



packetvideo™

OSCL API

Build Version: CORE_8.000.1.1_RC4

November 1, 2009

Contents

| | | |
|----------|---|------------|
| 1 | oscl Module Index | 1 |
| 1.1 | oscl Modules | 1 |
| 2 | oscl Hierarchical Index | 2 |
| 2.1 | oscl Class Hierarchy | 2 |
| 3 | oscl Data Structure Index | 9 |
| 3.1 | oscl Data Structures | 9 |
| 4 | oscl File Index | 15 |
| 4.1 | oscl File List | 15 |
| 5 | oscl Page Index | 20 |
| 5.1 | oscl Related Pages | 20 |
| 6 | oscl Module Documentation | 21 |
| 6.1 | OSCL config | 21 |
| 6.2 | OSCL Base | 25 |
| 6.3 | OSCL Memory | 46 |
| 6.4 | OSCL Util | 62 |
| 6.5 | OSCL Error | 84 |
| 6.6 | OSCL IO | 94 |
| 6.7 | OSCL Proc | 102 |
| 6.8 | OSCL Init | 106 |
| 7 | oscl Data Structure Documentation | 107 |
| 7.1 | _OscIBasicAllocator Class Reference | 107 |
| 7.2 | _OscIHeapBase Class Reference | 109 |
| 7.3 | AcceptParam Class Reference | 111 |
| 7.4 | allocator Class Reference | 112 |

| | |
|---|-----|
| 7.5 AllPassFilter Class Reference | 113 |
| 7.6 BindParam Class Reference | 115 |
| 7.7 BufferFragment Class Reference | 116 |
| 7.8 BufferMgr Class Reference | 117 |
| 7.9 BufferState Class Reference | 118 |
| 7.10 BufFragGroup< ChainClass, max_frags > Class Template Reference | 119 |
| 7.11 BufFragStatusClass Class Reference | 122 |
| 7.12 CallbackTimer< Alloc > Class Template Reference | 123 |
| 7.13 CallbackTimerObserver Class Reference | 125 |
| 7.14 CFastRep Class Reference | 126 |
| 7.15 CHeapRep Class Reference | 128 |
| 7.16 ConnectParam Class Reference | 130 |
| 7.17 CStackRep Class Reference | 131 |
| 7.18 DNSRequestParam Class Reference | 132 |
| 7.19 GetHostByNameParam Class Reference | 134 |
| 7.20 HeapBase Class Reference | 136 |
| 7.21 internalLeave Class Reference | 138 |
| 7.22 LinkedListElement< LLClass > Class Template Reference | 139 |
| 7.23 ListenParam Class Reference | 140 |
| 7.24 MediaData< ChainClass, max_frags, local_bufsize > Class Template Reference | 141 |
| 7.25 MediaStatusClass Class Reference | 144 |
| 7.26 MemAllocator< T > Class Template Reference | 145 |
| 7.27 MM_AllocBlockFence Struct Reference | 146 |
| 7.28 MM_AllocBlockHdr Struct Reference | 147 |
| 7.29 MM_AllocInfo Struct Reference | 148 |
| 7.30 MM_AllocNode Struct Reference | 150 |
| 7.31 MM_AllocQueryInfo Struct Reference | 151 |
| 7.32 MM_Audit_Imp Class Reference | 152 |
| 7.33 MM_AuditOverheadStats Struct Reference | 160 |
| 7.34 MM_FailInsertParam Struct Reference | 161 |
| 7.35 MM_Stats_CB Struct Reference | 162 |
| 7.36 MM_Stats_t Struct Reference | 163 |
| 7.37 NTPTIME Class Reference | 165 |
| 7.38 Osci_Alloc Class Reference | 169 |
| 7.39 Osci_Dealloc Class Reference | 170 |
| 7.40 Osci_DefAlloc Class Reference | 171 |

| | | |
|------|---|-----|
| 7.41 | OscL_DefAllocWithRefCount< DefAlloc > Class Template Reference | 172 |
| 7.42 | OSCL_FastString Class Reference | 174 |
| 7.43 | OscL_File Class Reference | 178 |
| 7.44 | OscL_File::OscLCacheObserver Class Reference | 186 |
| 7.45 | OscL_File::OscLFixedCacheParam Class Reference | 187 |
| 7.46 | OscL_FileFind Class Reference | 188 |
| 7.47 | OscL_FileServer Class Reference | 192 |
| 7.48 | oscl_fsstat Struct Reference | 194 |
| 7.49 | OSCL_HeapString< Alloc > Class Template Reference | 195 |
| 7.50 | OSCL_HeapStringA Class Reference | 197 |
| 7.51 | OscL_Int64_Utils Class Reference | 202 |
| 7.52 | OscL_Less< T > Struct Template Reference | 204 |
| 7.53 | OscL_Linked_List< LLClass, Alloc > Class Template Reference | 205 |
| 7.54 | OscL_Linked_List_Base Class Reference | 210 |
| 7.55 | OscL_Map< Key, T, Alloc, Compare > Class Template Reference | 215 |
| 7.56 | OscL_Map< Key, T, Alloc, Compare >::value_compare Class Reference | 222 |
| 7.57 | OscL_MTLinked_List< LLClass, Alloc, TheLock > Class Template Reference | 224 |
| 7.58 | OscL_Opaque_Type_Alloc Class Reference | 228 |
| 7.59 | OscL_Opaque_Type_Alloc_LL Class Reference | 229 |
| 7.60 | OscL_Opaque_Type_Compare Class Reference | 231 |
| 7.61 | OscL_Pair< T1, T2 > Struct Template Reference | 233 |
| 7.62 | OscL_Queue< T, Alloc > Class Template Reference | 234 |
| 7.63 | OscL_Queue_Base Class Reference | 237 |
| 7.64 | OscL_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > Class Template Reference | 240 |
| 7.65 | OscL_Rb_Tree_Base Class Reference | 244 |
| 7.66 | OscL_Rb_Tree_Const_Iterator< Value > Struct Template Reference | 245 |
| 7.67 | OscL_Rb_Tree_Iterator< Value > Struct Template Reference | 248 |
| 7.68 | OscL_Rb_Tree_Node< Value > Struct Template Reference | 251 |
| 7.69 | OscL_Rb_Tree_Node_Base Struct Reference | 252 |
| 7.70 | OscL_SelectIst< V, U > Struct Template Reference | 254 |
| 7.71 | OSCL_StackString< MaxBufSize > Class Template Reference | 255 |
| 7.72 | oscl_stat_buf Struct Reference | 257 |
| 7.73 | OSCL_String Class Reference | 258 |
| 7.74 | OscL_Tag< Alloc > Struct Template Reference | 263 |
| 7.75 | OscL_Tag_Base Struct Reference | 265 |
| 7.76 | OscL_TagTree< T, Alloc > Class Template Reference | 267 |

| | | |
|-------|---|-----|
| 7.77 | OscI_TagTree< T, Alloc >::const_iterator Struct Reference | 271 |
| 7.78 | OscI_TagTree< T, Alloc >::iterator Struct Reference | 274 |
| 7.79 | OscI_TagTree< T, Alloc >::Node Struct Reference | 277 |
| 7.80 | OscI_TAlloc< T, Alloc > Class Template Reference | 279 |
| 7.81 | OscI_TAlloc< T, Alloc >::rebind< U, V > Struct Template Reference | 282 |
| 7.82 | OscI_Vector< T, Alloc > Class Template Reference | 283 |
| 7.83 | OscI_Vector_Base Class Reference | 288 |
| 7.84 | OSCL_wFastString Class Reference | 292 |
| 7.85 | OSCL_wHeapString< Alloc > Class Template Reference | 295 |
| 7.86 | OSCL_wHeapStringA Class Reference | 297 |
| 7.87 | OSCL_wStackString< MaxBufSize > Class Template Reference | 300 |
| 7.88 | OSCL_wString Class Reference | 302 |
| 7.89 | OscIAcceptMethod Class Reference | 306 |
| 7.90 | OscIAcceptRequest Class Reference | 307 |
| 7.91 | OscIActiveObject Class Reference | 308 |
| 7.92 | OscIAllocDestructDealloc Class Reference | 312 |
| 7.93 | OscIAOStatus Class Reference | 313 |
| 7.94 | OscIAsyncFile Class Reference | 314 |
| 7.95 | OscIAsyncFileBuffer Class Reference | 317 |
| 7.96 | OscIAuditCB Class Reference | 319 |
| 7.97 | OscIBindMethod Class Reference | 320 |
| 7.98 | OscIBindRequest Class Reference | 321 |
| 7.99 | OscIBinIStream Class Reference | 322 |
| 7.100 | OscIBinIStreamBigEndian Class Reference | 324 |
| 7.101 | OscIBinIStreamLittleEndian Class Reference | 327 |
| 7.102 | OscIBinOStream Class Reference | 329 |
| 7.103 | OscIBinOStreamBigEndian Class Reference | 330 |
| 7.104 | OscIBinOStreamLittleEndian Class Reference | 332 |
| 7.105 | OscIBinStream Class Reference | 334 |
| 7.106 | OscIBuf Class Reference | 338 |
| 7.107 | OscICompareLess< T > Class Template Reference | 340 |
| 7.108 | OscIComponentRegistry Class Reference | 341 |
| 7.109 | OscIComponentRegistryData Class Reference | 343 |
| 7.110 | OscIComponentRegistryElement Class Reference | 344 |
| 7.111 | OscIConnectMethod Class Reference | 346 |
| 7.112 | OscIConnectRequest Class Reference | 347 |

| | |
|---|-----|
| 7.113OsciDeconstructDealloc Class Reference | 348 |
| 7.114OsciDNS Class Reference | 349 |
| 7.115OsciDNSI Class Reference | 351 |
| 7.116OsciDNSIBase Class Reference | 353 |
| 7.117OsciDNSMethod Class Reference | 356 |
| 7.118OsciDNSObserver Class Reference | 359 |
| 7.119OsciDNSRequest Class Reference | 360 |
| 7.120OsciDNSRequestAO Class Reference | 361 |
| 7.121OsciDoubleLink Class Reference | 364 |
| 7.122OsciDoubleList< T > Class Template Reference | 365 |
| 7.123OsciDoubleListBase Class Reference | 366 |
| 7.124OsciDoubleRunner< T > Class Template Reference | 368 |
| 7.125OsciError Class Reference | 370 |
| 7.126OsciErrorAllocator Class Reference | 372 |
| 7.127OsciErrorTrap Class Reference | 374 |
| 7.128OsciErrorTrapImp Class Reference | 375 |
| 7.129OsciException< LeaveCode > Class Template Reference | 377 |
| 7.130OsciExclusiveArrayPtr< T > Class Template Reference | 378 |
| 7.131OsciExclusivePtr< T > Class Template Reference | 381 |
| 7.132OsciExclusivePtrA< T, Alloc > Class Template Reference | 384 |
| 7.133OsciExecScheduler Class Reference | 387 |
| 7.134OsciExecSchedulerBase Class Reference | 389 |
| 7.135OsciExecSchedulerCommonBase Class Reference | 390 |
| 7.136OsciFileCache Class Reference | 399 |
| 7.137OsciFileCacheBuffer Class Reference | 401 |
| 7.138OsciFileHandle Class Reference | 403 |
| 7.139OsciFileManager Class Reference | 404 |
| 7.140OsciFileStats Class Reference | 409 |
| 7.141OsciFileStatsItem Class Reference | 410 |
| 7.142OsciGetHostByNameMethod Class Reference | 411 |
| 7.143OsciGetHostByNameRequest Class Reference | 412 |
| 7.144OsciInit Class Reference | 413 |
| 7.145OsciInteger64Transport Struct Reference | 414 |
| 7.146OsciIpMReq Class Reference | 415 |
| 7.147OsciIPSocketI Class Reference | 416 |
| 7.148OsciJump Class Reference | 419 |

| | |
|--|-----|
| 7.149OscListenMethod Class Reference | 420 |
| 7.150OscListenRequest Class Reference | 421 |
| 7.151OscLockBase Class Reference | 422 |
| 7.152OscMem Class Reference | 423 |
| 7.153OscMemAllocator Class Reference | 424 |
| 7.154OscMemAllocDestructDealloc< T > Class Template Reference | 425 |
| 7.155OscMemAudit Class Reference | 427 |
| 7.156OSCLMemAutoPtr< T, _Allocator > Class Template Reference | 433 |
| 7.157OscMemBasicAllocator Class Reference | 437 |
| 7.158OscMemBasicAllocDestructDealloc< T > Class Template Reference | 438 |
| 7.159OscMemGlobalAuditObject Class Reference | 439 |
| 7.160OscMemoryFragment Struct Reference | 440 |
| 7.161OscMemPoolAllocator Class Reference | 441 |
| 7.162OscMemPoolFixedChunkAllocator Class Reference | 442 |
| 7.163OscMemPoolFixedChunkAllocatorObserver Class Reference | 446 |
| 7.164OscMemPoolResizableAllocator Class Reference | 447 |
| 7.165OscMemPoolResizableAllocator::MemPoolBlockInfo Struct Reference | 453 |
| 7.166OscMemPoolResizableAllocator::MemPoolBufferInfo Struct Reference | 454 |
| 7.167OscMemPoolResizableAllocatorMemoryObserver Class Reference | 455 |
| 7.168OscMemPoolResizableAllocatorObserver Class Reference | 456 |
| 7.169OscMemStatsNode Class Reference | 457 |
| 7.170OscMutex Class Reference | 458 |
| 7.171OscNameString< __len > Class Template Reference | 460 |
| 7.172OscNativeFile Class Reference | 461 |
| 7.173OscNativeFileParams Class Reference | 464 |
| 7.174OscNetworkAddress Class Reference | 465 |
| 7.175OscNullLock Class Reference | 466 |
| 7.176OscPriorityLink Class Reference | 467 |
| 7.177OscPriorityList< T > Class Template Reference | 468 |
| 7.178OscPriorityQueue< Qelem, Alloc, Container, Compare > Class Template Reference | 469 |
| 7.179OscPriorityQueueBase Class Reference | 473 |
| 7.180OscProcStatus Class Reference | 474 |
| 7.181OscPtr Class Reference | 476 |
| 7.182OscPtrC Class Reference | 478 |
| 7.183OscRand Class Reference | 480 |
| 7.184OscReadyAlloc Class Reference | 481 |

| | |
|--|-----|
| 7.185OsciReadyCompare Class Reference | 482 |
| 7.186OsciReadyQ Class Reference | 483 |
| 7.187OsciRecvFromMethod Class Reference | 485 |
| 7.188OsciRecvFromRequest Class Reference | 487 |
| 7.189OsciRecvMethod Class Reference | 489 |
| 7.190OsciRecvRequest Class Reference | 490 |
| 7.191OsciRefCountCounter Class Reference | 491 |
| 7.192OsciRefCountCounterDA Class Reference | 493 |
| 7.193OsciRefCountCounterMemFrag Class Reference | 495 |
| 7.194OsciRefCountCounterMTDA< LockType > Class Template Reference | 497 |
| 7.195OsciRefCountCounterMTSA< DeallocType, LockType > Class Template Reference | 499 |
| 7.196OsciRefCountCounterSA< DeallocType > Class Template Reference | 501 |
| 7.197OsciRegistryAccessClient Class Reference | 503 |
| 7.198OsciRegistryAccessClientImpl Class Reference | 505 |
| 7.199OsciRegistryAccessClientTlsImpl Class Reference | 506 |
| 7.200OsciRegistryAccessElement Class Reference | 507 |
| 7.201OsciRegistryClient Class Reference | 508 |
| 7.202OsciRegistryClientImpl Class Reference | 510 |
| 7.203OsciRegistryClientTlsImpl Class Reference | 512 |
| 7.204OsciRegistryServTlsImpl Class Reference | 513 |
| 7.205OsciScheduler Class Reference | 515 |
| 7.206OsciSchedulerObserver Class Reference | 516 |
| 7.207OsciScopedLock< LockClass > Class Template Reference | 517 |
| 7.208OsciSelect Class Reference | 518 |
| 7.209OsciSemaphore Class Reference | 520 |
| 7.210OsciSendMethod Class Reference | 522 |
| 7.211OsciSendRequest Class Reference | 523 |
| 7.212OsciSendToMethod Class Reference | 524 |
| 7.213OsciSendToRequest Class Reference | 525 |
| 7.214OsciSharedPtr< TheClass > Class Template Reference | 526 |
| 7.215OsciShutdownMethod Class Reference | 529 |
| 7.216OsciShutdownRequest Class Reference | 530 |
| 7.217OsciSingleton< T, ID, Registry > Class Template Reference | 531 |
| 7.218OsciSingletonRegistry Class Reference | 533 |
| 7.219OsciSocketI Class Reference | 534 |
| 7.220OsciSocketIBase Class Reference | 539 |

| | |
|--|-----|
| 7.221OsciSocketMethod Class Reference | 544 |
| 7.222OsciSocketObserver Class Reference | 547 |
| 7.223OsciSocketRequest Class Reference | 548 |
| 7.224OsciSocketRequestAO Class Reference | 549 |
| 7.225OsciSocketServ Class Reference | 553 |
| 7.226OsciSocketServI Class Reference | 555 |
| 7.227OsciSocketServIBase Class Reference | 557 |
| 7.228OsciSocketServRequestList Class Reference | 559 |
| 7.229OsciSocketServRequestQElem Class Reference | 561 |
| 7.230OsciSocketTOS Class Reference | 562 |
| 7.231OsciTCPSocket Class Reference | 564 |
| 7.232OsciTCPSocketI Class Reference | 571 |
| 7.233OsciThread Class Reference | 574 |
| 7.234OsciThreadLock Class Reference | 578 |
| 7.235OsciTickCount Class Reference | 579 |
| 7.236OsciTimer< Alloc > Class Template Reference | 581 |
| 7.237OsciTimerCompare Class Reference | 584 |
| 7.238OsciTimerObject Class Reference | 585 |
| 7.239OsciTimerObserver Class Reference | 589 |
| 7.240OsciTimerQ Class Reference | 590 |
| 7.241OsciTLS< T, ID, Registry > Class Template Reference | 591 |
| 7.242OsciTLSEx< T, ID, Registry > Class Template Reference | 593 |
| 7.243OsciTLSRegistry Class Reference | 595 |
| 7.244OsciTLSRegistryEx Class Reference | 596 |
| 7.245OsciTrapItem Class Reference | 597 |
| 7.246OsciTrapStack Class Reference | 598 |
| 7.247OsciTrapStackItem Class Reference | 599 |
| 7.248OsciUDPSocket Class Reference | 600 |
| 7.249OsciUDPSocketI Class Reference | 606 |
| 7.250OsciUuid Struct Reference | 609 |
| 7.251PVActiveBase Class Reference | 611 |
| 7.252PVActiveStats Class Reference | 615 |
| 7.253PVLogger Class Reference | 616 |
| 7.254PVLoggerAppender Class Reference | 622 |
| 7.255PVLoggerFilter Class Reference | 623 |
| 7.256PVLoggerLayout Class Reference | 625 |

| | |
|--|------------|
| 7.257PVLoggerRegistry Class Reference | 627 |
| 7.258PVSchedulerStopper Class Reference | 630 |
| 7.259PVSockBufRecv Class Reference | 631 |
| 7.260PVSockBufSend Class Reference | 632 |
| 7.261PVThreadContext Class Reference | 633 |
| 7.262RecvFromParam Class Reference | 635 |
| 7.263RecvParam Class Reference | 637 |
| 7.264SendParam Class Reference | 638 |
| 7.265SendToParam Class Reference | 639 |
| 7.266ShutdownParam Class Reference | 640 |
| 7.267SocketRequestParam Class Reference | 641 |
| 7.268StrCSumPtrLen Struct Reference | 643 |
| 7.269StrPtrLen Struct Reference | 646 |
| 7.270TimeValue Class Reference | 648 |
| 7.271TLSStorageOps Class Reference | 655 |
| 7.272TReadyQueLink Class Reference | 656 |
| 7.273WStrPtrLen Struct Reference | 657 |
| | |
| 8 oscl File Documentation | 659 |
| 8.1 oscl_aostatus.h File Reference | 659 |
| 8.2 oscl_assert.h File Reference | 660 |
| 8.3 oscl_base.h File Reference | 661 |
| 8.4 oscl_base_alloc.h File Reference | 662 |
| 8.5 oscl_base_macros.h File Reference | 663 |
| 8.6 oscl_bin_stream.h File Reference | 664 |
| 8.7 oscl_byte_order.h File Reference | 665 |
| 8.8 oscl_defalloc.h File Reference | 666 |
| 8.9 oscl_dll.h File Reference | 667 |
| 8.10 oscl_dns.h File Reference | 668 |
| 8.11 oscl_dns_gethostbyname.h File Reference | 669 |
| 8.12 oscl_dns_imp.h File Reference | 670 |
| 8.13 oscl_dns_imp_base.h File Reference | 671 |
| 8.14 oscl_dns_imp_pv.h File Reference | 672 |
| 8.15 oscl_dns_method.h File Reference | 673 |
| 8.16 oscl_dns_param.h File Reference | 674 |
| 8.17 oscl_dns_request.h File Reference | 675 |
| 8.18 oscl_dns_tuneables.h File Reference | 676 |

| | | |
|------|---|-----|
| 8.19 | oscl_double_list.h File Reference | 677 |
| 8.20 | oscl_errno.h File Reference | 678 |
| 8.21 | oscl_error.h File Reference | 679 |
| 8.22 | oscl_error_allocator.h File Reference | 680 |
| 8.23 | oscl_error_codes.h File Reference | 681 |
| 8.24 | oscl_error_imp.h File Reference | 682 |
| 8.25 | oscl_error_imp_cppexceptions.h File Reference | 683 |
| 8.26 | oscl_error_imp_fatalerror.h File Reference | 684 |
| 8.27 | oscl_error_imp_jumps.h File Reference | 685 |
| 8.28 | oscl_error_trapcleanup.h File Reference | 687 |
| 8.29 | oscl_exception.h File Reference | 688 |
| 8.30 | oscl_exclusive_ptr.h File Reference | 689 |
| 8.31 | oscl_file_async_read.h File Reference | 690 |
| 8.32 | oscl_file_cache.h File Reference | 691 |
| 8.33 | oscl_file_dir_utils.h File Reference | 692 |
| 8.34 | oscl_file_find.h File Reference | 694 |
| 8.35 | oscl_file_handle.h File Reference | 695 |
| 8.36 | oscl_file_io.h File Reference | 696 |
| 8.37 | oscl_file_manager.h File Reference | 697 |
| 8.38 | oscl_file_native.h File Reference | 698 |
| 8.39 | oscl_file_server.h File Reference | 699 |
| 8.40 | oscl_file_stats.h File Reference | 700 |
| 8.41 | oscl_file_types.h File Reference | 701 |
| 8.42 | oscl_heapbase.h File Reference | 702 |
| 8.43 | oscl_init.h File Reference | 703 |
| 8.44 | oscl_int64_utils.h File Reference | 704 |
| 8.45 | oscl_ip_socket.h File Reference | 705 |
| 8.46 | oscl_linked_list.h File Reference | 706 |
| 8.47 | oscl_lock_base.h File Reference | 707 |
| 8.48 | oscl_map.h File Reference | 708 |
| 8.49 | oscl_math.h File Reference | 709 |
| 8.50 | oscl_media_data.h File Reference | 710 |
| 8.51 | oscl_media_status.h File Reference | 711 |
| 8.52 | oscl_mem.h File Reference | 712 |
| 8.53 | oscl_mem_align.h File Reference | 715 |
| 8.54 | oscl_mem_audit.h File Reference | 716 |

| | | |
|------|---|-----|
| 8.55 | oscl_mem_audit_internals.h File Reference | 718 |
| 8.56 | oscl_mem_auto_ptr.h File Reference | 719 |
| 8.57 | oscl_mem_basic_functions.h File Reference | 720 |
| 8.58 | oscl_mem_inst.h File Reference | 721 |
| 8.59 | oscl_mem_mempool.h File Reference | 722 |
| 8.60 | oscl_mempool_allocator.h File Reference | 723 |
| 8.61 | oscl_mutex.h File Reference | 724 |
| 8.62 | oscl_namestring.h File Reference | 725 |
| 8.63 | oscl_opaque_type.h File Reference | 726 |
| 8.64 | oscl_priqueue.h File Reference | 727 |
| 8.65 | oscl_procstatus.h File Reference | 728 |
| 8.66 | oscl_queue.h File Reference | 729 |
| 8.67 | oscl_rand.h File Reference | 730 |
| 8.68 | oscl_refcounter.h File Reference | 731 |
| 8.69 | oscl_refcounter_memfrag.h File Reference | 732 |
| 8.70 | oscl_registry_access_client.h File Reference | 733 |
| 8.71 | oscl_registry_client.h File Reference | 734 |
| 8.72 | oscl_registry_client_impl.h File Reference | 735 |
| 8.73 | oscl_registry_serv_impl.h File Reference | 736 |
| 8.74 | oscl_registry_serv_impl_global.h File Reference | 737 |
| 8.75 | oscl_registry_serv_impl_tls.h File Reference | 738 |
| 8.76 | oscl_registry_types.h File Reference | 739 |
| 8.77 | oscl_scheduler.h File Reference | 740 |
| 8.78 | oscl_scheduler_ao.h File Reference | 741 |
| 8.79 | oscl_scheduler_aobase.h File Reference | 742 |
| 8.80 | oscl_scheduler_readyq.h File Reference | 743 |
| 8.81 | oscl_scheduler_threadcontext.h File Reference | 744 |
| 8.82 | oscl_scheduler_tuneables.h File Reference | 745 |
| 8.83 | oscl_scheduler_types.h File Reference | 746 |
| 8.84 | oscl_semaphore.h File Reference | 747 |
| 8.85 | oscl_shared_ptr.h File Reference | 748 |
| 8.86 | oscl_singleton.h File Reference | 749 |
| 8.87 | oscl_snprintf.h File Reference | 751 |
| 8.88 | oscl_socket.h File Reference | 752 |
| 8.89 | oscl_socket_accept.h File Reference | 753 |
| 8.90 | oscl_socket_bind.h File Reference | 754 |

