stats\_CB, 158  
     Oscl\_Tag, 256  
     Oscl\_TagTree::Node, 271  
     OsclMemStatsNode, 444  
 tag\_ancestor  
     Oscl\_Tag\_Base, 259  
 tag\_base\_type  
     Oscl\_Tag\_Base, 259  
     Oscl\_TagTree, 261  
 tag\_base\_unit  
     Oscl\_Tag\_Base, 259  
 tag\_cmp  
     Oscl\_Tag\_Base, 259  
 tag\_copy  
     Oscl\_Tag\_Base, 259  
 tag\_depth  
     Oscl\_Tag\_Base, 259  
 tag\_len  
     Oscl\_Tag\_Base, 259  
 tag\_type  
     Oscl\_TagTree, 261  
 tagAllocator  
     Oscl\_Tag, 256  
 TagTree\_Allocator  
     osclmemory, 56  
 Tail  
     OsclDoubleList, 358  
     OsclPriorityList, 455  
 tail  
     Oscl\_Linked\_List\_Base, 207  
 takeOwnership  
     OSCLMemAutoPtr, 423  
 TDNSRequestParamAllocator  
     oscl\_dns\_param.h, 655  
 Tell

Oscl\_File, 180  
 OsclAsyncFile, 309  
 OsclFileCache, 393  
 OsclNativeFile, 450  
 tellg  
   OsclBinStream, 329  
 Terminate  
   OsclThread, 561  
 the\_list  
   Oscl\_MTLinked\_List, 219  
 THREAD\_1\_INACTIVE\_ERROR  
   OsclProcStatus, 461  
 THREAD\_BLOCK\_ERROR  
   OsclProcStatus, 462  
 THREAD\_NOT\_OWN\_MUTEX\_ERROR  
   OsclProcStatus, 462  
 ThreadHasScheduler  
   PVThreadContext, 616  
 ThreadLogoff  
   OsclReadyQ, 471  
 ThreadLogon  
   OsclReadyQ, 471  
 ThreadPriorityAboveNormal  
   oscl\_thread.h, 770  
 ThreadPriorityBelowNormal  
   oscl\_thread.h, 769  
 ThreadPriorityHighest  
   oscl\_thread.h, 770  
 ThreadPriorityLow  
   oscl\_thread.h, 769  
 ThreadPriorityLowest  
   oscl\_thread.h, 769  
 ThreadPriorityNormal  
   oscl\_thread.h, 769  
 ThreadPriorityTimeCritical  
   oscl\_thread.h, 770  
 TickCount  
   OsclTickCount, 563  
 TickCountFrequency  
   OsclTickCount, 563  
 TickCountPeriod  
   OsclTickCount, 563  
 TicksToMsec  
   OsclTickCount, 563  
 TimeoutOccurred  
   OsclTimerObserver, 573  
 TimerBaseElapsed  
   CallbackTimerObserver, 122  
   OsclTimer, 567  
 TimerCallback  
   OsclReadyQ, 471  
 timestamp  
   MediaData, 139  
 TimeUnits  
   osclbase, 33  
   TimeValue, 630  
     TimeValue, 631, 632  
 TimeValue  
   get\_local\_time, 632  
   get\_pv8601\_str\_time, 632  
   get\_rfc822\_gmtime\_str, 632  
   get\_sec, 633  
   get\_str\_ctime, 633  
   get\_timeval\_ptr, 633  
   get\_usec, 633  
   is\_zero, 633  
   NTPTime, 635  
   operator \*=, 634  
   operator!=, 635  
   operator+=, 634  
   operator-=, 634  
   operator<, 635  
   operator<=, 635  
   operator=, 634  
   operator==, 635  
   operator>, 635  
   operator>=, 635  
   set\_from\_ntp\_time, 634  
   set\_to\_current\_time, 634  
   set\_to\_zero, 634  
   TimeValue, 631, 632  
   to\_msec, 634  
 TLSStorageOps, 636  
 TLSStorageOps  
   get\_registry, 636  
   save\_registry, 636  
 to\_msec  
   TimeValue, 634  
 to\_system\_time  
   NTPTime, 164  
 TOO\_MANY\_FRAGS  
   BuffFragStatusClass, 119  
 TOO\_MANY\_THREADS\_ERROR  
   OsclProcStatus, 461  
 Top  
   OsclJump, 406  
   OsclReadyQ, 471  
   OsclTimerQ, 574  
 top  
   OsclPriorityQueue, 458  
 TOsclBasicLockObject  
   osclconfig\_unix\_android.h, 818  
   osclconfig\_unix\_common.h, 822  
 TOsclConditionObject  
   osclconfig\_proc\_unix\_android.h, 810  
   osclconfig\_proc\_unix\_common.h, 812  
 TOsclFileHandle  
   osclio, 93