| | | |
|-------|---|-----|
| 8.91 | oscl_socket_connect.h File Reference | 755 |
| 8.92 | oscl_socket_imp.h File Reference | 756 |
| 8.93 | oscl_socket_imp_base.h File Reference | 757 |
| 8.94 | oscl_socket_imp_pv.h File Reference | 758 |
| 8.95 | oscl_socket_listen.h File Reference | 759 |
| 8.96 | oscl_socket_method.h File Reference | 760 |
| 8.97 | oscl_socket_recv.h File Reference | 761 |
| 8.98 | oscl_socket_recv_from.h File Reference | 762 |
| 8.99 | oscl_socket_request.h File Reference | 763 |
| 8.100 | oscl_socket_send.h File Reference | 764 |
| 8.101 | oscl_socket_send_to.h File Reference | 765 |
| 8.102 | oscl_socket_serv_imp.h File Reference | 766 |
| 8.103 | oscl_socket_serv_imp_base.h File Reference | 767 |
| 8.104 | oscl_socket_serv_imp_pv.h File Reference | 768 |
| 8.105 | oscl_socket_serv_imp_reqlist.h File Reference | 769 |
| 8.106 | oscl_socket_shutdown.h File Reference | 770 |
| 8.107 | oscl_socket_stats.h File Reference | 771 |
| 8.108 | oscl_socket_tuneables.h File Reference | 773 |
| 8.109 | oscl_socket_types.h File Reference | 775 |
| 8.110 | oscl_stdstring.h File Reference | 777 |
| 8.111 | oscl_str_ptr_len.h File Reference | 779 |
| 8.112 | oscl_string.h File Reference | 780 |
| 8.113 | oscl_string_containers.h File Reference | 781 |
| 8.114 | oscl_string_rep.h File Reference | 782 |
| 8.115 | oscl_string_uri.h File Reference | 783 |
| 8.116 | oscl_string_utf8.h File Reference | 784 |
| 8.117 | oscl_string_utils.h File Reference | 785 |
| 8.118 | oscl_string_xml.h File Reference | 786 |
| 8.119 | oscl_tagtree.h File Reference | 787 |
| 8.120 | oscl_tcp_socket.h File Reference | 788 |
| 8.121 | oscl_thread.h File Reference | 789 |
| 8.122 | oscl_tickcount.h File Reference | 791 |
| 8.123 | oscl_time.h File Reference | 792 |
| 8.124 | oscl_timer.h File Reference | 794 |
| 8.125 | oscl_tls.h File Reference | 795 |
| 8.126 | oscl_tree.h File Reference | 796 |

| | |
|---|-----|
| 8.127oscl_types.h File Reference | 797 |
| 8.128oscl_udp_socket.h File Reference | 798 |
| 8.129oscl_utf8conv.h File Reference | 799 |
| 8.130oscl_uuid.h File Reference | 800 |
| 8.131oscl_vector.h File Reference | 802 |
| 8.132osclconfig.h File Reference | 803 |
| 8.133osclconfig_ansi_memory.h File Reference | 805 |
| 8.134osclconfig_check.h File Reference | 806 |
| 8.135osclconfig_compiler_warnings.h File Reference | 807 |
| 8.136osclconfig_error.h File Reference | 808 |
| 8.137osclconfig_error_check.h File Reference | 809 |
| 8.138osclconfig_global_new_delete.h File Reference | 810 |
| 8.139osclconfig_global_placement_new.h File Reference | 811 |
| 8.140osclconfig_io.h File Reference | 812 |
| 8.141osclconfig_io_check.h File Reference | 823 |
| 8.142osclconfig_ix86.h File Reference | 824 |
| 8.143osclconfig_lib.h File Reference | 825 |
| 8.144osclconfig_lib_check.h File Reference | 826 |
| 8.145osclconfig_limits_typedefs.h File Reference | 827 |
| 8.146osclconfig_memory.h File Reference | 828 |
| 8.147osclconfig_memory_check.h File Reference | 829 |
| 8.148osclconfig_no_os.h File Reference | 830 |
| 8.149osclconfig_proc.h File Reference | 831 |
| 8.150osclconfig_proc_check.h File Reference | 832 |
| 8.151osclconfig_proc_unix_android.h File Reference | 834 |
| 8.152osclconfig_proc_unix_common.h File Reference | 836 |
| 8.153osclconfig_time.h File Reference | 838 |
| 8.154osclconfig_time_check.h File Reference | 839 |
| 8.155osclconfig_unix_android.h File Reference | 840 |
| 8.156osclconfig_unix_common.h File Reference | 844 |
| 8.157osclconfig_util.h File Reference | 848 |
| 8.158osclconfig_util_check.h File Reference | 849 |
| 8.159pvlogger.h File Reference | 850 |
| 8.160pvlogger_accessories.h File Reference | 858 |
| 8.161pvlogger_c.h File Reference | 859 |
| 8.162pvlogger_registry.h File Reference | 861 |

| | | |
|----------|--------------------------------|------------|
| 9 | oscl Page Documentation | 862 |
| 9.1 | Todo List | 862 |

Chapter 1

oscl Module Index

1.1 oscl Modules

Here is a list of all modules:

| | |
|-----------------------|-----|
| OSCL config | 21 |
| OSCL Base | 25 |
| OSCL Memory | 46 |
| OSCL Util | 62 |
| OSCL Error | 84 |
| OSCL IO | 94 |
| OSCL Proc | 102 |
| OSCL Init | 106 |

Chapter 2

oscl Hierarchical Index

2.1 oscl Class Hierarchy

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

| | |
|--------------------------------|-----|
| _OscHeapBase | 109 |
| HeapBase | 136 |
| Osc_File | 178 |
| OSCL_String | 258 |
| OSCL_FastString | 174 |
| OSCL_HeapString< Alloc > | 195 |
| OSCL_HeapStringA | 197 |
| OSCL_StackString< MaxBufSize > | 255 |
| OscActiveObject | 308 |
| OscAsyncFile | 314 |
| OscDNSRequestAO | 361 |
| OscGetHostByNameRequest | 412 |
| OscSocketRequestAO | 549 |
| OscAcceptRequest | 307 |
| OscBindRequest | 321 |
| OscConnectRequest | 347 |
| OscListenRequest | 421 |
| OscRecvFromRequest | 487 |
| OscRecvRequest | 490 |
| OscSendRequest | 523 |
| OscSendToRequest | 525 |
| OscShutdownRequest | 530 |
| PVSchedulerStopper | 630 |
| OscAsyncFileBuffer | 317 |
| OscBuf | 338 |
| OscDNS | 349 |
| OscFileCache | 399 |
| OscNativeFile | 461 |
| OscPtr | 476 |
| OscPtrC | 478 |
| OscRegistryClient | 508 |
| OscSocketServ | 553 |
| OscTCPSocket | 564 |

| | |
|---|-----|
| OscTimerObject | 585 |
| CallbackTimer< Alloc > | 123 |
| OscDNSMethod | 356 |
| OscGetHostByNameMethod | 411 |
| OscSocketMethod | 544 |
| OscAcceptMethod | 306 |
| OscBindMethod | 320 |
| OscConnectMethod | 346 |
| OscListenMethod | 420 |
| OscRecvFromMethod | 485 |
| OscRecvMethod | 489 |
| OscSendMethod | 522 |
| OscSendToMethod | 524 |
| OscShutdownMethod | 529 |
| OscSocketServI | 555 |
| OscUDPSocket | 600 |
| OscExecSchedulerBase | 389 |
| OscExecScheduler | 387 |
| allocator | 112 |
| BufferMgr | 117 |
| BufferState | 118 |
| BufFragGroup< ChainClass, max_frags > | 119 |
| MediaData< ChainClass, max_frags, local_bufsize > | 141 |
| BufFragStatusClass | 122 |
| MediaStatusClass | 144 |
| CallbackTimerObserver | 125 |
| OscTimer< Alloc > | 581 |
| CFastRep | 126 |
| CHeapRep | 128 |
| CStackRep | 131 |
| DNSRequestParam | 132 |
| GetHostByNameParam | 134 |
| internalLeave | 138 |
| LinkedListElement< LLClass > | 139 |
| MemAllocator< T > | 145 |
| MM_AllocBlockFence | 146 |
| MM_AllocBlockHdr | 147 |
| MM_AllocInfo | 148 |
| MM_AllocNode | 150 |
| MM_AllocQueryInfo | 151 |
| MM_Audit_Imp | 152 |
| MM_AuditOverheadStats | 160 |
| MM_FailInsertParam | 161 |
| MM_Stats_CB | 162 |
| MM_Stats_t | 163 |
| NTPTime | 165 |
| Osc_Alloc | 169 |
| Osc_DefAlloc | 171 |
| _OscBasicAllocator | 107 |
| OscAllocDestructDealloc | 312 |
| OscMemAllocDestructDealloc< T > | 425 |
| OscMemBasicAllocDestructDealloc< T > | 438 |

| | |
|--|-----|
| OscMemAllocator | 424 |
| OscMemBasicAllocator | 437 |
| OscMemPoolFixedChunkAllocator | 442 |
| OscMemPoolResizableAllocator | 447 |
| OscReadyAlloc | 481 |
| Osc_Dealloc | 170 |
| Osc_DefAlloc | 171 |
| Osc_File::OscCacheObserver | 186 |
| Osc_File::OscFixedCacheParam | 187 |
| Osc_FileFind | 188 |
| Osc_FileServer | 192 |
| oscl_fsstat | 194 |
| Osc_Int64_Utils | 202 |
| Osc_Less< T > | 204 |
| Osc_Linked_List_Base | 210 |
| Osc_Linked_List< LLClass, Alloc > | 205 |
| Osc_Map< Key, T, Alloc, Compare > | 215 |
| Osc_Map< Key, T, Alloc, Compare >::value_compare | 222 |
| Osc_MTLinked_List< LLClass, Alloc, TheLock > | 224 |
| Osc_Opaque_Type_Alloc | 228 |
| Osc_Queue< T, Alloc > | 234 |
| Osc_Vector< T, Alloc > | 283 |
| Osc_Vector< TOscReady, OscReadyAlloc > | 283 |
| Osc_Opaque_Type_Alloc_LL | 229 |
| Osc_Linked_List< LLClass, Alloc > | 205 |
| Osc_Opaque_Type_Compare | 231 |
| OscPriorityQueue< Qelem, Alloc, Container, Compare > | 469 |
| OscPriorityQueue< TOscReady, OscReadyAlloc, Osc_Vector< TOscReady, OscReady-Alloc >, OscReadyCompare > | 469 |
| OscReadyQ | 483 |
| OscPriorityQueue< TOscReady, OscReadyAlloc, Osc_Vector< TOscReady, OscReady-Alloc >, OscTimerCompare > | 469 |
| OscTimerQ | 590 |
| Osc_Pair< T1, T2 > | 233 |
| Osc_Queue_Base | 237 |
| Osc_Queue< T, Alloc > | 234 |
| Osc_Rb_Tree_Base | 244 |
| Osc_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > | 240 |
| Osc_Rb_Tree_Const_Iterator< Value > | 245 |
| Osc_Rb_Tree_Iterator< Value > | 248 |
| Osc_Rb_Tree_Node_Base | 252 |
| Osc_Rb_Tree_Node< Value > | 251 |
| Osc_Select1st< V, U > | 254 |
| oscl_stat_buf | 257 |
| Osc_Tag_Base | 265 |
| Osc_Tag< Alloc > | 263 |
| Osc_TagTree< T, Alloc > | 267 |
| Osc_TagTree< T, Alloc >::const_iterator | 271 |
| Osc_TagTree< T, Alloc >::iterator | 274 |
| Osc_TagTree< T, Alloc >::Node | 277 |
| Osc_TAlloc< T, Alloc >::rebind< U, V > | 282 |

| | |
|---|-----|
| OscL_Vector_Base | 288 |
| OscL_Vector< T, Alloc > | 283 |
| OscL_Vector< TOscLReady, OscLReadyAlloc > | 283 |
| OSCL_wString | 302 |
| OSCL_wFastString | 292 |
| OSCL_wHeapString< Alloc > | 295 |
| OSCL_wHeapStringA | 297 |
| OSCL_wStackString< MaxBufSize > | 300 |
| OscLAOStatus | 313 |
| OscLAuditCB | 319 |
| OscLBinStream | 334 |
| OscLBinIStream | 322 |
| OscLBinIStreamBigEndian | 324 |
| OscLBinIStreamLittleEndian | 327 |
| OscLBinOStream | 329 |
| OscLBinOStreamBigEndian | 330 |
| OscLBinOStreamLittleEndian | 332 |
| OscLCompareLess< T > | 340 |
| OscLComponentRegistry | 341 |
| OscLComponentRegistryData | 343 |
| OscLComponentRegistryElement | 344 |
| OscLDestructDealloc | 348 |
| OscL_TAlloc< T, Alloc > | 279 |
| OscLAllocDestructDealloc | 312 |
| OscLDNSIBase | 353 |
| OscLDNSI | 351 |
| OscLDNSObserver | 359 |
| OscLDNSRequest | 360 |
| OscLDoubleLink | 364 |
| OscLPriorityLink | 467 |
| OscLDoubleListBase | 366 |
| OscLDoubleList< T > | 365 |
| OscLPriorityList< T > | 468 |
| OscLDoubleRunner< T > | 368 |
| OscLError | 370 |
| OscLErrorAllocator | 372 |
| OscLErrorTrap | 374 |
| OscLErrorTrapImp | 375 |
| OscLException< LeaveCode > | 377 |
| OscLExclusiveArrayPtr< T > | 378 |
| OscLExclusivePtr< T > | 381 |
| OscLExclusivePtrA< T, Alloc > | 384 |
| OscLExecSchedulerCommonBase | 390 |
| OscLExecScheduler | 387 |
| OscLFileCacheBuffer | 401 |
| OscLFileHandle | 403 |
| OscLFileManager | 404 |
| OscLFileStats | 409 |
| OscLFileStatsItem | 410 |
| OscLInit | 413 |
| OscLInteger64Transport | 414 |

| | |
|--|-----|
| OscIpMReq | 415 |
| OscIPSocketI | 416 |
| OscTCPSocketI | 571 |
| OscUDPSocketI | 606 |
| OscJump | 419 |
| OscLockBase | 422 |
| OscMutex | 458 |
| OscNullLock | 466 |
| OscThreadLock | 578 |
| OscMem | 423 |
| OscMemAudit | 427 |
| OscMemAutoPtr< T, _Allocator > | 433 |
| OscMemGlobalAuditObject | 439 |
| OscMemoryFragment | 440 |
| BufferFragment | 116 |
| OscMemPoolAllocator | 441 |
| OscMemPoolFixedChunkAllocatorObserver | 446 |
| OscMemPoolResizableAllocator::MemPoolBlockInfo | 453 |
| OscMemPoolResizableAllocator::MemPoolBufferInfo | 454 |
| OscMemPoolResizableAllocatorMemoryObserver | 455 |
| OscMemPoolResizableAllocatorObserver | 456 |
| OscMemStatsNode | 457 |
| OscNameString< __len > | 460 |
| OscNativeFileParams | 464 |
| OscNetworkAddress | 465 |
| OscPriorityQueueBase | 473 |
| OscPriorityQueue< Qelem, Alloc, Container, Compare > | 469 |
| OscPriorityQueue< TOscReady, OscReadyAlloc, Osc_Vector< TOscReady, OscReady-Alloc >, OscReadyCompare > | 469 |
| OscPriorityQueue< TOscReady, OscReadyAlloc, Osc_Vector< TOscReady, OscReady-Alloc >, OscTimerCompare > | 469 |
| OscProcStatus | 474 |
| OscRand | 480 |
| OscReadyCompare | 482 |
| OscRefCount | 491 |
| Osc_DefAllocWithRefCount< DefAlloc > | 172 |
| OscRefCountDA | 493 |
| OscRefCountMTDA< LockType > | 497 |
| OscRefCountMTSA< DeallocType, LockType > | 499 |
| OscRefCountSA< DeallocType > | 501 |
| OscRefCountMemFrag | 495 |
| OscRegistryAccessClient | 503 |
| OscRegistryAccessElement | 507 |
| OscRegistryClientImpl | 510 |
| OscRegistryAccessClientImpl | 505 |
| OscRegistryServTlsImpl | 513 |
| OscRegistryAccessClientTlsImpl | 506 |
| OscRegistryClientTlsImpl | 512 |
| OscScheduler | 515 |
| OscSchedulerObserver | 516 |
| OscScopedLock< LockClass > | 517 |
| OscSelect | 518 |

| | |
|---------------------------------|-----|
| OscSemaphore | 520 |
| OscSharedPtr< TheClass > | 526 |
| OscSingleton< T, ID, Registry > | 531 |
| OscSingletonRegistry | 533 |
| OscSocketIBase | 539 |
| OscSocketI | 534 |
| OscSocketObserver | 547 |
| OscSocketRequest | 548 |
| OscSocketServIBase | 557 |
| OscSocketServI | 555 |
| OscSocketServRequestList | 559 |
| OscSocketServRequestQElem | 561 |
| OscSocketTOS | 562 |
| OscThread | 574 |
| OscTickCount | 579 |
| OscTimerCompare | 584 |
| OscTimerObserver | 589 |
| OscTLS< T, ID, Registry > | 591 |
| OscTLSEx< T, ID, Registry > | 593 |
| OscTLSRegistry | 595 |
| OscTLSRegistryEx | 596 |
| OscTrapItem | 597 |
| OscTrapStack | 598 |
| OscTrapStackItem | 599 |
| OscUuid | 609 |
| PVActiveBase | 611 |
| OscActiveObject | 308 |
| OscTimerObject | 585 |
| PVActiveStats | 615 |
| PVLogger | 616 |
| PVLoggerAppender | 622 |
| PVLoggerFilter | 623 |
| AllPassFilter | 113 |
| PVLoggerLayout | 625 |
| PVLoggerRegistry | 627 |
| PVSockBufRecv | 631 |
| PVSockBufSend | 632 |
| PVThreadContext | 633 |
| SocketRequestParam | 641 |
| AcceptParam | 111 |
| BindParam | 115 |
| ConnectParam | 130 |
| ListenParam | 140 |
| RecvFromParam | 635 |
| RecvParam | 637 |
| SendParam | 638 |
| SendToParam | 639 |
| ShutdownParam | 640 |
| StrPtrLen | 646 |
| StrCSumPtrLen | 643 |
| TimeValue | 648 |
| TLSStorageOps | 655 |

| | |
|-------------------------|-----|
| TReadyQueLink | 656 |
| WStrPtrLen | 657 |

Chapter 3

oscl Data Structure Index

3.1 oscl Data Structures

Here are the data structures with brief descriptions:

| | |
|---|-----|
| _OscBasicAllocator | 107 |
| _OscHeapBase | 109 |
| AcceptParam | 111 |
| allocator | 112 |
| AllPassFilter | 113 |
| BindParam | 115 |
| BufferFragment | 116 |
| BufferMgr | 117 |
| BufferState | 118 |
| BufFragGroup< ChainClass, max_frags > | 119 |
| BufFragStatusClass | 122 |
| CallbackTimer< Alloc > | 123 |
| CallbackTimerObserver | 125 |
| CFastRep | 126 |
| CHeapRep | 128 |
| ConnectParam | 130 |
| CStackRep | 131 |
| DNSRequestParam | 132 |
| GetHostByNameParam | 134 |
| HeapBase | 136 |
| internalLeave | 138 |
| LinkedListElement< LLClass > | 139 |
| ListenParam | 140 |
| MediaData< ChainClass, max_frags, local_bufsize > | 141 |
| MediaStatusClass | 144 |
| MemAllocator< T > | 145 |
| MM_AllocBlockFence | 146 |
| MM_AllocBlockHdr | 147 |
| MM_AllocInfo | 148 |
| MM_AllocNode | 150 |
| MM_AllocQueryInfo | 151 |
| MM_Audit_Imp | 152 |
| MM_AuditOverheadStats | 160 |

| | |
|---|-----|
| MM_FailInsertParam | 161 |
| MM_Stats_CB | 162 |
| MM_Stats_t | 163 |
| NTPTime (Time value as the number of seconds since 0h (UTC) Jan. 1, 1900) | 165 |
| OscL_Alloc | 169 |
| OscL_Dealloc | 170 |
| OscL_DefAlloc | 171 |
| OscL_DefAllocWithRefCount< DefAlloc > | 172 |
| OSCL_FastString | 174 |
| OscL_File | 178 |
| OscL_File::OscLCacheObserver | 186 |
| OscL_File::OscLFixedCacheParam | 187 |
| OscL_FileFind | 188 |
| OscL_FileServer | 192 |
| oscl_fsstat | 194 |
| OSCL_HeapString< Alloc > | 195 |
| OSCL_HeapStringA | 197 |
| OscL_Int64_Utils (Wrapper for commonly used int64/uint64 operations) | 202 |
| OscL_Less< T > | 204 |
| OscL_Linked_List< LLClass, Alloc > | 205 |
| OscL_Linked_List_Base | 210 |
| OscL_Map< Key, T, Alloc, Compare > | 215 |
| OscL_Map< Key, T, Alloc, Compare >::value_compare | 222 |
| OscL_MTLinked_List< LLClass, Alloc, TheLock > | 224 |
| OscL_Opaque_Type_Alloc | 228 |
| OscL_Opaque_Type_Alloc_LL | 229 |
| OscL_Opaque_Type_Compare | 231 |
| OscL_Pair< T1, T2 > | 233 |
| OscL_Queue< T, Alloc > | 234 |
| OscL_Queue_Base | 237 |
| OscL_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > | 240 |
| OscL_Rb_Tree_Base | 244 |
| OscL_Rb_Tree_Const_Iterator< Value > | 245 |
| OscL_Rb_Tree_Iterator< Value > | 248 |
| OscL_Rb_Tree_Node< Value > | 251 |
| OscL_Rb_Tree_Node_Base | 252 |
| OscL_Select1st< V, U > | 254 |
| OSCL_StackString< MaxBufSize > | 255 |
| oscl_stat_buf | 257 |
| OSCL_String | 258 |
| OscL_Tag< Alloc > | 263 |
| OscL_Tag_Base | 265 |
| OscL_TagTree< T, Alloc > | 267 |
| OscL_TagTree< T, Alloc >::const_iterator | 271 |
| OscL_TagTree< T, Alloc >::iterator | 274 |
| OscL_TagTree< T, Alloc >::Node | 277 |
| OscL_TAlloc< T, Alloc > | 279 |
| OscL_TAlloc< T, Alloc >::rebind< U, V > | 282 |
| OscL_Vector< T, Alloc > | 283 |
| OscL_Vector_Base | 288 |
| OSCL_wFastString | 292 |
| OSCL_wHeapString< Alloc > | 295 |
| OSCL_wHeapStringA | 297 |
| OSCL_wStackString< MaxBufSize > | 300 |

| | |
|---|-----|
| OSCL_wString | 302 |
| OscIAcceptMethod | 306 |
| OscIAcceptRequest | 307 |
| OscIActiveObject | 308 |
| OscIAllocDestructDealloc | 312 |
| OscIAOStatus | 313 |
| OscIAsyncFile | 314 |
| OscIAsyncFileBuffer | 317 |
| OscIAuditCB | 319 |
| OscIBindMethod | 320 |
| OscIBindRequest | 321 |
| OscIBinIStream | 322 |
| OscIBinIStreamBigEndian | 324 |
| OscIBinIStreamLittleEndian | 327 |
| OscIBinOStream (Class OscIBinOStream implements the basic stream functions for an output stream) | 329 |
| OscIBinOStreamBigEndian (Class OscIBinOStreamBigEndian implements a binary output stream using big endian byte ordering) | 330 |
| OscIBinOStreamLittleEndian (Class OscIBinOStreamLittleEndian implements a binary output stream using little endian byte ordering) | 332 |
| OscIBinStream | 334 |
| OscIBuf | 338 |
| OscICompareLess< T > | 340 |
| OscIComponentRegistry | 341 |
| OscIComponentRegistryData | 343 |
| OscIComponentRegistryElement | 344 |
| OscIConnectMethod | 346 |
| OscIConnectRequest | 347 |
| OscIDestructDealloc | 348 |
| OscIDNS | 349 |
| OscIDNSI | 351 |
| OscIDNSIBase | 353 |
| OscIDNSMethod | 356 |
| OscIDNSObserver | 359 |
| OscIDNSRequest | 360 |
| OscIDNSRequestAO | 361 |
| OscIDoubleLink | 364 |
| OscIDoubleList< T > | 365 |
| OscIDoubleListBase | 366 |
| OscIDoubleRunner< T > | 368 |
| OscIError | 370 |
| OscIErrorAllocator (This class provides static methods to invoke the user defined memory allocation routines) | 372 |
| OscIErrorTrap | 374 |
| OscIErrorTrapImp | 375 |
| OscIException< LeaveCode > (OscI_exception.h contains all the exception handling macros and classes This template class provides the base exception class that all exceptions derive from) | 377 |
| OscIExclusiveArrayPtr< T > (Template class that defines an array pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the OscIExclusiveArrayPtr expires, its destructor uses delete to free the memory) | 378 |
| OscIExclusivePtr< T > (Template class that defines a pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the OscIExclusivePtr expires, its destructor uses delete to free the memory) | 381 |

| | |
|--|-----|
| <code>OscExclusivePtrA< T, Alloc ></code> (Template class that defines any pointer like object intended to be assigned an address obtained (directly or or indirectly) through Alloc. When the <code>OscExclusivePtrA</code> expires, Alloc is used to free the memory) | 384 |
| <code>OscExecScheduler</code> | 387 |
| <code>OscExecSchedulerBase</code> | 389 |
| <code>OscExecSchedulerCommonBase</code> | 390 |
| <code>OscFileCache</code> | 399 |
| <code>OscFileCacheBuffer</code> | 401 |
| <code>OscFileHandle</code> | 403 |
| <code>OscFileManager</code> | 404 |
| <code>OscFileStats</code> | 409 |
| <code>OscFileStatsItem</code> | 410 |
| <code>OscGetHostByNameMethod</code> | 411 |
| <code>OscGetHostByNameRequest</code> | 412 |
| <code>OscInit</code> | 413 |
| <code>OscInteger64Transport</code> | 414 |
| <code>OscIpMReq</code> | 415 |
| <code>OscIPSocketI</code> | 416 |
| <code>OscJump</code> | 419 |
| <code>OscListenMethod</code> | 420 |
| <code>OscListenRequest</code> | 421 |
| <code>OscLockBase</code> | 422 |
| <code>OscMem</code> | 423 |
| <code>OscMemAllocator</code> | 424 |
| <code>OscMemAllocDestructDealloc< T ></code> | 425 |
| <code>OscMemAudit</code> | 427 |
| <code>OSCLMemAutoPtr< T, _Allocator ></code> (The <code>oscl_auto_ptr</code> class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or or indirectly) by <code>new</code> . When the <code>oscl_auto_ptr</code> expires, its destructor uses <code>delete</code> to free the memory) | 433 |
| <code>OscMemBasicAllocator</code> | 437 |
| <code>OscMemBasicAllocDestructDealloc< T ></code> | 438 |
| <code>OscMemGlobalAuditObject</code> | 439 |
| <code>OscMemoryFragment</code> | 440 |
| <code>OscMemPoolAllocator</code> | 441 |
| <code>OscMemPoolFixedChunkAllocator</code> | 442 |
| <code>OscMemPoolFixedChunkAllocatorObserver</code> | 446 |
| <code>OscMemPoolResizableAllocator</code> | 447 |
| <code>OscMemPoolResizableAllocator::MemPoolBlockInfo</code> | 453 |
| <code>OscMemPoolResizableAllocator::MemPoolBufferInfo</code> | 454 |
| <code>OscMemPoolResizableAllocatorMemoryObserver</code> | 455 |
| <code>OscMemPoolResizableAllocatorObserver</code> | 456 |
| <code>OscMemStatsNode</code> | 457 |
| <code>OscMutex</code> | 458 |
| <code>OscNameString< __len ></code> | 460 |
| <code>OscNativeFile</code> | 461 |
| <code>OscNativeFileParams</code> | 464 |
| <code>OscNetworkAddress</code> | 465 |
| <code>OscNullLock</code> | 466 |
| <code>OscPriorityLink</code> | 467 |
| <code>OscPriorityList< T ></code> | 468 |
| <code>OscPriorityQueue< Qelem, Alloc, Container, Compare ></code> | 469 |
| <code>OscPriorityQueueBase</code> | 473 |
| <code>OscProcStatus</code> | 474 |

| | |
|--|-----|
| OscIPtr | 476 |
| OscIPtrC | 478 |
| OscIRand | 480 |
| OscIReadyAlloc | 481 |
| OscIReadyCompare | 482 |
| OscIReadyQ | 483 |
| OscIRecvFromMethod | 485 |
| OscIRecvFromRequest | 487 |
| OscIRecvMethod | 489 |
| OscIRecvRequest | 490 |
| OscIRefCounter | 491 |
| OscIRefCounterDA | 493 |
| OscIRefCounterMemFrag | 495 |
| OscIRefCounterMTDA< LockType > | 497 |
| OscIRefCounterMTSA< DeallocType, LockType > | 499 |
| OscIRefCounterSA< DeallocType > | 501 |
| OscIRegistryAccessClient | 503 |
| OscIRegistryAccessClientImpl | 505 |
| OscIRegistryAccessClientTlsImpl | 506 |
| OscIRegistryAccessElement | 507 |
| OscIRegistryClient | 508 |
| OscIRegistryClientImpl | 510 |
| OscIRegistryClientTlsImpl | 512 |
| OscIRegistryServTlsImpl | 513 |
| OscIScheduler | 515 |
| OscISchedulerObserver | 516 |
| OscIScopedLock< LockClass > (Template class that handles unlocking an abstract class on destruction. This is very useful for ensuring that the lock is released when the OscIScopedLock goes out of scope) | 517 |
| OscISelect | 518 |
| OscISemaphore | 520 |
| OscISendMethod | 522 |
| OscISendRequest | 523 |
| OscISendToMethod | 524 |
| OscISendToRequest | 525 |
| OscISharedPtr< TheClass > (A parameterized smart pointer class) | 526 |
| OscIShutdownMethod | 529 |
| OscIShutdownRequest | 530 |
| OscISingleton< T, ID, Registry > | 531 |
| OscISingletonRegistry | 533 |
| OscISocketI | 534 |
| OscISocketIBase | 539 |
| OscISocketMethod | 544 |
| OscISocketObserver | 547 |
| OscISocketRequest | 548 |
| OscISocketRequestAO | 549 |
| OscISocketServ | 553 |
| OscISocketServI | 555 |
| OscISocketServIBase | 557 |
| OscISocketServRequestList | 559 |
| OscISocketServRequestQElem | 561 |
| OscISocketTOS | 562 |
| OscITCPSocket | 564 |
| OscITCPSocketI | 571 |

| | |
|---|-----|
| OscThread | 574 |
| OscThreadLock | 578 |
| OscTickCount | 579 |
| OscTimer< Alloc > | 581 |
| OscTimerCompare | 584 |
| OscTimerObject | 585 |
| OscTimerObserver | 589 |
| OscTimerQ | 590 |
| OscTLS< T, ID, Registry > | 591 |
| OscTLSEx< T, ID, Registry > | 593 |
| OscTLSRegistry | 595 |
| OscTLSRegistryEx | 596 |
| OscTrapItem | 597 |
| OscTrapStack | 598 |
| OscTrapStackItem | 599 |
| OscUDPSocket | 600 |
| OscUDPSocketI | 606 |
| OscUuid | 609 |
| PVActiveBase | 611 |
| PVActiveStats | 615 |
| PVLogger | 616 |
| PVLoggerAppender | 622 |
| PVLoggerFilter | 623 |
| PVLoggerLayout | 625 |
| PVLoggerRegistry | 627 |
| PVSchedulerStopper | 630 |
| PVSockBufRecv | 631 |
| PVSockBufSend | 632 |
| PVThreadContext | 633 |
| RecvFromParam | 635 |
| RecvParam | 637 |
| SendParam | 638 |
| SendToParam | 639 |
| ShutdownParam | 640 |
| SocketRequestParam | 641 |
| StrCSumPtrLen (Same as StrPtrLen , but includes checksum field and method to speed up querying) | 643 |
| StrPtrLen (This data structure encapsulates a set of functions used to perform) | 646 |
| TimeValue (Time value in a format native to the system) | 648 |
| TLSStorageOps | 655 |
| TReadyQueLink | 656 |
| WStrPtrLen (This data structure encapsulates a set of functions used to perform) | 657 |

Chapter 4

oscl File Index

4.1 oscl File List

Here is a list of all files with brief descriptions:

| | |
|--|-----|
| oscl_aostatus.h (Some basic types used with active objects) | 659 |
| oscl_assert.h (The file oscl_assert.h provides an OSL_ASSERT macro to document assumptions and test them during development) | 660 |
| 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) | 661 |
| oscl_base_alloc.h (A basic allocator that does not rely on other modules) | 662 |
| oscl_base_macros.h (This file defines common macros and constants for basic compilation support) | 663 |
| 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) | 664 |
| oscl_byte_order.h (This file defines functions providing byte ordering utility (e.g., switching between big and little endian orders)) | 665 |
| oscl_defalloc.h (The file defines simple default memory allocator classes. These allocators are used by the OscL_Vector and OscL_Map class, etc) | 666 |
| oscl_dll.h (Defines a DLL entry point) | 667 |
| oscl_dns.h (The file oscl_socket.h defines the OSCL DNS APIs) | 668 |
| oscl_dns_gethostbyname.h | 669 |
| oscl_dns_imp.h | 670 |
| oscl_dns_imp_base.h | 671 |
| oscl_dns_imp_pv.h | 672 |
| oscl_dns_method.h | 673 |
| oscl_dns_param.h | 674 |
| oscl_dns_request.h | 675 |
| oscl_dns_tuneables.h | 676 |
| oscl_double_list.h (Internal use types for scheduler) | 677 |
| oscl_errno.h (Defines functions to access additional information on errors where supported through an errno or similar service) | 678 |
| oscl_error.h (OSCL Error trap and cleanup include file) | 679 |
| oscl_error_allocator.h (Defines a memory allocation class used by the oscl error layer) | 680 |
| oscl_error_codes.h (Defines basic error and leave codes) | 681 |
| oscl_error_imp.h (Internal error implementation support) | 682 |
| oscl_error_imp_cppexceptions.h (Implementation File for Leave using C++ exceptions) | 683 |
| oscl_error_imp_fatalerror.h (Implementation File for Leave using system fatal error) | 684 |
| oscl_error_imp_jumps.h (Implementation of using Setjmp / Longjmp) | 685 |

| | |
|---|-----|
| oscl_error_trapcleanup.h (OSCL Error trap and cleanup implementation include file) | 687 |
| oscl_exception.h (Contains all the exception handling macros and classes) | 688 |
| oscl_exclusive_ptr.h (This file defines the OscExclusivePtr template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error) | 689 |
| oscl_file_async_read.h | 690 |
| oscl_file_cache.h (The file oscl_file_cache.h defines the class OscFileCache) | 691 |
| oscl_file_dir_utils.h (The file oscl_file_dir_utils.h defines some unix-style directory ops) | 692 |
| oscl_file_find.h (The file oscl_file_find.h defines the class Osc_FileFind) | 694 |
| oscl_file_handle.h (The file oscl_file_handle.h defines the class OscFileHandle) | 695 |
| oscl_file_io.h (The file oscl_file_io.h defines the class Osc_File . This is the public API to the basic file I/O operations) | 696 |
| oscl_file_manager.h (File management class) | 697 |
| oscl_file_native.h (The file oscl_file_native.h defines the class OscNativeFile . This is the porting layer for basic file I/O operations) | 698 |
| oscl_file_server.h (The file oscl_file_server.h defines the class Osc_FileServer . This is the porting layer for file server implementations) | 699 |
| oscl_file_stats.h (File stats class) | 700 |
| 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) | 701 |
| oscl_heapbase.h (OSCL Heap Base include file) | 702 |
| oscl_init.h (Global oscl initialization) | 703 |
| oscl_int64_utils.h | 704 |
| oscl_ip_socket.h | 705 |
| oscl_linked_list.h (The file oscl_linked_list.h defines the template class Osc_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) | 706 |
| oscl_lock_base.h (This file defines an abstract lock class, OscLockBase , that is used for APIs potentially requiring multi-thread safety. A null-lock implementation, OscNullLock , is also provided for single-thread configurations (basically a noop for lock/unlock). Also provides the OscScopedLock class which is template class takes care of freeing the lock when the class goes out of scope) | 707 |
| oscl_map.h (The file oscl_map.h defines the template class Osc_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) | 708 |
| oscl_math.h (Provides math functions) | 709 |
| oscl_media_data.h (Defines a container class for media data made up of a collection of memory fragments) | 710 |
| oscl_media_status.h (Defines a status values for the MediaData containers) | 711 |
| oscl_mem.h (This file contains basic memory definitions for common use across platforms) | 712 |
| oscl_mem_align.h | 715 |
| oscl_mem_audit.h (This file contains the definition and partial implementation of MM_Audit class) | 716 |
| oscl_mem_audit_internals.h (This file contains the internal definitions for the mem audit library) | 718 |
| 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) | 719 |
| oscl_mem_basic_functions.h (This file contains prototypes for the basic memory functions) | 720 |
| oscl_mem_inst.h (The file defines default memory instrumentation level) | 721 |
| oscl_mem_mempool.h (This file contains the definition of memory pool allocators) | 722 |
| oscl_mempool_allocator.h (This file contains the definition of memory pool allocator for leave/trap) | 723 |
| oscl_mutex.h (This file provides implementation of mutex) | 724 |
| oscl_namestring.h (Name string class include file) | 725 |

| | |
|--|-----|
| oscl_opaque_type.h (The file oscl_opaque_type.h defines pure virtual classes for working with opaque types) | 726 |
| oscl_priqueue.h (Implements a priority queue data structure similar to STL) | 727 |
| oscl_procstatus.h | 728 |
| 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) | 729 |
| oscl_rand.h (Provides pseudo-random number generation) | 730 |
| oscl_refcounter.h (A general purpose reference counter to object lifetimes) | 731 |
| oscl_refcounter_memfrag.h (This file provides the definition of reference counted memory fragment, which provides access to a buffer and helps manage its manage its lifetime through the refcount) | 732 |
| oscl_registry_access_client.h (Client-side implementation Registry Access implementation) . . . | 733 |
| oscl_registry_client.h (Client-side implementation of OscLRegistry) | 734 |
| oscl_registry_client_impl.h (Client-side implementation of OscLRegistryInterface) | 735 |
| oscl_registry_serv_impl.h (Server-side implementation of OscLRegistry interfaces) | 736 |
| oscl_registry_serv_impl_global.h | 737 |
| oscl_registry_serv_impl_tls.h | 738 |
| oscl_registry_types.h (Common types used in OscL registry interfaces) | 739 |
| oscl_scheduler.h | 740 |
| oscl_scheduler_ao.h (OscL Scheduler user execution object classes) | 741 |
| oscl_scheduler_aobase.h (OscL Scheduler internal active object classes) | 742 |
| oscl_scheduler_readyq.h (Ready q types for oscl scheduler) | 743 |
| oscl_scheduler_threadcontext.h (Thread context functions needed by oscl scheduler) | 744 |
| oscl_scheduler_tuneables.h (Tuneable settings for OscL Scheduler) | 745 |
| oscl_scheduler_types.h (Scheduler common types include file) | 746 |
| oscl_semaphore.h (This file provides implementation of mutex) | 747 |
| oscl_shared_ptr.h (This file defines a template class OscLSharedPtr which is a "smart pointer" to the parameterized type) | 748 |
| oscl_singleton.h (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) | 749 |
| oscl_snprintf.h (Provides a portable implementation of snprintf) | 751 |
| oscl_socket.h (The file oscl_socket.h defines the OSCL Socket APIs) | 752 |
| oscl_socket_accept.h | 753 |
| oscl_socket_bind.h | 754 |
| oscl_socket_connect.h | 755 |
| oscl_socket_imp.h | 756 |
| oscl_socket_imp_base.h | 757 |
| oscl_socket_imp_pv.h | 758 |
| oscl_socket_listen.h | 759 |
| oscl_socket_method.h | 760 |
| oscl_socket_recv.h | 761 |
| oscl_socket_recv_from.h | 762 |
| oscl_socket_request.h | 763 |
| oscl_socket_send.h | 764 |
| oscl_socket_send_to.h | 765 |
| oscl_socket_serv_imp.h | 766 |
| oscl_socket_serv_imp_base.h | 767 |
| oscl_socket_serv_imp_pv.h | 768 |
| oscl_socket_serv_imp_reqlist.h | 769 |
| oscl_socket_shutdown.h | 770 |

| | |
|--|-----|
| oscl_socket_stats.h | 771 |
| oscl_socket_tuneables.h | 773 |
| oscl_socket_types.h | 775 |
| 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) | 777 |
| oscl_str_ptr_len.h (Defines a data structure for string containment/manipulations where the storage for the string is maintained externally) | 779 |
| oscl_string.h (Provides a standardized set of string containers that can be used in place of character arrays) | 780 |
| oscl_string_containers.h (Provides a standardized set of string containers that can be used in place of character arrays) | 781 |
| oscl_string_rep.h (Contains some internal implementation for string containers) | 782 |
| oscl_string_uri.h (Utilities to unescape URIs) | 783 |
| oscl_string_utf8.h (Utilities to validate and truncate UTF-8 encoded strings) | 784 |
| oscl_string_utils.h (Utilities to parse and convert strings) | 785 |
| oscl_string_xml.h (Utilities to escape special characters in XML strings) | 786 |
| oscl_tagtree.h (The file oscl_tagtree.h ..) | 787 |
| oscl_tcp_socket.h | 788 |
| oscl_thread.h | 789 |
| oscl_tickcount.h (Defines a data structure for string containment/manipulations where the storage for the string is maintained externally) | 791 |
| oscl_time.h (The file oscl_time.h defines to 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) | 792 |
| oscl_timer.h | 794 |
| oscl_tls.h | 795 |
| 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) | 796 |
| oscl_types.h (This file contains basic type definitions for common use across platforms) | 797 |
| oscl_udp_socket.h | 798 |
| oscl_utf8conv.h (Utilities to convert unicode to utf8 and vice versa) | 799 |
| oscl_uuid.h (This file defines the OSCL UUID structure used for unique identifiers as well as the short (32-bit) identifiers OsclUid32) | 800 |
| 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) | 802 |
| osclconfig.h (This file contains configuration information for the linux platform) | 803 |
| osclconfig_ansi_memory.h (This file contains common typedefs based on the ANSI C limits.h header) | 805 |
| osclconfig_check.h | 806 |
| osclconfig_compiler_warnings.h (This file contains the ability to turn off/on compiler warnings) | 807 |
| osclconfig_error.h (This file contains the common typedefs and header files needed to compile osclerror) | 808 |
| osclconfig_error_check.h | 809 |
| osclconfig_global_new_delete.h | 810 |
| osclconfig_global_placement_new.h | 811 |
| osclconfig_io.h (This file contains common typedefs based on the ANSI C limits.h header) | 812 |
| osclconfig_io_check.h | 823 |
| osclconfig_ix86.h (This file contains configuration information for the ix86 processor family) | 824 |

| | |
|---|-----|
| osclconfig_lib.h (This file contains configuration information for the ANSI build) | 825 |
| osclconfig_lib_check.h | 826 |
| osclconfig_limits_typedefs.h (This file contains common typedefs based on the ANSI C limits.h header) | 827 |
| osclconfig_memory.h | 828 |
| osclconfig_memory_check.h | 829 |
| osclconfig_no_os.h | 830 |
| osclconfig_proc.h (This file contains configuration information for the linux platform) | 831 |
| osclconfig_proc_check.h | 832 |
| osclconfig_proc_unix_android.h | 834 |
| osclconfig_proc_unix_common.h | 836 |
| osclconfig_time.h | 838 |
| osclconfig_time_check.h | 839 |
| osclconfig_unix_android.h | 840 |
| osclconfig_unix_common.h | 844 |
| osclconfig_util.h | 848 |
| osclconfig_util_check.h | 849 |
| pvlogger.h (This file contains basic logger interfaces for common use across platforms) | 850 |
| pvlogger_accessories.h | 858 |
| pvlogger_c.h (This file contains basic logger interfaces for common use across platforms. C-callable version) | 859 |
| pvlogger_registry.h | 861 |

Chapter 5

oscl Page Index

5.1 oscl Related Pages

Here is a list of all related documentation pages:

| | |
|---------------------|---------------------|
| Todo List | 862 |
|---------------------|---------------------|

Chapter 6

oscl Module Documentation

6.1 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_PRAGMA_PACK` 0
- #define `OSCL_HAS_UNIX_SUPPORT` 0
- #define `OSCL_HAS_MSWIN_SUPPORT` 0
- #define `OSCL_HAS_MSWIN_PARTIAL_SUPPORT` 0
- #define `OSCL_HAS_MSWIN_TIME_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__](#)

6.1.1 Define Documentation

6.1.1.1 `#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.

6.1.1.2 `#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.

6.1.1.3 `#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.

6.1.1.4 `#define OSCL_HAS_BERKELEY_SOCKETS 0`

6.1.1.5 `#define OSCL_HAS_MSWIN_PARTIAL_SUPPORT 0`

6.1.1.6 `#define OSCL_HAS_MSWIN_SUPPORT 0`

6.1.1.7 `#define OSCL_HAS_MSWIN_TIME_SUPPORT 0`

6.1.1.8 `#define OSCL_HAS_PRAGMA_PACK 0`

macro should be set to 1 if the compiler supports pragma pack, 0 if it does not.

6.1.1.9 #define OSCL_HAS_PTHREAD_SUPPORT 0

6.1.1.10 #define OSCL_HAS_PV_C_OS_API_MEMORY_FUNCS 0

6.1.1.11 #define OSCL_HAS_PV_C_OS_SUPPORT 0

6.1.1.12 #define OSCL_HAS_PV_C_OS_TIME_FUNCS 0

6.1.1.13 #define OSCL_HAS_SAVAJE_IO_SUPPORT 0

6.1.1.14 #define OSCL_HAS_SAVAJE_SUPPORT 0

6.1.1.15 #define OSCL_HAS_SEM_TIMEDWAIT_SUPPORT 0

6.1.1.16 #define OSCL_HAS_SYMBIAN_COMPATIBLE_IO_FUNCTION 0

6.1.1.17 #define OSCL_HAS_SYMBIAN_DNS_SERVER 0

6.1.1.18 #define OSCL_HAS_SYMBIAN_ERRORTRAP 0

6.1.1.19 #define OSCL_HAS_SYMBIAN_MATH 0

6.1.1.20 #define OSCL_HAS_SYMBIAN_MEMORY_FUNCS 0

6.1.1.21 #define OSCL_HAS_SYMBIAN_SCHEDULER 0

6.1.1.22 #define OSCL_HAS_SYMBIAN_SOCKET_SERVER 0

6.1.1.23 #define OSCL_HAS_SYMBIAN_SUPPORT 0

6.1.1.24 #define OSCL_HAS_SYMBIAN_TIMERS 0

6.1.1.25 #define OSCL_HAS_UNIX_SUPPORT 0

6.1.1.26 #define OSCL_HAS_UNIX_TIME_FUNCS 0

6.1.1.27 #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.

6.1.2 Typedef Documentation

6.1.2.1 typedef int16 __int16_check__

6.1.2.2 typedef int32 __int32_check__

6.1.2.3 typedef int8 __int8_check__

6.1.2.4 typedef uint16 __uint16_check__

6.1.2.5 typedef uint32 __uint32_check__

6.1.2.6 typedef uint8 __uint8_check__

6.2 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 to 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_DefAllocWithRefCount`
- 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 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 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 indirectly) through `Alloc`. When the `OscLExclusivePtrA` expires, `Alloc` is used to free the memory.

- class `OscLLockBase`
- struct `OscLMemoryFragment`
- class `OscLNullLock`
- class `OscLRefCount`
- class `OscLRefCountDA`

- class [OscRefCounterMemFrag](#)
- class [OscRefCounterMTDA](#)
- class [OscRefCounterMTSA](#)
- class [OscRefCounterSA](#)
- class [OscScopedLock](#)

The [OscScopedLock](#) 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 [OscScopedLock](#) goes out of scope.

- class [OscSharedPtr](#)

A parameterized smart pointer class.

- class [OscTLS](#)
- class [OscTLSRegistry](#)
- 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_VIRTUAL_BASE](#)(type) type
- #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_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 char `ISO8601timeStrBuf` [`ISO8601TIME_BUFFER_SIZE`]
- typedef `OscAny` `TOscTlsKey`
- typedef int `c_bool`
The `c_bool` type is mapped to an integer to provide a bool type for C interfaces.
- typedef void `OscAny`
The `OscAny` 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 `OscFloat`
The `Float` type defined as `OscFloat`.
- 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_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 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_CIstrcmp](#) (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_strcat` (char *dest, const char *src)
- OSCL_IMPORT_REF `oscl_wchar` * `oscl_strcat` (`oscl_wchar` *dest, const `oscl_wchar` *src)
- OSCL_IMPORT_REF void `PV8601ToRFC822` (`PV8601timeStrBuf` pv8601_buffer, `CtimeStrBuf` ctime_buffer)
- OSCL_IMPORT_REF void `ISO8601ToRFC822` (`ISO8601timeStrBuf` iso8601_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)
- OSCL_COND_IMPORT_REF `TimeValue` operator+ (const `TimeValue` &a, const int32 bSeconds)
- OSCL_COND_IMPORT_REF `TimeValue` operator+ (const int32 aSeconds, const `TimeValue` &b)
- OSCL_COND_IMPORT_REF `TimeValue` operator- (const `TimeValue` &a, const int32 bSeconds)
- OSCL_COND_IMPORT_REF `TimeValue` operator- (const int32 aSeconds, 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, `OsclRefCount` *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 int `ISO8601TIME_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

6.2.1 Detailed Description

Additional osclbase comment

Additional osclbase comment

Additional osclbase comment

6.2.2 Define Documentation

6.2.2.1 #define ALLOC_AND_CONSTRUCT(n) alloc_and_construct_fl(n, __FILE__, __LINE__)

6.2.2.2 #define ALLOCATE(n) allocate_fl(n, __FILE__, __LINE__)

6.2.2.3 #define NULL (0)

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

6.2.2.4 #define NULL_TERM_CHAR '\0'

The NULL_TERM_CHAR is used to terminate c-style strings.

6.2.2.5 #define OSCL_ABS(a) ((a) > (0) ? (a) : -(a))

6.2.2.6 #define OSCL_ASSERT(_expr) ((_expr)?((void)0):OSCL_Assert(#_expr, __FILE__, __LINE__))

6.2.2.7 #define OSCL_COND_EXPORT_REF

6.2.2.8 #define OSCL_COND_IMPORT_REF

6.2.2.9 #define OSCL_CONST_CAST(type, exp) ((type)(exp))

Type casting macros.

Parameters:

type Destination type of cast

exp Expression to cast

6.2.2.10 #define OSCL_DISABLE_WARNING_RETURN_TYPE_NOT_UDT

6.2.2.11 #define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE

6.2.2.12 #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\(\)](#)

6.2.2.13 #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\(\)](#)

6.2.2.14 #define OSCL_DYNAMIC_CAST(type, exp) ((type)(exp))

6.2.2.15 #define OSCL_HAS_SINGLETON_SUPPORT 1

6.2.2.16 #define OSCL_INLINE inline

6.2.2.17 #define OSCL_MAX(a, b) ((a) > (b) ? (a) : (b))

6.2.2.18 #define OSCL_MIN(a, b) ((a) < (b) ? (a) : (b))

6.2.2.19 #define OSCL_PACKED_VAR "error"

6.2.2.20 #define OSCL_REINTERPRET_CAST(type, exp) ((type)(exp))

6.2.2.21 #define OSCL_STATIC_CAST(type, exp) ((type)(exp))

6.2.2.22 #define OSCL_TEMPLATED_DESTRUCTOR_CALL(type, simple_type) type :: ~simple_type ()

6.2.2.23 #define OSCL_TLS_BASE_SLOTS [OSCL_TLS_ID_BASE_LAST](#) +1

6.2.2.24 #define OSCL_TLS_EXTERNAL_SLOTS 0

6.2.2.25 #define OSCL_TLS_MAX_SLOTS (OSCL_TLS_BASE_SLOTS + OSCL_TLS_EXTERNAL_SLOTS)

6.2.2.26 #define OSCL_UNSIGNED_CONST(x) x

6.2.2.27 #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.