**T**OscIFileOffset  
     osclconfig\_io.h, [797](#)  
**T**OscIFileOffsetInt32  
     osclio, [93](#)  
**T**OscIFileOp  
     osclio, [94](#)  
**T**OscIHostent  
     osclconfig\_io.h, [797](#)  
**T**OscIMutexObject  
     osclconfig\_proc\_unix\_android.h, [810](#)  
     osclconfig\_proc\_unix\_common.h, [812](#)  
**T**OscIReady  
     osclproc, [102](#)  
**T**OscISemaphoreObject  
     osclconfig\_proc\_unix\_android.h, [810](#)  
     osclconfig\_proc\_unix\_common.h, [812](#)  
**T**OscISockAddr  
     osclconfig\_io.h, [797](#)  
**T**OscISockAddrLen  
     osclconfig\_io.h, [797](#)  
**T**OscISocket  
     osclconfig\_io.h, [797](#)  
**T**OscISocketServStatEvent  
     oscl\_socket\_stats.h, [751](#)  
**T**OscISocketStatEvent  
     oscl\_socket\_stats.h, [751](#)  
**T**OscIThreadFuncArg  
     osclconfig\_proc\_unix\_android.h, [810](#)  
     osclconfig\_proc\_unix\_common.h, [812](#)  
**T**OscIThreadFuncPtr  
     oscl\_thread.h, [769](#)  
**T**OscIThreadFuncRet  
     osclconfig\_proc\_unix\_android.h, [810](#)  
     osclconfig\_proc\_unix\_common.h, [812](#)  
**T**OscIThreadId  
     osclconfig\_proc\_unix\_android.h, [810](#)  
     osclconfig\_proc\_unix\_common.h, [812](#)  
**T**OscIThreadObject  
     osclconfig\_proc\_unix\_android.h, [810](#)  
     osclconfig\_proc\_unix\_common.h, [812](#)  
**T**OscITlsKey  
     osclbase, [33](#)  
     osclconfig\_unix\_android.h, [818](#)  
     osclconfig\_unix\_common.h, [822](#)  
**totalbytes**  
     oscl\_fssstat, [190](#)  
**totalNumAllocs**  
     MM\_Stats\_t, [160](#)  
**totalNumBytes**  
     MM\_Stats\_t, [160](#)  
**T**OtherExecStats  
     OsclExecSchedulerCommonBase, [385](#)  
**T**PVDNSEvent  
     osclio, [95](#)  
**T**PVDNSFxn  
     osclio, [95](#)  
**T**PVSocketEvent  
     oscl\_socket\_types.h, [755](#)  
**T**PVSocketFxn  
     oscl\_socket\_types.h, [755](#)  
**T**PVSocketShutdown  
     oscl\_socket\_types.h, [756](#)  
**T**PVThreadContext  
     osclproc, [102](#)  
**Trap**  
     OsclErrorTrapImp, [368](#)  
**TrapNoTls**  
     OsclErrorTrapImp, [368](#)  
**T**ReadyQueLink, [637](#)  
     TReadyQueLink, [637](#)  
**T**ReadyQueLink  
     iAOPriority, [637](#)  
     iIsIn, [637](#)  
     iSeqNum, [637](#)  
     iTimeQueuedTicks, [637](#)  
     iTimeToRunTicks, [637](#)  
     TReadyQueLink, [637](#)  
**trim**  
     OsclMemPoolResizableAllocator, [438](#)  
**TryLock**  
     OsclMutex, [446](#)  
**TryWait**  
     OsclSemaphore, [508](#)  
**TSocketServState**  
     OsclSocketServIBase, [544](#)  
**TSymbianAccessMode**  
     Oscl\_File, [175](#)  
**uint**  
     osclbase, [33](#)  
**UINT64**  
     osclconfig\_unix\_android.h, [818](#)  
     osclconfig\_unix\_common.h, [822](#)  
**uint64**  
     osclbase, [33](#)  
**UINT64\_HILO**  
     osclconfig\_unix\_android.h, [818](#)  
     osclconfig\_unix\_common.h, [822](#)  
**Unbind**  
     OsclSharedPtr, [515](#)  
**UninstallScheduler**  
     OsclExecSchedulerCommonBase, [388](#)  
**unix\_ntp\_offset**  
     osclbase, [43](#)  
**Unlock**  
     OsclLockBase, [409](#)  
     OsclMutex, [446](#)  
     OsclNullLock, [453](#)