6.2.2.28 #define OSCL_UNUSED_RETURN(value) return value

6.2.2.29 #define OSCL_VIRTUAL_BASE(type) type

6.2.2.30 #define PVMEM_INST_LEVEL 1

6.2.3 Typedef Documentation

6.2.3.1 typedef int c_bool

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

6.2.3.2 typedef char CtimeStrBuf[CTIME_BUFFER_SIZE]

6.2.3.3 typedef OSCL_NATIVE_INT64_TYPE int64

6.2.3.4 typedef char ISO8601timeStrBuf[ISO8601TIME_BUFFER_SIZE]

6.2.3.5 typedef char mbchar

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

6.2.3.6 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.

6.2.3.7 typedef oscl_wchar OSCL_TCHAR

```
define OSCL_TCHAR
```

6.2.3.8 typedef OSCL_NATIVE_WCHAR_TYPE oscl_wchar

6.2.3.9 typedef void OsclAny

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

6.2.3.10 typedef float OsclFloat

The Float type defined as OsclFloat.

6.2.3.11 typedef char PV8601timeStrBuf[PV8601TIME_BUFFER_SIZE]

6.2.3.12 typedef [OscAny](#) TOscTlsKey

6.2.3.13 typedef unsigned int uint

The uint type is a convenient abbreviation for unsigned int.

6.2.3.14 typedef OSCL_NATIVE_UINT64_TYPE uint64

6.2.4 Enumeration Type Documentation

6.2.4.1 enum TimeUnits

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

Enumeration values:

SECONDS

MILLISECONDS

MICROSECONDS

6.2.5 Function Documentation

6.2.5.1 OSCL_COND_IMPORT_REF void _OSCL_Abort ()

This function terminates the current process abnormally.

6.2.5.2 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.

6.2.5.3 template<class TheClass> void OscSharedPtr< TheClass >::Bind (TheClass * *ptr*,
[OscRefCount](#) * *in_refcnt*) [inline, inherited]

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

6.2.5.4 template<class TheClass> void OscSharedPtr< TheClass >::Bind (const
[OscSharedPtr](#)< TheClass > & *inHandle*) [inline, inherited]

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

6.2.5.5 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.

6.2.5.6 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.

6.2.5.7 OSCL_IMPORT_REF void ISO8601ToRFC822 (ISO8601timeStrBuf *iso8601_buffer*, CtimeStrBuf *ctime_buffer*)**6.2.5.8 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.

- 6.2.5.9** OSCL_COND_IMPORT_REF **TimeValue** operator+ (const int32 *aSeconds*, const **TimeValue** & *b*)
- 6.2.5.10** OSCL_COND_IMPORT_REF **TimeValue** operator+ (const **TimeValue** & *a*, const int32 *bSeconds*)
- 6.2.5.11** OSCL_COND_IMPORT_REF **TimeValue** operator- (const int32 *aSeconds*, const **TimeValue** & *b*)
- 6.2.5.12** OSCL_COND_IMPORT_REF **TimeValue** operator- (const **TimeValue** & *a*, const int32 *bSeconds*)
- 6.2.5.13** OSCL_COND_IMPORT_REF **TimeValue** operator- (const **TimeValue** & *a*, const **TimeValue** & *b*)
- 6.2.5.14** template<class **TheClass**> bool OsciSharedPtr< **TheClass** >::operator==(const OsciSharedPtr< **TheClass** > & *b*) const [inline, inherited]

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

- 6.2.5.15** 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

- 6.2.5.16** OSCL_IMPORT_REF int32 osci_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

- 6.2.5.17** OSCL_IMPORT_REF int32 osci_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

6.2.5.18 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

6.2.5.19 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

6.2.5.20 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.

6.2.5.21 OSCL_IMPORT_REF oscl_wchar* oscl_strcat (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

6.2.5.22 OSCL_IMPORT_REF char* oscl_strcat (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

6.2.5.23 OSCL_IMPORT_REF oscl_wchar* oscl_strchr (oscl_wchar * str, int32 c)**6.2.5.24 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:**6.2.5.25 OSCL_IMPORT_REF char* oscl_strchr (char * str, int32 c)****6.2.5.26 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:**6.2.5.27 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

6.2.5.28 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

6.2.5.29 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.

6.2.5.30 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.

6.2.5.31 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

6.2.5.32 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

6.2.5.33 OSCL_IMPORT_REF int32 oscl_strncmp (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

6.2.5.34 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

6.2.5.35 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*.

6.2.5.36 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*.

6.2.5.37 OSCL_IMPORT_REF `oscl_wchar*` `oscl_strchr` (`oscl_wchar * str, int32 c`)

6.2.5.38 OSCL_IMPORT_REF `const oscl_wchar*` `oscl_strchr` (`const oscl_wchar * str, int32 c`)

6.2.5.39 OSCL_IMPORT_REF `char*` `oscl_strchr` (`char * str, int32 c`)

6.2.5.40 OSCL_IMPORT_REF `const char*` `oscl_strchr` (`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:

6.2.5.41 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*

6.2.5.42 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*

6.2.5.43 OSCL_IMPORT_REF `oscl_wchar*` `oscl_strstr` (`oscl_wchar * str1, const oscl_wchar * str2`)

6.2.5.44 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 beginning of sub-string.

6.2.5.45 OSCL_IMPORT_REF char* oscl_strstr (char * *str1*, const char * *str2*)**6.2.5.46 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 beginning of sub-string.

6.2.5.47 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.

6.2.5.48 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.

6.2.5.49 OSCL_IMPORT_REF void PV8601ToRFC822 ([PV8601timeStrBuf](#) *pv8601_buffer*, [CtimeStrBuf](#) *ctime_buffer*)**6.2.5.50 void PVOsclBase_Cleanup ()**

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

6.2.5.51 void PVOsclBase_Init ()

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

Exceptions:

leaves if out-of-memory

6.2.5.52 OSCL_IMPORT_REF void RFC822ToPV8601 (CtimeStrBuf ctime_buffer, PV8601timeStrBuf)
6.2.6 Variable Documentation

6.2.6.1 const int CTIME_BUFFER_SIZE = 26

6.2.6.2 const int ISO8601TIME_BUFFER_SIZE = 21

6.2.6.3 const long MSEC_PER_SEC = 1000

6.2.6.4 const uint32 OSCL_TLS_ID_BASE_LAST = 11

6.2.6.5 const uint32 OSCL_TLS_ID_ERRORHOOK = 1

6.2.6.6 const uint32 OSCL_TLS_ID_MAGICNUM = 0

6.2.6.7 const uint32 OSCL_TLS_ID_OSCLREGISTRY = 10

6.2.6.8 const uint32 OSCL_TLS_ID_PAYLOADPARSER = 7

6.2.6.9 const uint32 OSCL_TLS_ID_PERRORTRAP = 5

6.2.6.10 const uint32 OSCL_TLS_ID_PVLOGGER = 2

6.2.6.11 const uint32 OSCL_TLS_ID_PVMFRECOGNIZER = 8

6.2.6.12 const uint32 OSCL_TLS_ID_PVSCHEDULER = 4

6.2.6.13 const uint32 OSCL_TLS_ID_SDPMEDIAPARSER = 6

6.2.6.14 const uint32 OSCL_TLS_ID_SQLITE3 = 11

6.2.6.15 const uint32 OSCL_TLS_ID_TEST = 3

6.2.6.16 const uint32 OSCL_TLS_ID_WMDRM = 9

6.2.6.17 const int PV8601TIME_BUFFER_SIZE = 21

6.2.6.18 const uint32 unix_ntp_offset = 2208988800U

6.2.6.19 const long USEC_PER_SEC = 1000000

6.3 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 [OscIAuditCB](#)
- class [OscIMem](#)
- class [OscIMemAllocator](#)
- class [OscIMemAllocator](#)
- class [OscIMemAllocDestructDealloc](#)
- class [OscIMemAllocDestructDealloc](#)
- class [OscIMemAudit](#)
- 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 indirectly) by new. When the oscl_auto_ptr expires, its destructor uses delete to free the memory.

- class [OscMemBasicAllocator](#)
- class [OscMemBasicAllocator](#)
- class [OscMemBasicAllocDestructDealloc](#)
- class [OscMemBasicAllocDestructDealloc](#)
- class [OscMemGlobalAuditObject](#)
- class [OscMemPoolFixedChunkAllocator](#)
- class [OscMemPoolFixedChunkAllocatorObserver](#)
- class [OscMemPoolResizableAllocator](#)
- class [OscMemPoolResizableAllocatorMemoryObserver](#)
- class [OscMemPoolResizableAllocatorObserver](#)
- class [OscMemStatsNode](#)

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, Osci_TAlloc< char, OsciMemBasicAllocator > >` `MMAudit-CharAutoPtr`
- typedef `OSCLMemAutoPtr< uint8, Osci_TAlloc< uint8, _OsciBasicAllocator > >` `MMAudit-Uint8AutoPtr`
- typedef `OSCLMemAutoPtr< MM_AllocNode, Osci_TAlloc< MM_AllocNode, OsciMemBasicAllocator > >` `MM_AllocNodeAutoPtr`
- typedef `OSCLMemAutoPtr< OsciMemStatsNode, Osci_TAlloc< OsciMemStatsNode, OsciMemBasicAllocator > >` `MM_StatsNodeTagTreeType`
- typedef `OSCLMemAutoPtr< OsciMemStatsNode, Osci_TAlloc< OsciMemStatsNode, OsciMemBasicAllocator > >` `OsciMemStatsNodeAutoPtr`
- typedef `Osci_TAlloc< MM_StatsNodeTagTreeType, OsciMemBasicAllocator >` `TagTree_Allocator`
- typedef `Osci_TagTree< MM_StatsNodeTagTreeType, TagTree_Allocator >` `OsciTagTreeType`

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 `OscMemInit` (OscAuditCB &auditCB)
- OSCL_IMPORT_REF void * `_oscl_audit_malloc` (size_t, OscAuditCB &, const char *f=NULL, const int l=0)
- OSCL_IMPORT_REF void * `_oscl_audit_calloc` (size_t, size_t, OscAuditCB &, const char *f=NULL, const int l=0)
- OSCL_IMPORT_REF void * `_oscl_audit_realloc` (void *, size_t, OscAuditCB &, const char *f=NULL, const int l=0)
- OSCL_IMPORT_REF void * `_oscl_audit_new` (size_t, OscAuditCB &, 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

6.3.1 Define Documentation

6.3.1.1 #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.

6.3.1.2 #define _OSCL_TRAP_NEW(exp, freeFunc, T_ptr, T, params)

Value:

```
{\
  int32 __err;\
  OscAny* __ptr=exp;\
  OSCL_TRY(__err,T_ptr=new(__ptr) T params);\
  if(__err){\
    freeFunc(__ptr);\
    T_ptr=NULL;\
    OscError::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.

6.3.1.3 #define COMPUTE_MEM_ALIGN_SIZE(x, y, z) (y+(((x+y)%z) ? (z - (x+y)%z) : 0))

6.3.1.4 #define DEFAULT_MM_AUDIT_MODE 0

6.3.1.5 #define DEFAULT_POSTFILL_PATTERN 0x5A

6.3.1.6 #define DEFAULT_PREFILL_PATTERN 0x96

6.3.1.7 #define FENCE_PATTERN 0xAA

6.3.1.8 #define MEM_ALIGN_SIZE 8

6.3.1.9 #define MIN_FENCE_SIZE 4

6.3.1.10 #define MM_ALLOC_MAX_QUERY_FILENAME_LEN 128

6.3.1.11 #define MM_ALLOC_MAX_QUERY_TAG_LEN 64

6.3.1.12 #define MM_AUDIT_ALLOC_NODE_ENABLE_FLAG 0x10

6.3.1.13 #define MM_AUDIT_ALLOC_NODE_SUPPORT 1

6.3.1.14 #define MM_AUDIT_FAILURE_SIMULATION_SUPPORT 1

6.3.1.15 #define MM_AUDIT_FENCE_SUPPORT 0

6.3.1.16 #define MM_AUDIT_FILL_SUPPORT 0

6.3.1.17 #define MM_AUDIT_INCLUDE_ALL_HEAP_VALIDATION 1

6.3.1.18 #define MM_AUDIT_POSTFILL_FLAG 0x2

6.3.1.19 #define MM_AUDIT_PREFILL_FLAG 0x1

6.3.1.20 #define MM_AUDIT_SUPPRESS_FILENAME_FLAG 0x20

6.3.1.21 #define MM_AUDIT_VALIDATE_ALL_HEAP_FLAG 0x4

6.3.1.22 #define MM_AUDIT_VALIDATE_BLOCK 1

6.3.1.23 #define MM_AUDIT_VALIDATE_ON_FREE_FLAG 0x8

6.3.1.24 #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

**6.3.1.25 #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

6.3.1.26 #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

6.3.1.27 #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

**6.3.1.28 #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

**6.3.1.29 #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

6.3.1.30 #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

6.3.1.31 #define OSCL_AUDIT_NEW(auditCB, T, params) new(_oscl_audit_new(sizeof(T),auditCB)) T params

Osc "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

6.3.1.32 #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

6.3.1.33 #define osc1_malloc(a, b) OSCL_CALLOC(a,b)**6.3.1.34 #define OSCL_CALLOC(num, size) _oscl_default_audit_malloc(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

6.3.1.35 #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.

6.3.1.36 #define OSCL_DEFAULT_FREE(x) OSCL_FREE(x)

Another back-compatibility definition.

6.3.1.37 #define OSCL_DEFAULT_MALLOC(x) OSCL_MALLOC(x)

Another back-compatibility definition.

6.3.1.38 #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

6.3.1.39 #define OSCL_DISABLE_WARNING_RETURN_TYPE_NOT_UDT
6.3.1.40 #define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE

Previously this was in oscl_mem_imp.h

6.3.1.41 #define oscl_free(x) OSCL_FREE(x)
6.3.1.42 #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

6.3.1.43 #define OSCL_HAS_GLOBAL_NEW_DELETE 1

6.3.1.44 #define oscl_malloc(a) OSCL_MALLOC(a)

6.3.1.45 #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

6.3.1.46 #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

6.3.1.47 #define OSCL_PLACEMENT_NEW(ptr, constructor) new(ptr) constructor

6.3.1.48 #define oscl_realloc(a, b) OSCL_REALLOC(a,b)

6.3.1.49 #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

**6.3.1.50 #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

6.3.1.51 #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

6.3.1.52 #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

6.3.2 Typedef Documentation

6.3.2.1 typedef **OSCLMemAutoPtr**<MM_AllocNode, **OscI_TAlloc**<MM_AllocNode, **OscIMemBasicAllocator**> > **MM_AllocNodeAutoPtr**

6.3.2.2 typedef **OSCLMemAutoPtr**<**OscIMemStatsNode**, **OscI_TAlloc**<**OscIMemStatsNode**, **OscIMemBasicAllocator**> > **MM_StatsNodeTagTreeType**

6.3.2.3 typedef **OSCLMemAutoPtr**<char, **OscI_TAlloc**<char, **OscIMemBasicAllocator**> > **MMAuditCharAutoPtr**

6.3.2.4 typedef **OSCLMemAutoPtr**<uint8, **OscI_TAlloc**<uint8, **_OscIBasicAllocator**> > **MMAuditUint8AutoPtr**

6.3.2.5 typedef **OSCLMemAutoPtr**<**OscIMemStatsNode**, **OscI_TAlloc**<**OscIMemStatsNode**, **OscIMemBasicAllocator**> > **OscIMemStatsNodeAutoPtr**

6.3.2.6 typedef **OscI_TagTree**<**MM_StatsNodeTagTreeType**, **TagTree_Allocator**> **OscI_TagTreeType**

6.3.2.7 typedef **OscI_TAlloc**<**MM_StatsNodeTagTreeType**, **OscIMemBasicAllocator**> **TagTree_Allocator**

6.3.3 Function Documentation

6.3.3.1 **OSCL_IMPORT_REF** void* **_oscl_audit_calloc** (size_t, size_t, **OscIAuditCB** &, const char **f* = NULL, const int *l* = 0)

6.3.3.2 **OSCL_IMPORT_REF** void **_oscl_audit_free** (void *)

6.3.3.3 **OSCL_IMPORT_REF** void* **_oscl_audit_malloc** (size_t, **OscIAuditCB** &, const char **f* = NULL, const int *l* = 0)

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

6.3.3.4 OSCL_IMPORT_REF void* _oscl_audit_new (size_t, OslAuditCB &, const char *f = NULL, const int l = 0)

6.3.3.5 OSCL_IMPORT_REF void* _oscl_audit_realloc (void *, size_t, OslAuditCB &, const char *f = NULL, const int l = 0)

6.3.3.6 OSCL_COND_IMPORT_REF void* _oscl_calloc (int32 nelems, int32 size)

6.3.3.7 OSCL_IMPORT_REF void* _oscl_default_audit_calloc (size_t, size_t, const char *f = NULL, const int l = 0)

6.3.3.8 OSCL_IMPORT_REF void* _oscl_default_audit_malloc (size_t, const char *f = NULL, const int l = 0)

6.3.3.9 OSCL_IMPORT_REF void* _oscl_default_audit_new (size_t, const char *f = NULL, const int l = 0)

6.3.3.10 OSCL_IMPORT_REF void* _oscl_default_audit_realloc (void *, size_t, const char *f = NULL, const int l = 0)

6.3.3.11 OSCL_COND_IMPORT_REF void _oscl_free (void *src)

6.3.3.12 OSCL_COND_IMPORT_REF void* _oscl_malloc (int32 count)

6.3.3.13 OSCL_COND_IMPORT_REF void* _oscl_realloc (void *src, int32 count)

6.3.3.14 void operator delete (void *) [inline]

6.3.3.15]

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

6.3.3.16 void* operator new (size_t) [inline]

6.3.3.17 void* operator new (size_t aSize, const char *aFile, int aLine) [inline]

6.3.3.18]

void* operator new[] (size_t aSize) [inline]

6.3.3.19]

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

6.3.3.20 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

6.3.3.21 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***6.3.3.22 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`**6.3.3.23 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`

6.3.3.24 OSCL_COND_IMPORT_REF void* oscl_memmove32 (void * *dest*, const void * *src*, uint32 *count*)

Same functionality as `oscl_memmove`, yet optimized for memory alligned on 32-bit boundary

Parameters:

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

Returns:

the value of *dest*

6.3.3.25 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*

6.3.3.26 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

6.3.4 Variable Documentation**6.3.4.1 const uint32 MM_AllocBlockHdr::ALLOC_NODE_FLAG = 0x80000000 [static, inherited]**

6.4 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_priqueue.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 *OscRegistry*.*
- file [oscl_registry_client_impl.h](#)
*Client-side implementation of *OscRegistryInterface*.*
- file [oscl_registry_serv_impl.h](#)
*Server-side implementation of *OscRegistry* interfaces.*
- file [oscl_registry_types.h](#)
*Common types used in *Osc* 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 [CHeapRep](#)
- 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 [OscIBinIStream](#)
- class [OscIBinIStreamBigEndian](#)
- class [OscIBinIStreamLittleEndian](#)

- class [OscBinOStream](#)
Class [OscBinOStream](#) implements the basic stream functions for an output stream.
- class [OscBinOStreamBigEndian](#)
Class [OscBinOStreamBigEndian](#) implements a binary output stream using big endian byte ordering.
- class [OscBinOStreamLittleEndian](#)
Class [OscBinOStreamLittleEndian](#) implements a binary output stream using little endian byte ordering.
- class [OscBinStream](#)
- class [OscCompareLess](#)
- class [OscComponentRegistry](#)
- class [OscComponentRegistryData](#)
- class [OscComponentRegistryElement](#)
- class [OscPriorityQueue](#)
- class [OscPriorityQueueBase](#)
- class [OscRand](#)
- class [OscRegistryAccessClient](#)
- class [OscRegistryAccessClientImpl](#)
- class [OscRegistryAccessClientTlsImpl](#)
- class [OscRegistryAccessElement](#)
- class [OscRegistryClient](#)
- class [OscRegistryClientImpl](#)
- class [OscRegistryClientTlsImpl](#)
- class [OscRegistryServTlsImpl](#)
- class [OscTickCount](#)
- 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
- #define [MAX_NUMBER_OF_BYTE_PER_UTF8](#) 3

Typedefs

- typedef [OscAny](#) * [OscComponentFactory](#)
- 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](#)

Enumerations

- enum [TOSCL_StringOp](#) { [EOSCL_StringOp_CompressASCII](#), [EOSCL_StringOp_UTF16ToUTF8](#) }
- enum [TOSCL_wStringOp](#) { [EOSCL_wStringOp_ExpandASCII](#), [EOSCL_wStringOp_UTF8ToUTF16](#) }

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, [OscFloat](#) &value)
- [OSCL_IMPORT_REF](#) bool [PV_atof](#) (const char *buf, int length, [OscFloat](#) &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_UnicodeToUTF8](#) (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)
- void [set](#) (const other_chartype *buf, optype op)
- void [set](#) (const other_chartype *buf, optype op)
- void [set](#) (const other_chartype *buf, uint32 length, optype op)
- void [set](#) (const other_chartype *buf, uint32 length, optype op)
- [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)
- void [set](#) (const other_chartype *buf, optype op)
- void [set](#) (const other_chartype *buf, optype op)
- void [set](#) (const other_chartype *buf, uint32 length, optype op)
- void [set](#) (const other_chartype *buf, uint32 length, optype op)
- [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

6.4.1 Define Documentation

6.4.1.1 `#define MAX_NUMBER_OF_BYTE_PER_UTF8 3`

Define the maximum UTF8 representation in bytes.

Todo:

Handle 4-byte surrogate pair representation

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

6.4.1.3 `#define OSCLTICKCOUNT_MAX_TICKS 0xffffffff`

6.4.2 Typedef Documentation

6.4.2.1 `typedef void(* BufferFreeFuncPtr)(void *)`

6.4.2.2 `typedef uint32 MediaTimestamp`

6.4.2.3 `typedef WStrPtrLen OSCL_TStrPtrLen`

6.4.2.4 `typedef OsclAny* OsclComponentFactory`

`OsclComponentFactory` is an opaque pointer.

6.4.2.5 `typedef StrCsumPtrLen StrCsumPtrLen`

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

6.4.2.6 `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.

6.4.2.7 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.

6.4.3 Enumeration Type Documentation

6.4.3.1 enum TOSCL_StringOp

Conversion operations for [OSCL_String](#) classes

Enumeration values:

`EOSCL_StringOp_CompressASCII`

`EOSCL_StringOp_UTF16ToUTF8`

6.4.3.2 enum TOSCL_wStringOp

Conversion operations for [OSCL_wString](#) classes

Enumeration values:

`EOSCL_wStringOp_ExpandASCII`

`EOSCL_wStringOp_UTF8ToUTF16`

6.4.4 Function Documentation

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

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

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

Implements [OSCL_wString](#).

6.4.4.4 `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](#).

6.4.4.5 `template<class Alloc> const OSCL_wHeapString< Alloc >::chartype * OSCL_wHeapString< Alloc >::get_cstr () [virtual, inherited]`

Implements [OSCL_wString](#).

6.4.4.6 `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](#).

6.4.4.7 `template<uint32 MaxBufSize> uint32 OSCL_wStackString< MaxBufSize >::get_maxsize ()` [virtual, inherited]

Implements [OSCL_wString](#).

6.4.4.8 `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](#).

6.4.4.9 `template<class Alloc> uint32 OSCL_wHeapString< Alloc >::get_maxsize ()` [virtual, inherited]

Implements [OSCL_wString](#).

6.4.4.10 `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](#).

6.4.4.11 `template<uint32 MaxBufSize> uint32 OSCL_wStackString< MaxBufSize >::get_size ()` [virtual, inherited]

Implements [OSCL_wString](#).

6.4.4.12 `template<uint32 MaxBufSize> uint32 OSCL_StackString< MaxBufSize >::get_size ()` [virtual, inherited]

Pure virtuals from [OSCL_String](#)

Implements [OSCL_String](#).

6.4.4.13 `template<class Alloc> uint32 OSCL_wHeapString< Alloc >::get_size ()` [virtual, inherited]

Implements [OSCL_wString](#).

6.4.4.14 `template<class Alloc> uint32 OSCL_HeapString< Alloc >::get_size ()` [virtual, inherited]

Pure virtuals from [OSCL_String](#)

Implements [OSCL_String](#).

6.4.4.15 `template<uint32 MaxBufSize> OSCL_wStackString< MaxBufSize >::chartype * OSCL_wStackString< MaxBufSize >::get_str ()` [virtual, inherited]

Implements [OSCL_wString](#).

6.4.4.16 `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](#).

6.4.4.17 `template<class Alloc> OSCL_wHeapString< Alloc >::chartype * OSCL_wHeapString< Alloc >::get_str ()` [virtual, inherited]

Implements [OSCL_wString](#).

6.4.4.18 `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](#).

6.4.4.19 `template<class ChainClass, uint32 max_frags> BufferState * BufFragGroup< ChainClass, max_frags >::GetBufferState (const int32 idx)` [inline, inherited]

6.4.4.20 `template<class ChainClass, uint32 max_frags> BufferFragment * BufFragGroup< ChainClass, max_frags >::GetFragment (const int32 idx)` [inline, inherited]

6.4.4.21 `template<uint32 MaxBufSize> OSCL_wStackString< MaxBufSize > & OSCL_wStackString< MaxBufSize >::operator= (const chartype * cstr)` [inherited]

Reimplemented from [OSCL_wString](#).

6.4.4.22 `template<uint32 MaxBufSize> OSCL_StackString< MaxBufSize > & OSCL_StackString< MaxBufSize >::operator= (const chartype * cstr)` [inherited]

Assignment operator

am: null-terminated string

Reimplemented from [OSCL_String](#).

6.4.4.23 `template<uint32 MaxBufSize> OSCL_wStackString< MaxBufSize > & OSCL_wStackString< MaxBufSize >::operator= (const OSCL_wString & src)`
 [inherited]

Reimplemented from [OSCL_wString](#).

6.4.4.24 `template<uint32 MaxBufSize> OSCL_StackString< MaxBufSize > & OSCL_StackString< MaxBufSize >::operator= (const OSCL_String & src)`
 [inherited]

Assignment operator

Reimplemented from [OSCL_String](#).

6.4.4.25 `template<uint32 MaxBufSize> OSCL_wStackString< MaxBufSize > & OSCL_wStackString< MaxBufSize >::operator= (const OSCL_wStackString< MaxBufSize > & src)` [inherited]

6.4.4.26 `template<uint32 MaxBufSize> OSCL_StackString< MaxBufSize > & OSCL_StackString< MaxBufSize >::operator= (const OSCL_StackString< MaxBufSize > & src)` [inherited]

Assignment operators

6.4.4.27 `template<class Alloc> OSCL_wHeapString< Alloc > & OSCL_wHeapString< Alloc >::operator= (const chartype * cstr)` [inherited]

Reimplemented from [OSCL_wString](#).

6.4.4.28 `template<class Alloc> OSCL_HeapString< Alloc > & OSCL_HeapString< Alloc >::operator= (const chartype * cstr)` [inherited]

Assignment operator

am: null-terminated string

Reimplemented from [OSCL_String](#).

6.4.4.29 `template<class Alloc> OSCL_wHeapString< Alloc > & OSCL_wHeapString< Alloc >::operator= (const OSCL_wString & src)` [inherited]

Reimplemented from [OSCL_wString](#).

6.4.4.30 `template<class Alloc> OSCL_HeapString< Alloc > & OSCL_HeapString< Alloc >::operator= (const OSCL_String & src)` [inherited]

Assignment operator

Reimplemented from [OSCL_String](#).

6.4.4.31 `template<class Alloc> OSCL_wHeapString< Alloc > & OSCL_wHeapString< Alloc >::operator= (const OSCL_wHeapString< Alloc > & src) [inherited]`

6.4.4.32 `template<class Alloc> OSCL_HeapString< Alloc > & OSCL_HeapString< Alloc >::operator= (const OSCL_HeapString< Alloc > & src) [inherited]`

Assignment operators

6.4.4.33 `OSCL_IMPORT_REF int oscl_abs (int aVal)`

6.4.4.34 `OSCL_COND_IMPORT_REF double oscl_asin (double value)`

Calculates the arc since of a number

Parameters:

value source value

6.4.4.35 `OSCL_COND_IMPORT_REF double oscl_atan (double value)`

Calculates the arc tangent of a number

Parameters:

value source value

6.4.4.36 `OSCL_COND_IMPORT_REF double oscl_cos (double value)`

Calculates the cosine of a number

Parameters:

value source value

6.4.4.37 `OSCL_COND_IMPORT_REF double oscl_exp (double value)`

Calculates the exponential of e for a number

Parameters:

value source value

6.4.4.38 `OSCL_COND_IMPORT_REF double oscl_floor (double value)`

Calculates the floor of a number

Parameters:

value source value

6.4.4.39 `template<class Alloc> OSCL_HeapString< Alloc >::OSCL_HeapString (const OSCL_String & src)` [inherited]

6.4.4.40 `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.

6.4.4.41 `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.

6.4.4.42 `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.

6.4.4.43 `template<class Alloc> OSCL_HeapString< Alloc >::OSCL_HeapString ()` [inherited]

The default constructor creates an empty string.

6.4.4.44 `OSCL_COND_IMPORT_REF double oscl_log (double value)`

Calculates the natural log of a number

Parameters:

value source value

6.4.4.45 `OSCL_COND_IMPORT_REF double oscl_log10 (double value)`

Calculates the logarithm to base 10 of a number

Parameters:

value source value

6.4.4.46 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

6.4.4.47 OSCL_COND_IMPORT_REF double oscl_sin (double *value*)

Calculates the sine of a number

Parameters:

- value* source value

6.4.4.48 OSCL_IMPORT_REF int32 oscl_snprintf (oscl_wchar * *str*, uint32 *count*, const oscl_wchar * *fmt*, ...)

6.4.4.49 OSCL_IMPORT_REF int32 oscl_snprintf (char * *str*, uint32 *count*, const char * *fmt*, ...)

6.4.4.50 OSCL_COND_IMPORT_REF double oscl_sqrt (double *value*)

Calculates the square root of a number

Parameters:

- value* source value

6.4.4.51 template<uint32 MaxBufSize> OSCL_StackString< MaxBufSize >::OSCL_StackString (const OSCL_String & *src*) [inherited]

6.4.4.52 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.

6.4.4.53 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.

6.4.4.54 `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.

6.4.4.55 `template<uint32 MaxBufSize> OSCL_StackString< MaxBufSize >::OSCL_StackString () [inherited]`

Creates an OSCL_StackString initialized with an empty string.

6.4.4.56 `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_outing 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.

6.4.4.57 `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.

6.4.4.58 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.

6.4.4.59 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.

6.4.4.60 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.

6.4.4.61 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.

6.4.4.62 OSCL_COND_IMPORT_REF `double oscl_tan (double value)`

Calculates the tangential of a number

Parameters:

- value* source value

6.4.4.63 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 buffer 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' termination is not necessary.

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

6.4.4.64 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 buffer 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

- 6.4.4.65 **OSCL_IMPORT_REF** int32 oscl_vsnprintf (**oscl_wchar** * *str*, uint32 *count*, const **oscl_wchar** * *fmt*, va_list *args*)
- 6.4.4.66 **OSCL_IMPORT_REF** int32 oscl_vsnprintf (char * *str*, uint32 *count*, const char * *fmt*, va_list *args*)
- 6.4.4.67 template<class Alloc> OSCL_wHeapString< Alloc >::OSCL_wHeapString (const **OSCL_wString** & *src*) [inherited]
- 6.4.4.68 template<class Alloc> OSCL_wHeapString< Alloc >::OSCL_wHeapString (const OSCL_wHeapString< Alloc > & *src*) [inherited]
- 6.4.4.69 template<class Alloc> OSCL_wHeapString< Alloc >::OSCL_wHeapString (const **chartype** * *buf*, uint32 *length*) [inherited]
- 6.4.4.70 template<class Alloc> OSCL_wHeapString< Alloc >::OSCL_wHeapString (const **chartype** * *cstr*) [inherited]
- 6.4.4.71 template<class Alloc> OSCL_wHeapString< Alloc >::OSCL_wHeapString () [inherited]
- 6.4.4.72 template<uint32 MaxBufSize> OSCL_wStackString< MaxBufSize >::OSCL_wStackString (const **OSCL_wString** & *src*) [inherited]
- 6.4.4.73 template<uint32 MaxBufSize> OSCL_wStackString< MaxBufSize >::OSCL_wStackString (const OSCL_wStackString< MaxBufSize > & *src*) [inherited]
- 6.4.4.74 template<uint32 MaxBufSize> OSCL_wStackString< MaxBufSize >::OSCL_wStackString (const **chartype** * *buf*, uint32 *length*) [inherited]
- 6.4.4.75 template<uint32 MaxBufSize> OSCL_wStackString< MaxBufSize >::OSCL_wStackString (const **chartype** * *cstr*) [inherited]
- 6.4.4.76 template<uint32 MaxBufSize> OSCL_wStackString< MaxBufSize >::OSCL_wStackString () [inherited]
- 6.4.4.77 **OSCL_IMPORT_REF** bool PV_atof (const char * *buf*, int *length*, **OsclFloat** & *value*)
- 6.4.4.78 **OSCL_IMPORT_REF** bool PV_atof (const char * *buf*, **OsclFloat** & *value*)
- 6.4.4.79 **OSCL_IMPORT_REF** bool PV_atoi (const char * *buf*, const char *new_format*, int *length*, **uint64** & *value*)
- 6.4.4.80 **OSCL_IMPORT_REF** bool PV_atoi (const char * *buf*, const char *new_format*, int *length*, **uint32** & *value*)
- 6.4.4.81 **OSCL_IMPORT_REF** bool PV_atoi (const char * *buf*, const char *new_format*, **uint32** & *value*)
- 6.4.4.82 template<uint32 MaxBufSize> void OSCL_wStackString< MaxBufSize >::set (const **other_chartype** * *buf*, uint32 *length*, **optype** *op*) [inherited]
- 6.4.4.83 template<uint32 MaxBufSize> void OSCL_StackString< MaxBufSize >::set (const **other_chartype** * *buf*, uint32 *length*, **optype** *op*) [inherited]

Parameters:

- buf*: string or character array.
- length*: number of characters to copy.
- op*: conversion operation to apply

6.4.4.84 `template<uint32 MaxBufSize> void OSCL_wStackString< MaxBufSize >::set (const other_chartype * buf, optype op)` [inherited]

6.4.4.85 `template<uint32 MaxBufSize> void OSCL_StackString< MaxBufSize >::set (const other_chartype * buf, optype op)` [inherited]

Set the contents of this string to a new string, with conversion operation.

Parameters:

- buf*: NULL-terminated wide string.
- op*: conversion operation to apply

6.4.4.86 `template<uint32 MaxBufSize> void OSCL_wStackString< MaxBufSize >::set (const chartype * buf, uint32 length)` [inherited]

6.4.4.87 `template<uint32 MaxBufSize> void OSCL_StackString< MaxBufSize >::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.

6.4.4.88 `template<class Alloc> void OSCL_wHeapString< Alloc >::set (const other_chartype * buf, uint32 length, optype op)` [inherited]

6.4.4.89 `template<class Alloc> void OSCL_HeapString< Alloc >::set (const other_chartype * buf, uint32 length, optype op)` [inherited]

Set the contents of this string to a new string or character array, with conversion operation.

Parameters:

- buf*: string or character array.
- length*: number of characters to copy.
- op*: conversion operation to apply

6.4.4.90 `template<class Alloc> void OSCL_wHeapString< Alloc >::set (const other_chartype * buf, optype op)` [inherited]

6.4.4.91 `template<class Alloc> void OSCL_HeapString< Alloc >::set (const other_chartype * buf, optype op)` [inherited]

Set the contents of this string to a new string, with conversion operation.

Parameters:

- buf*: NULL-terminated wide string.
op: conversion operation to apply

6.4.4.92 `template<class Alloc> void OSCL_wHeapString< Alloc >::set (const chartype * buf,
 uint32 length)` [inherited]

6.4.4.93 `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.

6.4.4.94 `OSCL_IMPORT_REF const char* skip_to_line_term (const char * start_ptr, const char *
end_ptr)`

6.4.4.95 `OSCL_IMPORT_REF const char* skip_to_whitespace (const char * start, const char *
end)`

6.4.4.96 `OSCL_IMPORT_REF const char* skip_whitespace (const char * start, const char * end)`

6.4.4.97 `OSCL_IMPORT_REF char* skip_whitespace (char * ptr)`

6.4.4.98 `OSCL_IMPORT_REF const char* skip_whitespace (const char * ptr)`

6.4.4.99 `OSCL_IMPORT_REF const char* skip_whitespace_and_line_term (const char * start,
 const char * end)`

6.4.4.100 `template<class Alloc> OSCL_HeapString< Alloc >::~OSCL_HeapString ()`
 [inherited]

6.4.4.101 `template<uint32 MaxBufSize> OSCL_StackString< MaxBufSize
 >::~OSCL_StackString ()` [inherited]

6.4.4.102 `template<class Alloc> OSCL_wHeapString< Alloc >::~OSCL_wHeapString ()`
 [inherited]

6.4.4.103 `template<uint32 MaxBufSize> OSCL_wStackString< MaxBufSize
 >::~OSCL_wStackString ()` [inherited]

6.4.5 Variable Documentation

6.4.5.1 `const int32 APPEND_MEDIA_AT_END = -1`

6.4.5.2 `const uint8 OSCL_ASCII_CASE_MAGIC_BIT = 0x20`

6.5 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 [_OscHeapBase](#)
- class [internalLeave](#)
- class [OscError](#)

- 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 [OscLTLRegistryEx](#)
- 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 [OscErrNone](#) 0
- #define [OscErrGeneral](#) 100
- #define [OscErrNoMemory](#) 101
- #define [OscErrCancelled](#) 102
- #define [OscErrNotSupported](#) 103
- #define [OscErrArgument](#) 104
- #define [OscErrBadHandle](#) 105
- #define [OscErrAlreadyExists](#) 106
- #define [OscErrBusy](#) 107
- #define [OscErrNotReady](#) 108
- #define [OscErrCorrupt](#) 109
- #define [OscErrTimeout](#) 110
- #define [OscErrOverflow](#) 111
- #define [OscErrUnderflow](#) 112
- #define [OscErrInvalidState](#) 113
- #define [OscErrNoResources](#) 114
- #define [OscErrNotInstalled](#) 115
- #define [OscErrAlreadyInstalled](#) 116
- #define [OscErrSystemCallFailed](#) 117
- #define [OscErrNoHandler](#) 118
- #define [OscErrThreadContextIncorrect](#) 119
- #define [OSCL_ERR_NONE](#) [OscErrNone](#)
- #define [OSCL_BAD_ALLOC_EXCEPTION_CODE](#) [OscErrNoMemory](#)
- #define [OscSuccess](#) 0
- #define [OscPending](#) 1
- #define [OscFailure](#) -1
- #define [PVEERROR_IMP_JUMPS](#)

- `#define PVErrDoLeave() 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 OscITLSRegistry`
- `#define OSCL_LEAVE(_leave_status) OscError::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!=OscErrNone) { _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!=OscErrNone && _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!=OscErrNone && _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!=OscErrNone){ _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!=OscErrNone){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 OscLeaveCode`
- `typedef int32 OscReturnCode`
- `typedef void(* OscTrapOperation)(OscAny *)`

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.

6.5.1 Define Documentation

6.5.1.1 #define _PV_TRAP(__r, __s)

Value:

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

6.5.1.2 #define _PV_TRAP_NO_TLS(__trapimp, __r, __s)

Value:

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

6.5.1.3 #define internalLeave (-1)

6.5.1.4 #define OSCL_BAD_ALLOC_EXCEPTION_CODE OscErrNoMemory

6.5.1.5 #define OSCL_CATCH(leave_status, catch_value, statements) else if (leave_status!=OscErrNone && 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

**6.5.1.6 #define OSCL_CATCH_ANY(_leave_status, _statements) else if
 (_leave_status!=OscErrNone){ _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.

6.5.1.7 #define OSCL_ERR_NONE OscErrNone

For backward compatibility with old definitions

**6.5.1.8 #define OSCL_FIRST_CATCH(_leave_status, _catch_value, _statements) if
 (_leave_status!=OscErrNone && _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

**6.5.1.9 #define OSCL_FIRST_CATCH_ANY(_leave_status, _statements) if
 (_leave_status!=OscErrNone) { _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

6.5.1.10 #define OSCL_JUMP_MAX_JUMP_MARKS OSCL_MAX_TRAP_LEVELS

6.5.1.11 #define OSCL_LAST_CATCH(_leave_status) else if (_leave_status!=OscErrNone){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.

6.5.1.12 #define OSCL_LEAVE(_leave_status) OsciError::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

6.5.1.13 #define OSCL_MAX_TRAP_LEVELS 20
6.5.1.14 #define OSCL_TRAPSTACK_POP() OsciError::Pop()
6.5.1.15 #define OSCL_TRAPSTACK_POPDEALLOC() OsciError::PopDealloc()
6.5.1.16 #define OSCL_TRAPSTACK_PUSH(a) OsciError::PushL(a)

Cleanup Stack user macros

6.5.1.17 #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

- 6.5.1.18 #define OSCL_TRY_NO_TLS(__trapimp, _leave_status, _statements)
_PV_TRAP_NO_TLS(__trapimp, _leave_status, _statements)
- 6.5.1.19 #define OsclErrAlreadyExists 106
- 6.5.1.20 #define OsclErrAlreadyInstalled 116
- 6.5.1.21 #define OsclErrArgument 104
- 6.5.1.22 #define OsclErrBadHandle 105
- 6.5.1.23 #define OsclErrBusy 107
- 6.5.1.24 #define OsclErrCancelled 102
- 6.5.1.25 #define OsclErrCorrupt 109
- 6.5.1.26 #define OsclErrGeneral 100
- 6.5.1.27 #define OsclErrInvalidState 113
- 6.5.1.28 #define OsclErrNoHandler 118
- 6.5.1.29 #define OsclErrNoMemory 101
- 6.5.1.30 #define OsclErrNone 0
- 6.5.1.31 #define OsclErrNoResources 114
- 6.5.1.32 #define OsclErrNotInstalled 115
- 6.5.1.33 #define OsclErrNotReady 108
- 6.5.1.34 #define OsclErrNotSupported 103
- 6.5.1.35 #define OsclErrOverflow 111
- 6.5.1.36 #define OsclErrSystemCallFailed 117
- 6.5.1.37 #define OsclErrThreadContextIncorrect 119
- 6.5.1.38 #define OsclErrTimeout 110
- 6.5.1.39 #define OsclErrUnderflow 112
- 6.5.1.40 #define OsclFailure -1
- 6.5.1.41 #define OsclPending 1
- 6.5.1.42 #define OsclSuccess 0
- 6.5.1.43 #define PVErrDoLeave() [internalLeave](#) __ilv; __ilv.a=0;throw(__ilv)
- 6.5.1.44 #define PVErrImpJumps

6.5.1.45 `#define PVErrorTrap_Registry` [OscTLsRegistry](#)

6.5.1.46 `#define PVErrorTrap_Registry_ID` [OSCL_TLS_ID_PVErrorTrap](#)

6.5.2 Typedef Documentation

6.5.2.1 `typedef int32` [OscLeaveCode](#)

Leave Codes

6.5.2.2 `typedef int32` [OscReturnCode](#)

Return Codes

6.5.2.3 `typedef void(*` [OscTrapOperation](#)`)(`[OscAny*](#)`)`

[OscTrapItem](#) may be used in the cleanup stack when a custom cleanup operation is needed.

6.5.3 Function Documentation

6.5.3.1 `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

6.5.3.2 `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

6.5.3.3 `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

6.5.3.4 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

6.6 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 [OscFileCache](#).
- 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 [Osc_FileFind](#).
- file [oscl_file_handle.h](#)
The file [oscl_file_handle.h](#) defines the class [OscFileHandle](#).
- file [oscl_file_io.h](#)
The file [oscl_file_io.h](#) defines the class [Osc_File](#). This is the public API to the basic file I/O operations.
- file [oscl_file_manager.h](#)
File management class.
- file [oscl_file_native.h](#)
The file [oscl_file_native.h](#) defines the class [OscNativeFile](#). 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 [Osc_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 [Osc_File](#)
- class [Osc_FileFind](#)
- class [Osc_FileServer](#)
- struct [oscl_fsstat](#)

- struct `oscl_stat_buf`
- class `OscIDNS`
- class `OscIDNSObserver`
- class `OscFileCache`
- class `OscFileCacheBuffer`
- class `OscFileHandle`
- class `OscFileManager`
- class `OscFileStats`
- class `OscFileStatsItem`
- class `OscNativeFile`
- class `OscNativeFileParams`
- class `OscSocketServ`
- class `OscTCPSocket`
- class `OscUDPSocket`

Defines

- #define `TOscFileOffsetInt32` `int32`
- #define `OSCL_FILE_STATS_LOGGER_NODE` `"OscFileStats"`
- #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 *` `TOscFileHandle`

Enumerations

- enum `TPVDNSFxN` { `EPVDNSGetHostByName` }
- enum `TPVDNSEvent` { `EPVDNSSuccess`, `EPVDNSPending`, `EPVDNSTimeout`, `EPVDNSFailure`, `EPVDNSCancel` }
- 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 `TOscFileOp` { `EOscFileOp_Open`, `EOscFileOp_Close`, `EOscFileOp_Read`, `EOscFileOp_Write`, `EOscFileOp_Seek`, `EOscFileOp_Tell`, `EOscFileOp_Size`, `EOscFileOp_Flush`, `EOscFileOp_EndOfFile`, `EOscFileOp_SetSize`, `EOscFileOp_NativeOpen`, `EOscFileOp_NativeClose`, `EOscFileOp_NativeRead`, `EOscFileOp_NativeWrite`, `EOscFileOp_NativeSeek`, `EOscFileOp_NativeTell`, `EOscFileOp_NativeSize`, `EOscFileOp_NativeFlush`, `EOscFileOp_NativeEndOfFile`, `EOscFileOp_NativeSetSize`, `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_statfs (OSCL_FSSTAT *stats, const char *path)
- OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_statfs (OSCL_FSSTAT *stats, const oscl_wchar *path)

6.6.1 Define Documentation

6.6.1.1 `#define OSCL_FILE_CHAR_PATH_DELIMITER _STRLIT_CHAR("/")`

6.6.1.2 `#define OSCL_FILE_STATS_LOGGER_NODE "OscFileStats"`

6.6.1.3 `#define OSCL_FILE_WCHAR_PATH_DELIMITER _STRLIT("/")`

6.6.1.4 `#define OSCL_IO_EXTENSION_MAXLEN 512`

6.6.1.5 `#define OSCL_IO_FILENAME_MAXLEN 512`

6.6.1.6 `#define TOscFileOffsetInt32 int32`

6.6.2 Typedef Documentation

6.6.2.1 `typedef struct oscl_fsstat OSCL_FSSTAT`

6.6.2.2 `typedef struct oscl_stat_buf OSCL_STAT_BUF`

6.6.2.3 `typedef FILE* TOscFileHandle`

TOscFileHandle 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.

6.6.3 Enumeration Type Documentation

6.6.3.1 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

6.6.3.2 enum OSCL_FILEMGMT_MODES

Enumeration values:

- OSCL_FILEMGMT_MODE_DIR

6.6.3.3 enum OSCL_FILEMGMT_PERMS

Enumeration values:

- OSCL_FILEMGMT_PERMS_READ
- OSCL_FILEMGMT_PERMS_WRITE
- OSCL_FILEMGMT_PERMS_EXECUTE

6.6.3.4 enum TOsclFileOp

Enumeration values:

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

EOsclFileOp_NativeRead
EOsclFileOp_NativeWrite
EOsclFileOp_NativeSeek
EOsclFileOp_NativeTell
EOsclFileOp_NativeSize
EOsclFileOp_NativeFlush
EOsclFileOp_NativeEndOfFile
EOsclFileOp_NativeSetSize
EOsclFileOp_Last

6.6.3.5 enum TPVDNSEvent

Enumeration values:

EPVDNSSuccess
EPVDNSPending
EPVDNSTimeout
EPVDNSFailure
EPVDNSCancel

6.6.3.6 enum TPVDNSFxn

Enumeration values:

EPVDNSGetHostByName

6.6.4 Function Documentation

6.6.4.1 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.

6.6.4.2 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.

6.6.4.3 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.

6.6.4.4 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.

6.6.4.5 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.

6.6.4.6 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.

6.6.4.7 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.

6.6.4.8 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.

6.6.4.9 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.

6.6.4.10 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.

6.6.4.11 OSL_IMPORT_REF OSL_FILEMGMT_ERR_TYPE oscl_stat (const char * path, OSL_STAT_BUF * statbuf)

oscl_stat function can be used to determine the attributes 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.

6.6.4.12 OSL_IMPORT_REF OSL_FILEMGMT_ERR_TYPE oscl_stat (const oscl_wchar * path, OSL_STAT_BUF * statbuf)

oscl_stat function can be used to determine the attributes 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.

6.6.4.13 OSL_IMPORT_REF OSL_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.

6.6.4.14 OSL_IMPORT_REF OSL_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.

6.7 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](#)
OscI Scheduler user execution object classes.
- file [oscl_scheduler_aobase.h](#)
OscI Scheduler internal active object classes.
- file [oscl_scheduler_readyq.h](#)
ready q types for oscI scheduler
- file [oscl_scheduler_threadcontext.h](#)
Thread context functions needed by oscI scheduler.
- file [oscl_scheduler_tuneables.h](#)
Tuneable settings for OscI Scheduler.
- file [oscl_scheduler_types.h](#)
Scheduler common types include file.

Data Structures

- class [OscIActiveObject](#)
- class [OscIAOStatus](#)
- class [OscIDoubleLink](#)
- class [OscIDoubleList](#)
- class [OscIDoubleListBase](#)
- class [OscIDoubleRunner](#)
- class [OscIExecScheduler](#)
- class [OscIExecSchedulerBase](#)
- class [OscIExecSchedulerCommonBase](#)
- class [OscIPriorityLink](#)
- class [OscIPriorityList](#)
- class [OscIReadyAlloc](#)
- class [OscIReadyCompare](#)
- class [OscIReadyQ](#)
- class [OscIScheduler](#)
- class [OscISchedulerObserver](#)
- class [OscITimerCompare](#)
- class [OscITimerObject](#)

- class `OscTimerQ`
- 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 * TOscReady`

Enumerations

- enum `TPVThreadContext { EPVThreadContext_InThread, EPVThreadContext_OscThread, EPVThreadContext_NonOscThread, EPVThreadContext_Undetermined }`

Functions

- template<class T, class S> T * `OscPtrAdd` (T *aPtr, S aVal)
- template<class T, class S> T * `OscPtrSub` (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)`

6.7.1 Define Documentation

6.7.1.1 #define OSCL_PERF_SUMMARY_LOGGING 0

6.7.1.2 #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.

6.7.1.3 #define PV_SCHED_CHECK_Q 0

6.7.1.4 #define PV_SCHED_ENABLE_AO_STATS 1

6.7.1.5 #define PV_SCHED_ENABLE_LOOP_STATS 0

6.7.1.6 #define PV_SCHED_ENABLE_PERF_LOGGING 1

6.7.1.7 #define PV_SCHED_ENABLE_THREAD_CONTEXT_CHECKS 1

6.7.1.8 #define PV_SCHED_FAIR_SCHEDULING 1

6.7.1.9 #define PV_SCHED_LOG_Q 0

6.7.1.10 #define PVEXECNAMELEN 30

6.7.1.11 #define PVSCHEDNAMELEN 30

PV Scheduler class

6.7.1.12 #define QUE_ITER_BEGIN(_type, _qname)

Value:

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

6.7.1.13 #define QUE_ITER_END(_qname)

Value:

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

6.7.2 Typedef Documentation

6.7.2.1 typedef **PVActiveBase*** TOsclReady

6.7.3 Enumeration Type Documentation

6.7.3.1 enum TPVThreadContext

Thread context type

Enumeration values:

EPVThreadContext_InThread

EPVThreadContext_OsclThread

EPVThreadContext_NonOsclThread

EPVThreadContext_Undetermined

6.7.4 Function Documentation

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

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

6.7.5 Variable Documentation

6.7.5.1 `const int32 OSCL_REQUEST_ERR_CANCEL = (-1)`

6.7.5.2 `const int32 OSCL_REQUEST_ERR_GENERAL = (-2)`

6.7.5.3 `const int32 OSCL_REQUEST_ERR_NONE = 0`

6.7.5.4 `const int32 OSCL_REQUEST_PENDING = (-0x7fffffff)`

6.8 OSCL Init

Files

- file [oscl_init.h](#)
Global oscl initialization.

Data Structures

- class [OscInit](#)
- class [OscSelect](#)

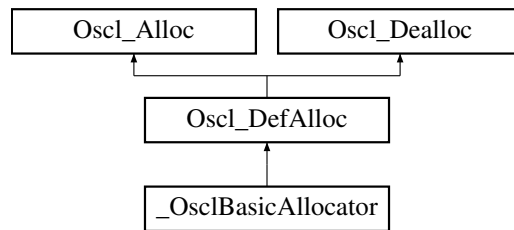
Chapter 7

oscl Data Structure Documentation

7.1 `_OscBasicAllocator` Class Reference

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

Inheritance diagram for `_OscBasicAllocator`:



Public Methods

- `OscAny * allocate (const uint32 size)`
- `void deallocate (OscAny *p)`
- `virtual ~_OscBasicAllocator ()`

7.1.1 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 `Osc` code only. Higher level code should use `OscMemAllocator` defined in "`oscl_mem.h`".

7.1.2 Constructor & Destructor Documentation

7.1.2.1 `virtual _OsciBasicAllocator::~_OsciBasicAllocator () [inline, virtual]`

7.1.3 Member Function Documentation

7.1.3.1 `OsciAny* _OsciBasicAllocator::allocate (const uint32 size) [inline, virtual]`

Implements `Osci_DefAlloc`.

7.1.3.2 `void _OsciBasicAllocator::deallocate (OsciAny *p) [inline, virtual]`

Implements `Osci_DefAlloc`.

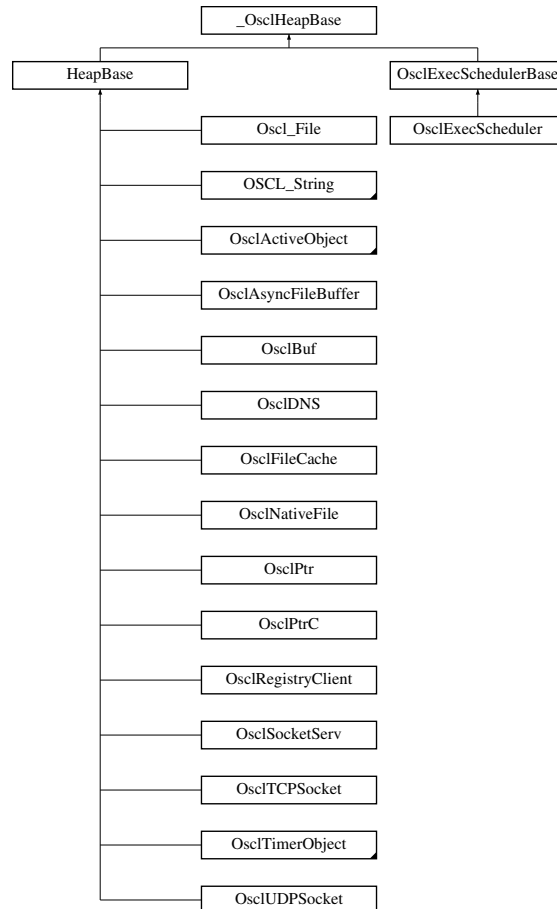
The documentation for this class was generated from the following file:

- [oscl_base_alloc.h](#)

7.2 `_OscHeapBase` Class Reference

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

Inheritance diagram for `_OscHeapBase`:



Public Methods

- virtual `~_OscHeapBase ()`

Protected Methods

- `_OscHeapBase ()`
- `_OscHeapBase (const _OscHeapBase &)`

Friends

- class `PVCleanupStack`

7.2.1 Detailed Description

`_OsciHeapBase` is used as the base for cleanup stack items with virtual destructor.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 `virtual _OsciHeapBase::~~_OsciHeapBase()` [inline, virtual]

7.2.2.2 `_OsciHeapBase::_OsciHeapBase()` [inline, protected]

7.2.2.3 `_OsciHeapBase::_OsciHeapBase(const _OsciHeapBase &)` [inline, protected]

7.2.3 Friends And Related Function Documentation

7.2.3.1 `friend class PVCleanupStack` [friend]

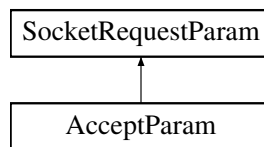
The documentation for this class was generated from the following file:

- [oscl_heapbase.h](#)

7.3 AcceptParam Class Reference

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

Inheritance diagram for AcceptParam::



Public Methods

- [AcceptParam](#) ([OscSocketI](#) &[aBlankSocket](#))

Data Fields

- [OscSocketI](#) * [iBlankSocket](#)

7.3.1 Constructor & Destructor Documentation

7.3.1.1 [AcceptParam::AcceptParam](#) ([OscSocketI](#) & [aBlankSocket](#)) [[inline](#)]

7.3.2 Field Documentation

7.3.2.1 [OscSocketI](#)* [AcceptParam::iBlankSocket](#)

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

- [oscl_socket_request.h](#)

7.4 allocator Class Reference

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

7.4.1 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.

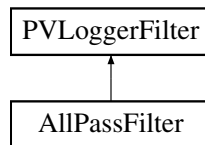
The documentation for this class was generated from the following file:

- [oscl_mem_mempool.h](#)

7.5 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 FilterOpaqueMessage](#) (char *tag, [message_id_type](#) msgID, [log_level_type](#) level)

7.5.1 Detailed Description

Example filter that allows all messages to be logged.

7.5.2 Member Typedef Documentation

7.5.2.1 typedef PVLoggerFilter::filter_status_type AllPassFilter::filter_status_type

Reimplemented from [PVLoggerFilter](#).

7.5.2.2 typedef PVLoggerFilter::log_level_type AllPassFilter::log_level_type

Reimplemented from [PVLoggerFilter](#).

7.5.2.3 typedef PVLoggerFilter::message_id_type AllPassFilter::message_id_type

Reimplemented from [PVLoggerFilter](#).

7.5.3 Constructor & Destructor Documentation

7.5.3.1 AllPassFilter::AllPassFilter () [inline]

7.5.3.2 virtual AllPassFilter::~~AllPassFilter () [inline, virtual]

7.5.4 Member Function Documentation

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

Implements [PVLoggerFilter](#).

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

Implements [PVLoggerFilter](#).

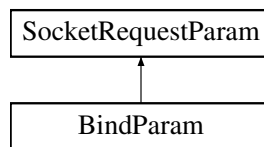
The documentation for this class was generated from the following file:

- [pvlogger_accessories.h](#)

7.6 BindParam Class Reference

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

Inheritance diagram for BindParam::



Public Methods

- [BindParam](#) ([OscNetworkAddress](#) &anAddr)

Data Fields

- [OscNetworkAddress](#) iAddr

7.6.1 Constructor & Destructor Documentation

7.6.1.1 [BindParam::BindParam](#) ([OscNetworkAddress](#) & *anAddr*) [inline]

7.6.2 Field Documentation

7.6.2.1 [OscNetworkAddress](#) [BindParam::iAddr](#)

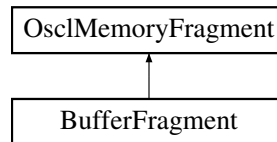
The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

7.7 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](#)

7.8 BufferMgr Class Reference