OsclThreadLock, 562  
 UnRegister  
   OsclRegistryClient, 496  
   OsclRegistryClientImpl, 498  
   OsclRegistryServTlsImpl, 501  
 Unregister  
   OsclComponentRegistry, 335  
 UnTrap  
   OsclErrorTrapImp, 368  
 update  
   MM\_Stats\_t, 160  
 UpdateData  
   OsclAsyncFileBuffer, 311  
 updateEnd  
   OsclFileCacheBuffer, 395  
 updateStart  
   OsclFileCacheBuffer, 395  
 updateStatsNode  
   MM\_Audit\_Imp, 154  
 updateStatsNodeInFailure  
   MM\_Audit\_Imp, 154  
 UpdateTimers  
   OsclExecSchedulerCommonBase, 388  
 UpdateTimersMsec  
   OsclExecSchedulerCommonBase, 388  
 upper\_bound  
   Oscl\_Map, 213, 214  
   Oscl\_Rb\_Tree, 235  
 usableSize  
   OsclFileCacheBuffer, 395  
 USEC\_PER\_SEC  
   osclbase, 43  
  
 validate  
   MM\_Audit\_Imp, 154  
   OsclPriorityQueue, 459  
 validate\_all\_heap  
   MM\_Audit\_Imp, 154  
 validateblock  
   OsclMemPoolResizableAllocator, 438  
 Value  
   OsclAOStatus, 306  
 value  
   Oscl\_Rb\_Tree\_Node, 244  
   Oscl\_TagTree::Node, 271  
 value\_comp  
   Oscl\_Map, 214  
 value\_compare  
   Oscl\_Map::value\_compare, 215  
 value\_type  
   Oscl\_Map, 210  
   Oscl\_Queue, 228  
   Oscl\_Rb\_Tree, 235  
   Oscl\_Rb\_Tree\_Const\_Iterator, 239  
  
 Oscl\_Rb\_Tree\_Iterator, 242  
 Oscl\_Rb\_Tree\_Node, 244  
 Oscl\_TagTree, 261  
 Oscl\_TAlloc, 273  
 Oscl\_Vector, 277  
 OsclPriorityQueue, 457  
 vec  
   OsclPriorityQueue, 459  
  
 Wait  
   OsclSemaphore, 508  
 WAIT\_ABANDONED\_ERROR  
   OsclProcStatus, 462  
 WAIT\_TIMEOUT\_ERROR  
   OsclProcStatus, 462  
 WaitAndPopTop  
   OsclReadyQ, 471  
 WaitForReadyAO  
   OsclExecSchedulerCommonBase, 388  
 WaitForRequestComplete  
   OsclReadyQ, 471  
 WaitOnRequests  
   OsclSocketServRequestList, 546  
 Wakeup  
   OsclSocketServRequestList, 546  
 writable  
   CFastRep, 124  
 Write  
   Oscl\_File, 180  
   OsclAsyncFile, 309  
   OsclFileCache, 393  
   OsclNativeFile, 450  
 write  
   OSCL\_String, 255  
   OSCL\_wString, 298  
   OsclBinOStream, 322  
 WriteUnsignedLong  
   OsclBinOStreamBigEndian, 324  
   OsclBinOStreamLittleEndian, 326  
 WriteUnsignedShort  
   OsclBinOStreamBigEndian, 324  
   OsclBinOStreamLittleEndian, 326  
 WriteUpdatesToFile  
   OsclFileCacheBuffer, 395  
 WStrPtrLen, 638  
   osclutil, 66  
   WStrPtrLen, 639  
 WStrPtrLen  
   c\_str, 639  
   isCIEquivalentTo, 639  
   len, 639  
   length, 639  
   operator!=, 639  
   operator=, 639

operator==, [639](#)

ptr, [639](#)

setPtrLen, [639](#)

size, [639](#)

WStrPtrLen, [639](#)

xsubi

MM\_FailInsertParam, [157](#)

Zero

OsclPtr, [463](#)

OsclPtrC, [466](#)