```
#include <oscl_media_data.h>
```

Public Methods

- virtual void [BufferReleased](#) (void *ptr, [BufferState](#) *state=NULL)=0
- virtual [~BufferMgr](#) ()

7.8.1 Constructor & Destructor Documentation

7.8.1.1 virtual [BufferMgr::~BufferMgr](#) () [inline, virtual]

7.8.2 Member Function Documentation

7.8.2.1 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](#)

7.9 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 [get_refcount](#) ()
- [BufferFreeFuncPtr](#) [get_free_function](#) ()
- [BufferMgr](#) * [get_buf_mgr](#) ()
- void [reset](#) ()

7.9.1 Constructor & Destructor Documentation

7.9.1.1 [BufferState::BufferState](#) ([BufferFreeFuncPtr](#) the_free_function, void * bufptr = 0) [inline]

7.9.1.2 [BufferState::BufferState](#) ([BufferMgr](#) * the_buf_mgr = 0, void * bufptr = 0) [inline]

7.9.2 Member Function Documentation

7.9.2.1 void [BufferState::bind](#) (void * in_ptr, [BufferMgr](#) * in_buf_mgr) [inline]

7.9.2.2 void [BufferState::bind](#) (void * in_ptr, [BufferFreeFuncPtr](#) in_free_function) [inline]

7.9.2.3 void [BufferState::decrement_refcnt](#) () [inline]

7.9.2.4 [BufferMgr](#)* [BufferState::get_buf_mgr](#) () [inline]

7.9.2.5 [BufferFreeFuncPtr](#) [BufferState::get_free_function](#) () [inline]

7.9.2.6 void* [BufferState::get_ptr](#) () [inline]

7.9.2.7 int32 [BufferState::get_refcount](#) () [inline]

7.9.2.8 void [BufferState::increment_refcnt](#) () [inline]

7.9.2.9 void [BufferState::reset](#) () [inline]

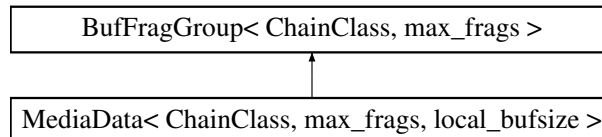
The documentation for this class was generated from the following file:

- [oscl_media_data.h](#)

7.10 BufFragGroup< ChainClass, max_frgs > Class Template Reference

```
#include <oscl_media_data.h>
```

Inheritance diagram for BufFragGroup< ChainClass, max_frgs >::



Public Methods

- [BufFragGroup](#) ()
- virtual [~BufFragGroup](#) ()
- int32 [GetMaxFrgs](#) () const
- int32 [GetNumFrgs](#) () 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_frgs)

Protected Attributes

- [BufferFragment](#) [fragments](#) [max_frgs]
- [BufferState](#) * [buffer_states](#) [max_frgs]
- ChainClass * [next](#)
- uint32 [num_fragments](#)
- uint32 [length](#)

```
template<class ChainClass, uint32 max_fragments> class BuffragGroup< ChainClass, max_fragments >
```

7.10.1 Constructor & Destructor Documentation

7.10.1.1 `template<class ChainClass, uint32 max_fragments> BuffragGroup< ChainClass, max_fragments >::BuffragGroup () [inline]`

7.10.1.2 `template<class ChainClass, uint32 max_fragments> virtual BuffragGroup< ChainClass, max_fragments >::~~BuffragGroup () [inline, virtual]`

7.10.2 Member Function Documentation

7.10.2.1 `template<class ChainClass, uint32 max_fragments> BuffragStatusClass::status_t BuffragGroup< ChainClass, max_fragments >::AddFragment (const BufferFragment & frag, BufferState * in_buffer_state, int32 location_offset = max_fragments) [inline, protected]`

7.10.2.2 `template<class ChainClass, uint32 max_fragments> void BuffragGroup< ChainClass, max_fragments >::AppendNext (ChainClass * next_ptr) [inline]`

7.10.2.3 `template<class ChainClass, uint32 max_fragments> virtual void BuffragGroup< ChainClass, max_fragments >::Clear () [inline, protected, virtual]`

Reimplemented in [MediaData< ChainClass, max_fragments, local_bufsize >](#).

- 7.10.2.4 `template<class ChainClass, uint32 max_frgs> uint32 BuffragGroup< ChainClass, max_frgs >::GetLength () const [inline]`
- 7.10.2.5 `template<class ChainClass, uint32 max_frgs> int32 BuffragGroup< ChainClass, max_frgs >::GetMaxFrgs () const [inline]`
- 7.10.2.6 `template<class ChainClass, uint32 max_frgs> ChainClass* BuffragGroup< ChainClass, max_frgs >::GetNext () const [inline]`
- 7.10.2.7 `template<class ChainClass, uint32 max_frgs> int32 BuffragGroup< ChainClass, max_frgs >::GetNumFrgs () const [inline]`

7.10.3 Field Documentation

- 7.10.3.1 `template<class ChainClass, uint32 max_frgs> BufferState* BuffragGroup< ChainClass, max_frgs >::buffer_states[max_frgs] [protected]`
- 7.10.3.2 `template<class ChainClass, uint32 max_frgs> BufferFragment BuffragGroup< ChainClass, max_frgs >::fragments[max_frgs] [protected]`
- 7.10.3.3 `template<class ChainClass, uint32 max_frgs> uint32 BuffragGroup< ChainClass, max_frgs >::length [protected]`
- 7.10.3.4 `template<class ChainClass, uint32 max_frgs> ChainClass* BuffragGroup< ChainClass, max_frgs >::next [protected]`
- 7.10.3.5 `template<class ChainClass, uint32 max_frgs> uint32 BuffragGroup< ChainClass, max_frgs >::num_fragments [protected]`

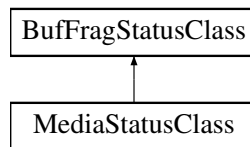
The documentation for this class was generated from the following file:

- [oscl_media_data.h](#)

7.11 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` }

7.11.1 Member Enumeration Documentation

7.11.1.1 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

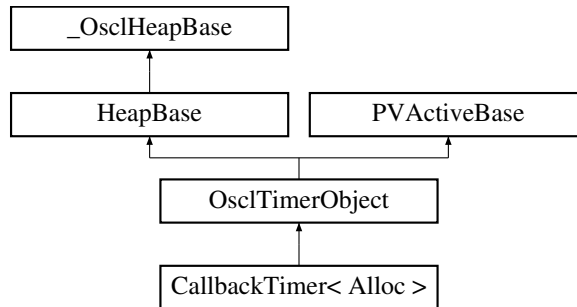
The documentation for this class was generated from the following file:

- [oscl_media_status.h](#)

7.12 CallbackTimer< Alloc > Class Template Reference

```
#include <oscl_timer.h>
```

Inheritance diagram for CallbackTimer< Alloc >::



Public Methods

- [CallbackTimer](#) ([CallbackTimerObserver](#) &aContainer, const char *name, int32 aPriority=OsciActiveObject::EPriorityNominal)
- [~CallbackTimer](#) ()
- void [Run](#) ()

```
template<class Alloc> class CallbackTimer< Alloc >
```

7.12.1 Constructor & Destructor Documentation

7.12.1.1 `template<class Alloc> CallbackTimer< Alloc >::CallbackTimer`
 ([CallbackTimerObserver](#) &aContainer, const char * name, int32 aPriority =
 OsciActiveObject::EPriorityNominal) [inline]

7.12.1.2 `template<class Alloc> CallbackTimer< Alloc >::~~CallbackTimer` () [inline]

7.12.2 Member Function Documentation

7.12.2.1 `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](#).

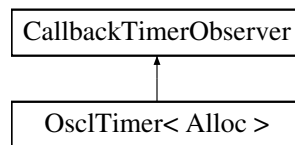
The documentation for this class was generated from the following file:

- [oscl_timer.h](#)

7.13 CallbackTimerObserver Class Reference

```
#include <oscl_timer.h>
```

Inheritance diagram for CallbackTimerObserver::



Public Methods

- virtual void [TimerBaseElapsed](#) ()=0
- virtual [~CallbackTimerObserver](#) ()

7.13.1 Constructor & Destructor Documentation

7.13.1.1 virtual [CallbackTimerObserver::~CallbackTimerObserver](#) () [inline, virtual]

7.13.2 Member Function Documentation

7.13.2.1 virtual void [CallbackTimerObserver::TimerBaseElapsed](#) () [pure virtual]

Implemented in [OsciTimer< Alloc >](#).

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

- [oscl_timer.h](#)

7.14 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](#)
- bool [overwrite](#)

7.14.1 Detailed Description

For internal use only– fast string representation

7.14.2 Constructor & Destructor Documentation

7.14.2.1 CFastRep::CFastRep() [inline]

7.14.3 Member Function Documentation

7.14.3.1 OSCL_IMPORT_REF void CFastRep::append (const [oscl_wchar](#) * cp, uint32 len)

7.14.3.2 OSCL_IMPORT_REF void CFastRep::append (const char * cp, uint32 len)

7.14.3.3 OSCL_IMPORT_REF void CFastRep::set_r (const [oscl_wchar](#) * cp, uint32 len)

7.14.3.4 OSCL_IMPORT_REF void CFastRep::set_r (const char * cp, uint32 len)

7.14.3.5 OSCL_IMPORT_REF void CFastRep::set_w ([oscl_wchar](#) * cp, uint32 len, uint32 maxlen)

7.14.3.6 OSCL_IMPORT_REF void CFastRep::set_w (char * cp, uint32 len, uint32 maxlen)

7.14.4 Field Documentation

7.14.4.1 [OsclAny](#)* CFastRep::buffer

7.14.4.2 uint32 CFastRep::maxsize

7.14.4.3 bool CFastRep::overwrite

7.14.4.4 uint32 CFastRep::size

7.14.4.5 bool CFastRep::writable

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

- [oscl_string_rep.h](#)

7.15 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](#) (CHeapRep *&, [OscL_DefAlloc](#) &, const char *, uint32)
- OSCL_IMPORT_REF void [set_rep](#) (CHeapRep *&, [OscL_DefAlloc](#) &, const [oscl_wchar](#) *, uint32)
- OSCL_IMPORT_REF void [append_rep](#) (CHeapRep *&, [OscL_DefAlloc](#) &, const char *, uint32)
- OSCL_IMPORT_REF void [append_rep](#) (CHeapRep *&, [OscL_DefAlloc](#) &, const [oscl_wchar](#) *, uint32)
- OSCL_IMPORT_REF void [assign](#) (CHeapRep *&, CHeapRep *, [OscL_DefAlloc](#) &)

Data Fields

- uint32 [refcount](#)
- [OscLAny](#) * [buffer](#)
- uint32 [maxsize](#)
- uint32 [size](#)

7.15.1 Detailed Description

For internal use only– heap string representation

7.15.2 Constructor & Destructor Documentation

7.15.2.1 CHeapRep::CHeapRep () [inline]

7.15.3 Member Function Documentation

7.15.3.1 OSCL_IMPORT_REF void CHeapRep::add_ref ()

7.15.3.2 OSCL_IMPORT_REF bool CHeapRep::append (uint32, const [oscl_wchar](#) *, uint32, const [oscl_wchar](#) *, [Oscl_DefAlloc](#) &)

7.15.3.3 OSCL_IMPORT_REF bool CHeapRep::append (uint32, const char *, uint32, const char *, [Oscl_DefAlloc](#) &)

7.15.3.4 OSCL_IMPORT_REF void CHeapRep::append_rep (CHeapRep * &, [Oscl_DefAlloc](#) &, const [oscl_wchar](#) *, uint32) [static]

7.15.3.5 OSCL_IMPORT_REF void CHeapRep::append_rep (CHeapRep * &, [Oscl_DefAlloc](#) &, const char *, uint32) [static]

7.15.3.6 OSCL_IMPORT_REF void CHeapRep::assign (CHeapRep * &, CHeapRep *, [Oscl_DefAlloc](#) &) [static]

7.15.3.7 OSCL_IMPORT_REF void CHeapRep::remove_ref ([Oscl_DefAlloc](#) &)

7.15.3.8 OSCL_IMPORT_REF bool CHeapRep::set (uint32, const [oscl_wchar](#) *, [Oscl_DefAlloc](#) &)

7.15.3.9 OSCL_IMPORT_REF bool CHeapRep::set (uint32, const char *, [Oscl_DefAlloc](#) &)

7.15.3.10 OSCL_IMPORT_REF void CHeapRep::set_rep (CHeapRep * &, [Oscl_DefAlloc](#) &, const [oscl_wchar](#) *, uint32) [static]

7.15.3.11 OSCL_IMPORT_REF void CHeapRep::set_rep (CHeapRep * &, [Oscl_DefAlloc](#) &, const char *, uint32) [static]

7.15.4 Field Documentation

7.15.4.1 [OsclAny](#)* CHeapRep::buffer

7.15.4.2 uint32 CHeapRep::maxsize

7.15.4.3 uint32 CHeapRep::refcount

7.15.4.4 uint32 CHeapRep::size

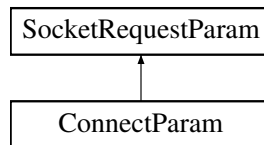
The documentation for this class was generated from the following file:

- [oscl_string_rep.h](#)

7.16 ConnectParam Class Reference

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

Inheritance diagram for ConnectParam::



Public Methods

- [ConnectParam](#) ([OscNetworkAddress](#) &anAddr)

Data Fields

- [OscNetworkAddress](#) iAddr

7.16.1 Constructor & Destructor Documentation

7.16.1.1 [ConnectParam::ConnectParam](#) ([OscNetworkAddress](#) & *anAddr*) [inline]

7.16.2 Field Documentation

7.16.2.1 [OscNetworkAddress](#) [ConnectParam::iAddr](#)

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

- [oscl_socket_request.h](#)

7.17 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](#)

7.17.1 Detailed Description

For internal use only– stack string representation

7.17.2 Constructor & Destructor Documentation

7.17.2.1 [CStackRep::CStackRep \(\)](#) [inline]

7.17.3 Member Function Documentation

7.17.3.1 OSCL_IMPORT_REF void [CStackRep::append](#) (const [oscl_wchar](#) * cp, uint32 len)

7.17.3.2 OSCL_IMPORT_REF void [CStackRep::append](#) (const char * cp, uint32 len)

7.17.3.3 OSCL_IMPORT_REF void [CStackRep::set](#) (const [oscl_wchar](#) * cp, uint32 len)

7.17.3.4 OSCL_IMPORT_REF void [CStackRep::set](#) (const char * cp, uint32 len)

7.17.4 Field Documentation

7.17.4.1 [OsclAny](#)* [CStackRep::buffer](#)

7.17.4.2 uint32 [CStackRep::maxsize](#)

7.17.4.3 uint32 [CStackRep::size](#)

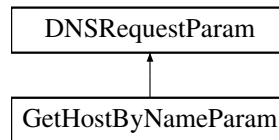
The documentation for this class was generated from the following file:

- [oscl_string_rep.h](#)

7.18 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

- [TPVDNSFxn iFxn](#)
- [OscIDNSRequest * iDNSRequest](#)

Protected Methods

- [DNSRequestParam \(TPVDNSFxn aFxn\)](#)

Protected Attributes

- uint32 [iRefCount](#)

7.18.1 Constructor & Destructor Documentation

7.18.1.1 virtual [DNSRequestParam::~~DNSRequestParam \(\)](#) [inline, virtual]

7.18.1.2 [DNSRequestParam::DNSRequestParam \(TPVDNSFxn aFxn\)](#) [protected]

7.18.2 Member Function Documentation

7.18.2.1 virtual void [DNSRequestParam::Destroy \(\)](#) [pure virtual]

Implemented in [GetHostByNameParam](#).

7.18.2.2 void DNSRequestParam::InThread ()

7.18.2.3 void DNSRequestParam::RemoveRef ()

7.18.3 Field Documentation

7.18.3.1 [OsciDNSRequest*](#) DNSRequestParam::iDNSRequest

7.18.3.2 [TPVDNSFxn](#) DNSRequestParam::iFxn

7.18.3.3 uint32 DNSRequestParam::iRefCount [protected]

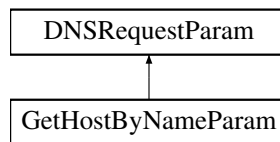
The documentation for this class was generated from the following file:

- [oscl_dns_param.h](#)

7.19 GetHostByNameParam Class Reference

```
#include <oscl_dns_param.h>
```

Inheritance diagram for GetHostByNameParam::



Public Types

- enum { `addressListCapacity` = 10 }

Public Methods

- void `Destroy` ()
- `~GetHostByNameParam` ()
- void `PersistHostAddress` (const `OscNetworkAddress` &addr)
- bool `canPersistMoreHostAddresses` ()

Static Public Methods

- `GetHostByNameParam * Create` (const char *name, `OscNetworkAddress` *&addr, `Osc_Vector`<`OscNetworkAddress`, `OscMemAllocator`> *aAddressList)

Data Fields

- char * `iName`
- `OscNetworkAddress` * `iAddr`
- `Osc_Vector`<`OscNetworkAddress`, `OscMemAllocator`> * `iAddressList`

7.19.1 Member Enumeration Documentation

7.19.1.1 anonymous enum

Enumeration values:

`addressListCapacity`

7.19.2 Constructor & Destructor Documentation

7.19.2.1 `GetHostByNameParam::~GetHostByNameParam ()`

7.19.3 Member Function Documentation

7.19.3.1 `bool GetHostByNameParam::canPersistMoreHostAddresses ()` [inline]

7.19.3.2 `GetHostByNameParam* GetHostByNameParam::Create (const char * name, OsciNetworkAddress *& addr, Osci_Vector< OsciNetworkAddress, OsciMemAllocator > * aAddressList)` [static]

7.19.3.3 `void GetHostByNameParam::Destroy ()` [virtual]

Implements [DNSRequestParam](#).

7.19.3.4 `void GetHostByNameParam::PersistHostAddress (const OsciNetworkAddress & addr)` [inline]

7.19.4 Field Documentation

7.19.4.1 `OsciNetworkAddress* GetHostByNameParam::iAddr`

7.19.4.2 `Osci_Vector<OsciNetworkAddress, OsciMemAllocator>* GetHostByNameParam::i-AddressList`

7.19.4.3 `char* GetHostByNameParam::iName`

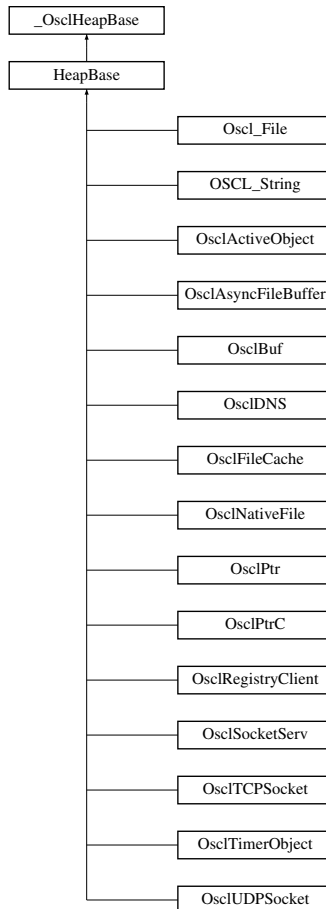
The documentation for this class was generated from the following file:

- [oscl_dns_param.h](#)

7.20 HeapBase Class Reference

```
#include <oscl_mem.h>
```

Inheritance diagram for HeapBase::



Public Methods

- [HeapBase \(\)](#)
- virtual [~HeapBase \(\)](#)

7.20.1 Detailed Description

HeapBase is the base class for all classes that allocates memory.

HeapBase has overloaded new and delete operators.

Derived from [_OscHeapBase](#) 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.

7.20.2 Constructor & Destructor Documentation

7.20.2.1 HeapBase::HeapBase () [inline]

7.20.2.2 virtual HeapBase::~HeapBase () [inline, virtual]

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

- [oscl_mem.h](#)

7.21 internalLeave Class Reference

```
#include <oscl_error_imp_cppexceptions.h>
```

Data Fields

- [int a](#)

7.21.1 Field Documentation

7.21.1.1 `int internalLeave::a`

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

- [oscl_error_imp_cppexceptions.h](#)

7.22 LinkedListElement< LLClass > Class Template Reference

```
#include <oscl_linked_list.h>
```

Public Methods

- [LinkedListElement](#) (LLClass in_data)

Data Fields

- LinkedListElement< LLClass > * [next](#)
- LLClass [data](#)

7.22.1 Detailed Description

```
template<class LLClass> class LinkedListElement< LLClass >
```

Linked List Element Class

7.22.2 Constructor & Destructor Documentation

7.22.2.1 `template<class LLClass> LinkedListElement< LLClass >::LinkedListElement (LLClass in_data) [inline]`

7.22.3 Field Documentation

7.22.3.1 `template<class LLClass> LLClass LinkedListElement< LLClass >::data`

7.22.3.2 `template<class LLClass> LinkedListElement<LLClass>* LinkedListElement< LLClass >::next`

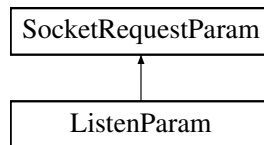
The documentation for this class was generated from the following file:

- [oscl_linked_list.h](#)

7.23 ListenParam Class Reference

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

Inheritance diagram for ListenParam::



Public Methods

- [ListenParam](#) (uint32 aSize)

Data Fields

- uint32 [iQSize](#)

7.23.1 Constructor & Destructor Documentation

7.23.1.1 [ListenParam::ListenParam](#) (uint32 aSize) [inline]

7.23.2 Field Documentation

7.23.2.1 uint32 [ListenParam::iQSize](#)

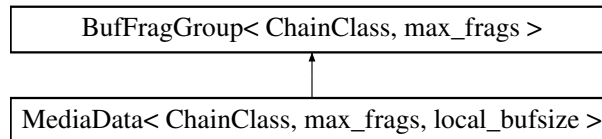
The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

7.24 `MediaData< ChainClass, max_frgs, local_bufsize >` Class Template Reference

```
#include <oscl_media_data.h>
```

Inheritance diagram for `MediaData< ChainClass, max_frgs, 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 GetNumMediaFrgs (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_fragments, uint32 local_bufsize> class MediaData< ChainClass,
max_fragments, local_bufsize >
```

7.24.1 Constructor & Destructor Documentation

7.24.1.1 `template<class ChainClass, uint32 max_fragments, uint32 local_bufsize> MediaData< ChainClass, max_fragments, local_bufsize >::MediaData () [inline]`

7.24.1.2 `template<class ChainClass, uint32 max_fragments, uint32 local_bufsize> virtual MediaData< ChainClass, max_fragments, local_bufsize >::~MediaData () [inline, virtual]`

7.24.2 Member Function Documentation

7.24.2.1 `template<class ChainClass, uint32 max_fragments, uint32 local_bufsize> MediaStatusClass::status_t MediaData< ChainClass, max_fragments, local_bufsize >::AddLocalFragment (const BufferFragment & frag, int32 location_offset) [inline, protected]`

7.24.2.2 `template<class ChainClass, uint32 max_fragments, uint32 local_bufsize> virtual void MediaData< ChainClass, max_fragments, local_bufsize >::Clear () [inline, virtual]`

Reimplemented from `BuffFragGroup< ChainClass, max_fragments >`.

- 7.24.2.3 `template<class ChainClass, uint32 max_frags, uint32 local_bufsize> uint32 MediaData< ChainClass, max_frags, local_bufsize >::GetAvailableBufferSize () const` [inline]
- 7.24.2.4 `template<class ChainClass, uint32 max_frags, uint32 local_bufsize> uint32 MediaData< ChainClass, max_frags, local_bufsize >::GetLocalBufsize () const` [inline]
- 7.24.2.5 `template<class ChainClass, uint32 max_frags, uint32 local_bufsize> MediaStatusClass::status_t MediaData< ChainClass, max_frags, local_bufsize >::GetLocalFragment (BufferFragment & fragment)` [inline]
- 7.24.2.6 `template<class ChainClass, uint32 max_frags, uint32 local_bufsize> BufferFragment* MediaData< ChainClass, max_frags, local_bufsize >::GetMediaFragment (const uint32 idx)` [inline]
- 7.24.2.7 `template<class ChainClass, uint32 max_frags, uint32 local_bufsize> int MediaData< ChainClass, max_frags, local_bufsize >::GetMediaSize () const` [inline]
- 7.24.2.8 `template<class ChainClass, uint32 max_frags, uint32 local_bufsize> uint32 MediaData< ChainClass, max_frags, local_bufsize >::GetNumMediaFrag (const uint32 idx) const` [inline]
- 7.24.2.9 `template<class ChainClass, uint32 max_frags, uint32 local_bufsize> MediaTimestamp MediaData< ChainClass, max_frags, local_bufsize >::GetTimestamp () const` [inline]
- 7.24.2.10 `template<class ChainClass, uint32 max_frags, uint32 local_bufsize> bool MediaData< ChainClass, max_frags, local_bufsize >::IsLocalData (const OscMemoryFragment & frag) const` [inline]
- 7.24.2.11 `template<class ChainClass, uint32 max_frags, uint32 local_bufsize> void MediaData< ChainClass, max_frags, local_bufsize >::SetTimestamp (MediaTimestamp in_timestamp)` [inline]

7.24.3 Field Documentation

- 7.24.3.1 `template<class ChainClass, uint32 max_frags, uint32 local_bufsize> uint32 MediaData< ChainClass, max_frags, local_bufsize >::available_localbuf` [protected]
- 7.24.3.2 `template<class ChainClass, uint32 max_frags, uint32 local_bufsize> uint8 MediaData< ChainClass, max_frags, local_bufsize >::localbuf[local_bufsize]` [protected]
- 7.24.3.3 `template<class ChainClass, uint32 max_frags, uint32 local_bufsize> int MediaData< ChainClass, max_frags, local_bufsize >::num_reserved_fragments` [protected]
- 7.24.3.4 `template<class ChainClass, uint32 max_frags, uint32 local_bufsize> MediaTimestamp MediaData< ChainClass, max_frags, local_bufsize >::timestamp` [protected]

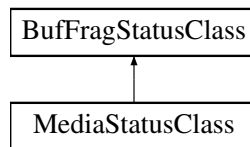
The documentation for this class was generated from the following file:

- [oscl_media_data.h](#)

7.25 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](#)

7.26 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 >
```

7.26.1 Member Typedef Documentation

7.26.1.1 `template<class T> typedef T* MemAllocator< T >::pointer`

7.26.2 Constructor & Destructor Documentation

7.26.2.1 `template<class T> virtual MemAllocator< T >::~~MemAllocator ()` [inline, virtual]

7.26.3 Member Function Documentation

7.26.3.1 `template<class T> virtual pointer MemAllocator< T >::allocate (void * hint = 0, const int num_reserved_frags = 1)` [pure virtual]

7.26.3.2 `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](#)

7.27 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)]

7.27.1 Constructor & Destructor Documentation

7.27.1.1 [MM_AllocBlockFence::MM_AllocBlockFence \(\)](#) [inline]

7.27.2 Member Function Documentation

7.27.2.1 [bool MM_AllocBlockFence::check_fence \(\)](#) [inline]

7.27.2.2 [void MM_AllocBlockFence::fill_fence \(\)](#) [inline]

7.27.3 Field Documentation

7.27.3.1 [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](#)

7.28 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

7.28.1 Constructor & Destructor Documentation

7.28.1.1 [MM_AllocBlockHdr::MM_AllocBlockHdr](#) () [inline]

7.28.1.2 [MM_AllocBlockHdr::MM_AllocBlockHdr](#) (void *ptr, uint32 inSize) [inline]

7.28.2 Member Function Documentation

7.28.2.1 [bool MM_AllocBlockHdr::isAllocNodePtr](#) () [inline]

7.28.2.2 [void MM_AllocBlockHdr::setAllocNodeFlag](#) () [inline]

7.28.3 Field Documentation

7.28.3.1 [uint32 MM_AllocBlockHdr::pad](#)

7.28.3.2 [void* MM_AllocBlockHdr::pNode](#)

7.28.3.3 [void* MM_AllocBlockHdr::pRootNode](#)

7.28.3.4 [uint32 MM_AllocBlockHdr::size](#)

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

- [oscl_mem_audit_internals.h](#)

7.29 MM_AllocInfo Struct Reference

```
#include <oscl_mem_audit.h>
```

Public Methods

- [MM_AllocInfo \(\)](#)
- [~MM_AllocInfo \(\)](#)
- void * [operator new](#) ([oscl_memsized_t size](#))
- void * [operator new](#) ([oscl_memsized_t size](#), [MM_AllocInfo *ptr](#))
- void [operator delete](#) (void *ptr) throw ()

Data Fields

- uint32 [allocNum](#)
- char * [pFileName](#)
- uint32 [lineNo](#)
- uint32 [size](#)
- void * [pMemBlock](#)
- [OscMemStatsNode](#) * [pStatsNode](#)
- bool [bSetFailure](#)

7.29.1 Constructor & Destructor Documentation

7.29.1.1 `MM_AllocInfo::MM_AllocInfo ()` [inline]

7.29.1.2 `MM_AllocInfo::~~MM_AllocInfo ()` [inline]

7.29.2 Member Function Documentation

7.29.2.1 `void MM_AllocInfo::operator delete (void * ptr) throw ()` [inline]

7.29.2.2 `void* MM_AllocInfo::operator new (oscl_memsize_t size, MM_AllocInfo * ptr)`
[inline]

7.29.2.3 `void* MM_AllocInfo::operator new (oscl_memsize_t size)` [inline]

7.29.3 Field Documentation

7.29.3.1 `uint32 MM_AllocInfo::allocNum`

7.29.3.2 `bool MM_AllocInfo::bSetFailure`

7.29.3.3 `uint32 MM_AllocInfo::lineNo`

7.29.3.4 `char* MM_AllocInfo::pFileName`

7.29.3.5 `void* MM_AllocInfo::pMemBlock`

7.29.3.6 `OsclMemStatsNode* MM_AllocInfo::pStatsNode`

7.29.3.7 `uint32 MM_AllocInfo::size`

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

- [oscl_mem_audit.h](#)

7.30 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](#)

7.30.1 Constructor & Destructor Documentation

7.30.1.1 [MM_AllocNode::MM_AllocNode \(\)](#) [inline]

7.30.1.2 [MM_AllocNode::~~MM_AllocNode \(\)](#) [inline]

7.30.2 Member Function Documentation

7.30.2.1 [void MM_AllocNode::operator delete \(void * ptr\) throw \(\)](#) [inline]

7.30.2.2 [void* MM_AllocNode::operator new \(oscl_memsize_t size, MM_AllocNode * ptr\)](#)
[inline]

7.30.2.3 [void* MM_AllocNode::operator new \(oscl_memsize_t size\)](#) [inline]

7.30.3 Field Documentation

7.30.3.1 [MM_AllocInfo*](#) [MM_AllocNode::pAllocInfo](#)

7.30.3.2 [MM_AllocNode*](#) [MM_AllocNode::pNext](#)

7.30.3.3 [MM_AllocNode*](#) [MM_AllocNode::pPrev](#)

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

- [oscl_mem_audit.h](#)

7.31 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]

7.31.1 Field Documentation

7.31.1.1 uint32 MM_AllocQueryInfo::allocNum

7.31.1.2 char MM_AllocQueryInfo::fileName[MM_ALLOC_MAX_QUERY_FILENAME_LEN]

7.31.1.3 uint32 MM_AllocQueryInfo::lineNo

7.31.1.4 const void* MM_AllocQueryInfo::pMemBlock

7.31.1.5 uint32 MM_AllocQueryInfo::size

7.31.1.6 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](#)

7.32 MM_Audit_Imp Class Reference

```
#include <oscl_mem_audit.h>
```

Public Methods

- [MM_Audit_Imp](#) ()
- [~MM_Audit_Imp](#) ()
- O_SCL_IMPORT_REF void * [MM_allocate](#) (const [OscMemStatsNode](#) *statsNode, uint32 sizeIn, const char *pFileName, uint32 lineNumber, bool allocNodeTracking=false)
- O_SCL_IMPORT_REF bool [MM_deallocate](#) (void *pMemBlockIn)
- O_SCL_IMPORT_REF [MM_Stats_t](#) * [MM_GetStats](#) (const char *const tagIn)
- O_SCL_IMPORT_REF uint32 [MM_GetStatsInDepth](#) (const char *tagIn, [MM_Stats_CB](#) *array_ptr, uint32 max_nodes)
- O_SCL_IMPORT_REF uint32 [MM_GetTreeNodees](#) (const char *tagIn)
- O_SCL_IMPORT_REF bool [MM_AddTag](#) (const char *tagIn)
- O_SCL_IMPORT_REF const [OscMemStatsNode](#) * [MM_GetTagNode](#) (const char *tagIn)
- O_SCL_IMPORT_REF const [OscMemStatsNode](#) * [MM_GetExistingTag](#) (const char *tagIn)
- O_SCL_IMPORT_REF const [OscMemStatsNode](#) * [MM_GetRootNode](#) ()
- O_SCL_IMPORT_REF [MM_AllocQueryInfo](#) * [MM_CreateAllocNodeInfo](#) (uint32 max_array_size)
- O_SCL_IMPORT_REF void [MM_ReleaseAllocNodeInfo](#) ([MM_AllocQueryInfo](#) *info)
- O_SCL_IMPORT_REF uint32 [MM_GetAllocNodeInfo](#) ([MM_AllocQueryInfo](#) *output_array, uint32 max_array_size, uint32 offset)
- O_SCL_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)
- O_SCL_IMPORT_REF void [MM_SetMode](#) (uint32 inMode)
- O_SCL_IMPORT_REF void [MM_SetPrefillPattern](#) (uint8 pattern)
- O_SCL_IMPORT_REF void [MM_SetPostfillPattern](#) (uint8 pattern)
- O_SCL_IMPORT_REF void [MM_SetTagLevel](#) (uint32 level)
- O_SCL_IMPORT_REF bool [MM_SetFailurePoint](#) (const char *tagIn, uint32 alloc_number)
- O_SCL_IMPORT_REF void [MM_UnsetFailurePoint](#) (const char *tagIn)
- [MM_AllocNode](#) * [addAllocNode](#) (void *pMem, uint32 sizeIn, [OscMemStatsNode](#) *pStatsNode, const char *pFileName, uint32 lineNumber)
- [OscMemStatsNode](#) * [removeAllocNode](#) (void *pMemBlockIn, uint32 &size)
- void [removeALLAllocNodes](#) ()
- [OscMemStatsNode](#) * [createStatsNode](#) (const char *tagIn)
- bool [updateStatsNode](#) ([OscMemStatsNode](#) *pCurrStatsNode, const [MM_Stats_t](#) &pDelta, bool b-Add)
- bool [updateStatsNodeInFailure](#) (const char *tagIn)
- bool [updateStatsNodeInFailure](#) ([OscMemStatsNode](#) *pStatsNode)
- bool [pruneSubtree](#) ([OscMemStatsNode](#) *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](#) (OscMemStatsNode *statsNode)
- bool [validate_all_heap](#) ()

Static Public Methods

- bool [validate](#) (void *ptrIn)
- [OscMemAudit](#) * [getAuditRoot](#) (void *ptrIn)
- uint32 [getSize](#) (void *ptrIn)

7.32.1 Constructor & Destructor Documentation

7.32.1.1 MM_Audit_Imp::MM_Audit_Imp ()

Constructor, create the root node in statistics table

7.32.1.2 MM_Audit_Imp::~~MM_Audit_Imp ()

A destructor, remove all the nodes in allocation and statistics table

7.32.2 Member Function Documentation

7.32.2.1 [MM_AllocNode](#)* MM_Audit_Imp::addAllocNode (void * pMem, uint32 sizeIn, [OscMemStatsNode](#) * pStatsNode, const char * pFileName, uint32 lineNumber)

Returns:

true if operation succeeds;

7.32.2.2 [OscMemStatsNode](#)* MM_Audit_Imp::createStatsNode (const char * tagIn)

Returns:

true if operation succeeds;

7.32.2.3 [OscMemAudit](#)* MM_Audit_Imp::getAuditRoot (void * ptrIn) [static]

Returns:

audit root pointer.

7.32.2.4 uint32 MM_Audit_Imp::getSize (void * ptrIn) [static]

Returns:

original block size. leaves if bad pointer.

7.32.2.5 uint32 MM_Audit_Imp::getTagActualSize (const char * tagIn)
Returns:

the size of the truncated tag; 0 means NO truncation

7.32.2.6 bool MM_Audit_Imp::isSetFailure (OscMemStatsNode * statsNode)
7.32.2.7 bool MM_Audit_Imp::isSetFailure (const char * tagIn)
Returns:

true if operation succeeds;

7.32.2.8 void MM_Audit_Imp::makeValidTag (const char * tagIn, MMAuditCharAutoPtr & autoptr)
Returns:

a valid tag; NULL will be converted into root tag

7.32.2.9 OSSL_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;

7.32.2.10 OSSL_IMPORT_REF void* MM_Audit_Imp::MM_allocate (const OscMemStatsNode * 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.

7.32.2.11 OSSL_IMPORT_REF MM_AllocQueryInfo* MM_Audit_Imp::MM_CreateAllocNodeInfo (uint32 max_array_size)

These APIs will allocate and release space for alloc node info, to be used with the MM_GetAllocNodeInfo API.

7.32.2.12 OSSL_IMPORT_REF bool MM_Audit_Imp::MM_deallocate (void * pMemBlockIn)
Returns:

true if operation succeeds;

7.32.2.13 uint32 MM_Audit_Imp::MM_GetAllocNo (void) [inline]

API to get the current allocation number

Returns:

the current allocation number

7.32.2.14 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

7.32.2.15 OSCL_IMPORT_REF const OsciMemStatsNode* 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;

7.32.2.16 uint32 MM_Audit_Imp::MM_GetMode (void) [inline]

API to get the operating mode of the mm_audit class.

7.32.2.17 uint32 MM_Audit_Imp::MM_GetNumAllocNodes () [inline]

API to get the number of allocation nodes (records) for allocations that are being tracked individually.

7.32.2.18 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.

7.32.2.19 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.

7.32.2.20 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.

7.32.2.21 OSCL_IMPORT_REF const OsciMemStatsNode* MM_Audit_Imp::MM_GetRootNode () [inline]
7.32.2.22 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

7.32.2.23 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

7.32.2.24 OSCL_IMPORT_REF const OsciMemStatsNode* MM_Audit_Imp::MM_GetTagNode (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 [OsciMemStatsNode](#) which should be passed to MM_allocate

7.32.2.25 OSCL_IMPORT_REF uint32 MM_Audit_Imp::MM_GetTreeNodees (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

7.32.2.26 OSCL_IMPORT_REF void MM_Audit_Imp::MM_ReleaseAllocNodeInfo (MM_AllocQueryInfo * info)

7.32.2.27 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;

7.32.2.28 OSCL_IMPORT_REF void MM_Audit_Imp::MM_SetMode (uint32 inMode)

API to set the operating mode of the mm_audit class.

7.32.2.29 OSCL_IMPORT_REF void MM_Audit_Imp::MM_SetPostfillPattern (uint8 pattern)

API to set the postfill pattern.

7.32.2.30 OSCL_IMPORT_REF void MM_Audit_Imp::MM_SetPrefillPattern (uint8 pattern)

API to set the prefill pattern.

7.32.2.31 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

7.32.2.32 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

7.32.2.33 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;

7.32.2.34 `bool MM_Audit_Imp::pruneSubtree (const char * tagIn)`

7.32.2.35 `bool MM_Audit_Imp::pruneSubtree (OscMemStatsNode * pNode)`

Returns:

true if operation succeeds;

7.32.2.36 `void MM_Audit_Imp::removeALLAllocNodes ()`

7.32.2.37 `OscMemStatsNode* MM_Audit_Imp::removeAllocNode (void * pMemBlockIn, uint32 & size)`

Returns:

true if operation succeeds;

7.32.2.38 `void MM_Audit_Imp::retrieveParentTag (char * tag)`

7.32.2.39 `int32 MM_Audit_Imp::retrieveParentTagLength (const char * tag, int32 bound)`

Returns:

the length of a immediate parent tag for the input tag

7.32.2.40 `bool MM_Audit_Imp::updateStatsNode (OscMemStatsNode * pCurrStatsNode, const MM_Stats_t & pDelta, bool bAdd)`

Returns:

true if operation succeeds;

7.32.2.41 `bool MM_Audit_Imp::updateStatsNodeInFailure (OscMemStatsNode * pStatsNode)`

7.32.2.42 `bool MM_Audit_Imp::updateStatsNodeInFailure (const char * tagIn)`

Returns:

true if operation succeeds;

7.32.2.43 `bool MM_Audit_Imp::validate (void * ptrIn) [static]`

Returns:

true if operation succeeds;

7.32.2.44 `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](#)

7.33 MM_AuditOverheadStats Struct Reference

```
#include <oscl_mem_audit.h>
```

Data Fields

- uint32 [per_allocation_overhead](#)
- uint32 [stats_overhead](#)

7.33.1 Field Documentation

7.33.1.1 uint32 MM_AuditOverheadStats::per_allocation_overhead

7.33.1.2 uint32 MM_AuditOverheadStats::stats_overhead

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

- [oscl_mem_audit.h](#)

7.34 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 \[3\]](#)

7.34.1 Constructor & Destructor Documentation

7.34.1.1 [MM_FailInsertParam::MM_FailInsertParam \(\) \[inline\]](#)

7.34.2 Member Function Documentation

7.34.2.1 [void MM_FailInsertParam::operator delete \(void * ptr\) throw \(\) \[inline\]](#)

7.34.2.2 [void* MM_FailInsertParam::operator new \(oscl_memsize_t size, MM_FailInsertParam * ptr\) \[inline\]](#)

7.34.2.3 [void* MM_FailInsertParam::operator new \(oscl_memsize_t size\) \[inline\]](#)

7.34.2.4 [void MM_FailInsertParam::reset \(\) \[inline\]](#)

7.34.3 Field Documentation

7.34.3.1 [uint32 MM_FailInsertParam::nAllocNum](#)

7.34.3.2 [uint16 MM_FailInsertParam::xsubi\[3\]](#)

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

- [oscl_mem_audit.h](#)

7.35 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](#)

7.35.1 Constructor & Destructor Documentation

7.35.1.1 [MM_Stats_CB::MM_Stats_CB \(\)](#) [inline]

7.35.2 Member Function Documentation

7.35.2.1 [void MM_Stats_CB::operator delete \(void * ptr\) throw \(\)](#) [inline]

7.35.2.2 [void* MM_Stats_CB::operator new \(oscl_memsize_t size, MM_Stats_CB * ptr\)](#)
[inline]

7.35.2.3 [void* MM_Stats_CB::operator new \(oscl_memsize_t size\)](#) [inline]

7.35.3 Field Documentation

7.35.3.1 [uint32 MM_Stats_CB::num_child_nodes](#)

7.35.3.2 [const MM_Stats_t* MM_Stats_CB::pStats](#)

7.35.3.3 [const char* MM_Stats_CB::tag](#)

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

- [oscl_mem_audit.h](#)

7.36 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](#)

7.36.1 Constructor & Destructor Documentation

7.36.1.1 `MM_Stats_t::MM_Stats_t()` [inline]

7.36.1.2 `MM_Stats_t::MM_Stats_t(uint32 sizeIn)` [inline]

7.36.2 Member Function Documentation

7.36.2.1 `void MM_Stats_t::operator delete (void * ptr) throw ()` [inline]

7.36.2.2 `void* MM_Stats_t::operator new (oscl_memsize_t size, MM_Stats_t * ptr)` [inline]

7.36.2.3 `void* MM_Stats_t::operator new (oscl_memsize_t size)` [inline]

7.36.2.4 `void MM_Stats_t::reset()` [inline]

7.36.2.5 `void MM_Stats_t::update (const MM_Stats_t & delta, bool add)` [inline]

7.36.3 Field Documentation

7.36.3.1 `uint32 MM_Stats_t::numAllocFails`

7.36.3.2 `uint32 MM_Stats_t::numAllocs`

7.36.3.3 `uint32 MM_Stats_t::numBytes`

7.36.3.4 `uint32 MM_Stats_t::peakNumAllocs`

7.36.3.5 `uint32 MM_Stats_t::peakNumBytes`

7.36.3.6 `uint32 MM_Stats_t::totalNumAllocs`

7.36.3.7 `uint32 MM_Stats_t::totalNumBytes`

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

- [oscl_mem_audit.h](#)

7.37 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 &ntpt) 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.

7.37.1 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.

7.37.2 Constructor & Destructor Documentation

7.37.2.1 OSCL_COND_IMPORT_REF NTPTIME::NTPTIME ()

The default constructor creates an NTPTIME instance representing the current system time.

7.37.2.2 OSCL_COND_IMPORT_REF NTPTIME::NTPTIME (const NTPTIME & src)

Copy constructor to create a new NTPTIME from an existing one.

7.37.2.3 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.

7.37.2.4 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.

7.37.2.5 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.

7.37.2.6 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.

7.37.3 Member Function Documentation

7.37.3.1 OSCL_COND_IMPORT_REF [uint32](#) NTPTIME::get_lower32 ()

This method returns the lower 32 bits of the 32.32 representation.

7.37.3.2 OSCL_COND_IMPORT_REF [uint32](#) NTPTIME::get_middle32 ()

Grab the middle 32 bits of the 64 bit 32.32 representation.

7.37.3.3 OSCL_COND_IMPORT_REF [uint32](#) NTPTIME::get_upper32 ()

This method returns the upper 32 bits of the 32.32 representation.

7.37.3.4 OSCL_COND_IMPORT_REF [uint64](#) NTPTIME::get_value ()

This method returns the 32.32 ntp representation.

7.37.3.5 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.

7.37.3.6 OSCL_COND_IMPORT_REF NTPTIME NTPTIME::operator- (const NTPTIME & *ntpt*) const

The - operator allows subtraction of one NTPTIME value from another. This is useful to measure an interval.

Parameters:

ntpt A reference to the NTPTIME object to be subtracted from this one.

7.37.3.7 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.

7.37.3.8 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).

7.37.3.9 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.

7.37.3.10 OSCL_IMPORT_REF int NTPTime::set_to_current_time ()

This method sets the 32.32 representation to the current system time value.

7.37.3.11 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.

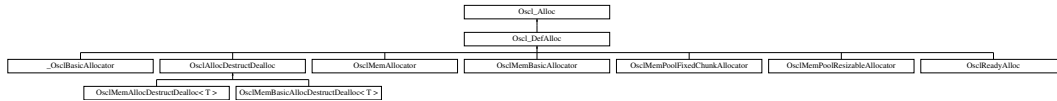
The documentation for this class was generated from the following file:

- [oscl_time.h](#)

7.38 Osci_Alloc Class Reference

```
#include <osci_defalloc.h>
```

Inheritance diagram for Osci_Alloc::



Public Methods

- virtual [OsciAny](#) * [allocate](#) (const uint32 size)=0
- virtual [OsciAny](#) * [allocate_fl](#) (const uint32 size, const char *file_name, const int line_num)

7.38.1 Member Function Documentation

7.38.1.1 virtual [OsciAny](#)* [Osci_Alloc::allocate](#) (const uint32 *size*) [pure virtual]

Implemented in [_OsciBasicAllocator](#), [Osci_DefAlloc](#), [OsciMemAllocator](#), [OsciMemBasicAllocator](#), [OsciMemAllocDestructDealloc< T >](#), [OsciMemBasicAllocDestructDealloc< T >](#), [OsciMemPoolFixedChunkAllocator](#), [OsciMemPoolResizableAllocator](#), and [OsciReadyAlloc](#).

7.38.1.2 virtual [OsciAny](#)* [Osci_Alloc::allocate_fl](#) (const uint32 *size*, const char **file_name*, const int *line_num*) [inline, virtual]

Reimplemented in [Osci_DefAlloc](#), [OsciMemAllocator](#), [OsciMemAllocDestructDealloc< T >](#), and [OsciReadyAlloc](#).

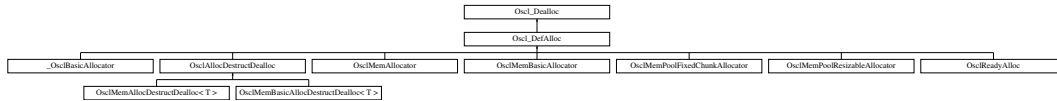
The documentation for this class was generated from the following file:

- [osci_defalloc.h](#)

7.39 Osl_Dealloc Class Reference

```
#include <osl_defalloc.h>
```

Inheritance diagram for Osl_Dealloc::



Public Methods

- virtual void [dealloc](#) ([OslAny](#) *p)=0

7.39.1 Member Function Documentation

7.39.1.1 virtual void [Osl_Dealloc::dealloc](#) ([OslAny](#) *p) [pure virtual]

Implemented in [_OslBasicAllocator](#), [Osl_DefAlloc](#), [OslMemAllocator](#), [OslMemBasicAllocator](#), [OslMemAllocDestructDealloc< T >](#), [OslMemBasicAllocDestructDealloc< T >](#), [OslMemPoolFixedChunkAllocator](#), [OslMemPoolResizableAllocator](#), and [OslReadyAlloc](#).

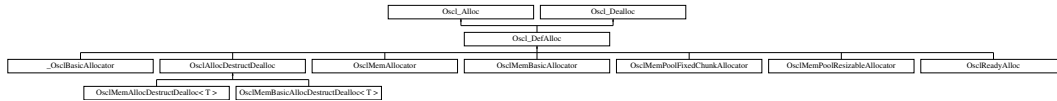
The documentation for this class was generated from the following file:

- [osl_defalloc.h](#)

7.40 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

7.40.1 Member Function Documentation

7.40.1.1 virtual [OsclAny](#)* [Oscl_DefAlloc::allocate](#) (const uint32 *size*) [pure virtual]

Implements [Oscl_Alloc](#).

Implemented in [_OsclBasicAllocator](#), [OsclMemAllocator](#), [OsclMemBasicAllocator](#), [OsclMemAllocDestructDealloc< T >](#), [OsclMemBasicAllocDestructDealloc< T >](#), [OsclMemPoolFixedChunkAllocator](#), [OsclMemPoolResizableAllocator](#), and [OsclReadyAlloc](#).

7.40.1.2 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](#).

7.40.1.3 virtual void [Oscl_DefAlloc::deallocate](#) ([OsclAny](#) **p*) [pure virtual]

Implements [Oscl_Dealloc](#).

Implemented in [_OsclBasicAllocator](#), [OsclMemAllocator](#), [OsclMemBasicAllocator](#), [OsclMemAllocDestructDealloc< T >](#), [OsclMemBasicAllocDestructDealloc< T >](#), [OsclMemPoolFixedChunkAllocator](#), [OsclMemPoolResizableAllocator](#), and [OsclReadyAlloc](#).

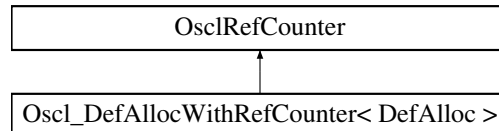
The documentation for this class was generated from the following file:

- [oscl_defalloc.h](#)

7.41 Osci_DefAllocWithRefCounter< DefAlloc > Class Template Reference

```
#include <osci_refcounter.h>
```

Inheritance diagram for Osci_DefAllocWithRefCounter< DefAlloc >::



Public Methods

- void [Delete](#) ()
- void [addRef](#) ()
- void [removeRef](#) ()
- uint32 [getCount](#) ()

Static Public Methods

- Osci_DefAllocWithRefCounter * [New](#) ()

7.41.1 Detailed Description

```
template<class DefAlloc> class Osci_DefAllocWithRefCounter< DefAlloc >
```

Implementation of an [Osci_DefAlloc](#) class with a built-in ref counter.

7.41.2 Member Function Documentation

7.41.2.1 `template<class DefAlloc> void Osci_DefAllocWithRefCounter< DefAlloc >::addRef ()`
 [inline, virtual]

Add to the reference count

Implements [OsciRefCounter](#).

7.41.2.2 `template<class DefAlloc> void Osci_DefAllocWithRefCounter< DefAlloc >::Delete ()`
 [inline]

Delete object

7.41.2.3 `template<class DefAlloc> uint32 Osci_DefAllocWithRefCounter< DefAlloc >::getCount ()`
 [inline, virtual]

Gets the current number of references



7.41 Osci_DefAllocWithRefCounter< DefAlloc > Class Template Reference

Implements [OsciRefCounter](#).

7.41.2.4 `template<class DefAlloc> Osci_DefAllocWithRefCounter*`
`Osci_DefAllocWithRefCounter< DefAlloc >::New() [inline, static]`

Create object

7.41.2.5 `template<class DefAlloc> void Osci_DefAllocWithRefCounter< DefAlloc >::removeRef`
`() [inline, virtual]`

Delete from reference count

Implements [OsciRefCounter](#).

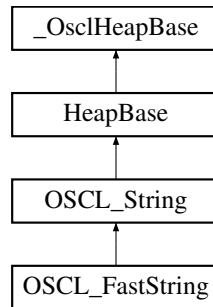
The documentation for this class was generated from the following file:

- [oscl_refcounter.h](#)

7.42 OSCL_FastString Class Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL_FastString::



Public Types

- typedef OSCL_String::chartype [chartype](#)
- typedef [TOSCL_StringOp](#) optype
- typedef [OSCL_wString::chartype](#) other_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](#) (const [other_chartype](#) *buf, uint32 numofbyte, [optype](#) op)
- OSCL_IMPORT_REF void [set_length](#) ()

Friends

- class [OSCL_String](#)

7.42.1 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.

7.42.2 Member Typedef Documentation

7.42.2.1 typedef OSCL_String::chartype OSCL_FastString::chartype

Reimplemented from [OSCL_String](#).

7.42.2.2 typedef TOSCL_StringOp OSCL_FastString::optype

7.42.2.3 typedef OSCL_wString::chartype OSCL_FastString::other_chartype

7.42.3 Constructor & Destructor Documentation

7.42.3.1 OSCL_IMPORT_REF OSCL_FastString::OSCL_FastString ()

Default constructor.

7.42.3.2 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.

7.42.3.3 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.

7.42.3.4 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.

7.42.3.5 OSCL_IMPORT_REF OSCL_FastString::~OSCL_FastString ()

7.42.4 Member Function Documentation

7.42.4.1 OSCL_IMPORT_REF const **char*** OSCL_FastString::get_cstr () [virtual]

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

Implements [OSCL_String](#).

7.42.4.2 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](#).

7.42.4.3 OSCL_IMPORT_REF uint32 OSCL_FastString::get_size () [virtual]

Pure virtuals from [OSCL_String](#)

Implements [OSCL_String](#).

7.42.4.4 OSCL_IMPORT_REF **char*** 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](#).

7.42.4.5 OSCL_IMPORT_REF OSCL_FastString& OSCL_FastString::operator= (const **char*** *cstr*)

Assignment operator

am: null-terminated string

Reimplemented from [OSCL_String](#).

7.42.4.6 OSCL_IMPORT_REF OSCL_FastString& OSCL_FastString::operator= (const OSCL_FastString & *src*)

Assignment operators

7.42.4.7 OSCL_IMPORT_REF void OSCL_FastString::set (const **other_char*** *buf*, uint32 *numofbyte*, **optype** *op*)

Set the contents of this string to a new string or character array, with conversion operation.

Parameters:

***buf*:** string or character array.

numofbyte: number of bytes available in the buffer. There must be enough space available for the converted string including its NULL terminator. The converted string may be smaller or larger than the original string.

op: conversion operation to apply. If numofbyte is not large enough for conversion, the function leaves. If input string is not null-terminated, the function leaves.

7.42.4.8 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.

7.42.4.9 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.

7.42.5 Friends And Related Function Documentation

7.42.5.1 friend class OSCL_String [friend]

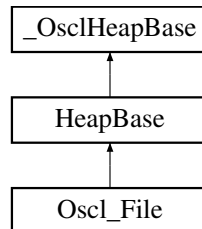
The documentation for this class was generated from the following file:

- [oscl_string_containers.h](#)

7.43 Osl_File Class Reference

```
#include <osl_file_io.h>
```

Inheritance diagram for Osl_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 `Osl_File` ()
- OSCL_IMPORT_REF `Osl_File` (uint32 aCacheSize)
- OSCL_IMPORT_REF `Osl_File` (uint32 aCacheSize, `OslFileHandle` *aFileHandle)
- OSCL_IMPORT_REF `~Osl_File` ()
- OSCL_IMPORT_REF void `SetPVCacheSize` (uint32 aSize)
- void `AddFixedCache` (const `OslFixedCacheParam` &aParam)
- void `RemoveFixedCache` (const `TOslFileOffset` &aPos)
- void `SetCacheObserver` (`OslCacheObserver` *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` (`OslFileHandle` *aHandle)
- OSCL_IMPORT_REF int32 `Open` (const char *filename, uint32 mode, `Osl_FileServer` &fileserv)
- OSCL_IMPORT_REF int32 `Open` (const `osl_wchar` *filename, uint32 mode, `Osl_FileServer` &fileserv)
- OSCL_IMPORT_REF uint32 `Read` (`OslAny` *buffer, uint32 size, uint32 numelements)
- OSCL_IMPORT_REF uint32 `Write` (const `OslAny` *buffer, uint32 size, uint32 numelements)
- OSCL_IMPORT_REF int32 `Seek` (`TOslFileOffset` offset, `seek_type` origin)
- OSCL_IMPORT_REF `TOslFileOffset` `Tell` ()
- OSCL_IMPORT_REF int32 `Close` ()
- OSCL_IMPORT_REF int32 `Flush` ()
- OSCL_IMPORT_REF int32 `SetSize` (uint32 size)
- OSCL_IMPORT_REF int32 `EndOfFile` ()
- OSCL_IMPORT_REF int32 `GetError` ()

- [OsciFileHandle * Handle \(\)](#)
- OSCL_IMPORT_REF [TOsciFileOffset Size \(\)](#)
- OSCL_IMPORT_REF void [SetLoggingEnable \(bool aEnable\)](#)
- OSCL_IMPORT_REF void [SetSummaryStatsLoggingEnable \(bool aEnable\)](#)

Friends

- class [OsciFileCache](#)
- class [OsciFileCacheBuffer](#)
- class [asyncfilereadwrite_test](#)
- class [largeasyncfilereadwrite_test](#)
- class [asyncfilereadcancel_test](#)

7.43.1 Member Enumeration Documentation

7.43.1.1 enum `Osci_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'.

7.43.1.2 enum `Osci_File::seek_type`

Enumeration values:

SEEKSET Beginning of file

SEEKCUR Current position of file pointer

SEEKEND End of file

7.43.1.3 enum `Osci_File::TSymbianAccessMode`

Defines mode options for SetNativeAccessMode on Symbian.

Enumeration values:

ESymbianAccessMode_Rfile

ESymbianAccessMode_RfileBuf

7.43.2 Constructor & Destructor Documentation

7.43.2.1 OSCL_IMPORT_REF Osl_File::Osl_File ()

Constructor

7.43.2.2 OSCL_IMPORT_REF Osl_File::Osl_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.

7.43.2.3 OSCL_IMPORT_REF Osl_File::Osl_File (uint32 *aCacheSize*, OslFileHandle * *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.

7.43.2.4 OSCL_IMPORT_REF Osl_File::~~Osl_File ()

Destructor

7.43.3 Member Function Documentation

7.43.3.1 void Osl_File::AddFixedCache (const OslFixedCacheParam & *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.

7.43.3.2 OSCL_IMPORT_REF int32 Osl_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

7.43.3.3 OSCL_IMPORT_REF int32 Osl_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:

7.43.3.4 OSCL_IMPORT_REF int32 Osl_File::Flush ()

The File Flush operation On an output stream Osl_FileFlush causes any buffered but unwritten data to be written to the file.

Returns:

returns 0 if successful, and a non-zero value otherwise

7.43.3.5 OSCL_IMPORT_REF int32 Osl_File::GetError ()

The File Error operation If no error has occurred on stream, returns 0. Otherwise, it returns a nonzero value

Returns:

7.43.3.6 [OslFileHandle*](#) Osl_File::Handle () [inline]

Retrieve the file handle.

Returns:

file handle

7.43.3.7 OSCL_IMPORT_REF int32 Osl_File::Open (const [oscl_wchar](#) * *filename*, uint32 *mode*, [Osl_FileServer](#) & *fileserv*)

Opens a file.

Note: when an external file handle is used, Open will attach to the file handle and initialize caching features, but will not do a native file open.

Parameters:

filename name of file to open (Unicode)

mode combination of open mode flags

fileserv fileserver to use

Returns:

returns 0 if successful and a non-zero value otherwise

7.43.3.8 OSCL_IMPORT_REF int32 Osl_File::Open (const char * *filename*, uint32 *mode*, Osl_FileServer & *fileserv*)

Opens a file.

Note: when an external file handle is used, Open will attach to the file handle and initialize caching features, but will not do a native file open.

Parameters:

filename name of file to open (Utf8)
mode combination of open mode flags
fileserv fileserver to use

Returns:

returns 0 if successful and a non-zero value otherwise

7.43.3.9 OSCL_IMPORT_REF uint32 Osl_File::Read (OslAny * *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.

7.43.3.10 void Osl_File::RemoveFixedCache (const TOslFileOffset & *aPos*) [inline]

RemoveFixedCache removes a fixed cache.

Parameters:

aPos: Cache location and size.

7.43.3.11 OSCL_IMPORT_REF int32 Osl_File::Seek (TOslFileOffset *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

7.43.3.12 OSCL_IMPORT_REF void Osl_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.

7.43.3.13 void Osl_File::SetCacheObserver (OslCacheObserver * *aObs*) [inline]

7.43.3.14 OSCL_IMPORT_REF int32 Osl_File::SetFileHandle (OslFileHandle * *aHandle*)

SetFileHandle adds an open file handle to the Osl_File object. The Osl_File object will use that handle to access the file.

This call is not available when the Osl_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 [OslFileHandle](#) object, pass the [OslFileHandle](#) pointer to SetFileHandle, call [Osl_File::Open](#), then proceed to use the Osl_File object, finally calling [Osl_File::Close](#). In this usage mode, [Osl_File::Open](#) and [Osl_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.

7.43.3.15 OSCL_IMPORT_REF void Osl_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 "Osl_File", plus full logging of native operation entry & exit using logger object "[OslNativeFile](#)".

Parameters:

aEnable: true to enable, false to disable logging.

7.43.3.16 OSCL_IMPORT_REF void Osl_File::SetNativeAccessMode (uint32 *aMode*)

SetNativeAccessMode allows switching between different native file access modes, when available.

Note: for 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.

7.43.3.17 OSCL_IMPORT_REF void Osl_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 buffing is not available on the platform, this call will have no effect.

Parameters:

aSize: native buffer size in bytes. Zero disables the feature.

7.43.3.18 OSCL_IMPORT_REF void Osl_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.

7.43.3.19 OSCL_IMPORT_REF int32 Osl_File::SetSize (uint32 size)

The File SetSize operation If the file has been opened for writing This sets the size of the file. The file pointer position is unchanged unless the pointers position is greater than the new filesize.

Returns:

returns 0 if successful, and a non-zero value otherwise

7.43.3.20 OSCL_IMPORT_REF void Osl_File::SetSummaryStatsLoggingEnable (bool aEnable)

SetSummaryStatsLoggingEnable configures the [PVLogger](#) output for this file. This will enable summary statistics logging only, using the logger object "OslFileStats".

Parameters:

aEnable: true to enable, false to disable stats logging.

7.43.3.21 OSCL_IMPORT_REF TOSclFileOffset Osl_File::Size ()

Get the file size in bytes.

Returns:

- The size of the file, or -1 on error.

7.43.3.22 OSCL_IMPORT_REF TOsciFileOffset Osci_File::Tell ()

The File Tell operation Returns the current file position for file specified by fp

7.43.3.23 OSCL_IMPORT_REF uint32 Osci_File::Write (const OsciAny * 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

7.43.4 Friends And Related Function Documentation

7.43.4.1 friend class `asyncfilereadcancel_test` [friend]

7.43.4.2 friend class `asyncfilereadwrite_test` [friend]

7.43.4.3 friend class `largeasyncfilereadwrite_test` [friend]

7.43.4.4 friend class `OsciFileCache` [friend]

7.43.4.5 friend class `OsciFileCacheBuffer` [friend]

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

- [osci_file_io.h](#)

7.44 `OscFile::OscCacheObserver` Class Reference

```
#include <oscl_file_io.h>
```

Public Methods

- virtual `OscFileCacheBuffer` * `ChooseCurCache` (`OscFileCache` &`aContext`, `TOscFileOffset` `aPos`)=0

7.44.1 Detailed Description

For defining a cache observer. Cache observer can implement customized cache schemes by replacing the `SetCachePosition` routine.

7.44.2 Member Function Documentation

- 7.44.2.1 virtual `OscFileCacheBuffer`* `OscFile::OscCacheObserver::ChooseCurCache` (`OscFileCache` & `aContext`, `TOscFileOffset` `aPos`) [pure virtual]

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

- `oscl_file_io.h`

7.45 Osl_File::OslFixedCacheParam Class Reference

```
#include <osl_file_io.h>
```

Public Methods

- bool [Contains](#) ([TOslFileOffset](#) pos) const

Data Fields

- [TOslFileOffset](#) iFilePosition
- uint32 iSize

7.45.1 Detailed Description

Parameters for defining a fixed cache

7.45.2 Member Function Documentation

7.45.2.1 bool [Osl_File::OslFixedCacheParam::Contains](#) ([TOslFileOffset](#) pos) const
[inline]

7.45.3 Field Documentation

7.45.3.1 [TOslFileOffset](#) [Osl_File::OslFixedCacheParam::iFilePosition](#)

7.45.3.2 uint32 [Osl_File::OslFixedCacheParam::iSize](#)

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

- [osl_file_io.h](#)

7.46 Osl_FileFind Class Reference

```
#include <osl_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 `osl_wchar` * `FindFirst` (const `osl_wchar` *directory, const `osl_wchar` *pattern, `osl_wchar` *buf, uint32 buflen)
- `OSCL_IMPORT_REF` char * `FindNext` (char *buf, uint32 buflen)
- `OSCL_IMPORT_REF` `osl_wchar` * `FindNext` (`osl_wchar` *buf, uint32 buflen)
- `OSCL_IMPORT_REF` void `Close` ()
- `OSCL_IMPORT_REF` `element_type` `GetElementType` ()
- `OSCL_IMPORT_REF` `error_type` `GetLastError` ()
- `OSCL_IMPORT_REF` `Osl_FileFind` ()
- `OSCL_IMPORT_REF` `~Osl_FileFind` ()

7.46.1 Detailed Description

Osl_FileFind class defines the generic way of finding filesystem elements that match a pattern within a directory

7.46.2 Member Enumeration Documentation

7.46.2.1 enum Osl_FileFind::element_type

Enumeration values:

`FILE_TYPE`
`DIR_TYPE`
`INVALID_TYPE`

7.46.2.2 enum Osl_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

7.46.3 Constructor & Destructor Documentation

7.46.3.1 OSCL_IMPORT_REF Osl_FileFind::Osl_FileFind ()

constructor.

Returns:
none

7.46.3.2 OSCL_IMPORT_REF Osl_FileFind::~~Osl_FileFind ()

destructor. will deallocate open handles if necessary

Returns:
none

7.46.4 Member Function Documentation

7.46.4.1 OSCL_IMPORT_REF void Osl_FileFind::Close ()

closes the handle to directory.

Returns:
none

7.46.4.2 OSCL_IMPORT_REF const **oscl_wchar*** Osl_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 first found element, or NULL otherwise. On a NULL return value, [GetLastError\(\)](#) returns a more detailed error.

7.46.4.3 OSL_IMPORT_REF `const char* Osl_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 first found element, or NULL otherwise. On a NULL return value, `GetLastError()` returns a more detailed error.

7.46.4.4 OSL_IMPORT_REF `oscl_wchar* Osl_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.

7.46.4.5 OSL_IMPORT_REF `char* Osl_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.

7.46.4.6 OSL_IMPORT_REF `element_type Osl_FileFind::GetElementType ()`

Returns the element type for the last element returned

Returns:

see enumeration above for more info.

7.46.4.7 OSCL_IMPORT_REF [error_type](#) Oslc_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](#)

7.47 Osl_FileServer Class Reference

```
#include <osl_file_server.h>
```

Public Methods

- OSCL_IMPORT_REF [Osl_FileServer](#) ()
- OSCL_IMPORT_REF [~Osl_FileServer](#) ()
- OSCL_IMPORT_REF int32 [Connect](#) (bool aShareSession=false)
- OSCL_IMPORT_REF int32 [Close](#) ()
- OSCL_IMPORT_REF int32 [Osl_DeleteFile](#) (const char *filename)
- OSCL_IMPORT_REF int32 [Osl_DeleteFile](#) (const [osl_wchar](#) *filename)

Friends

- class [Osl_File](#)
- class [OslNativeFile](#)

7.47.1 Constructor & Destructor Documentation

7.47.1.1 OSCL_IMPORT_REF Osl_FileServer::Osl_FileServer ()

Constructor

7.47.1.2 OSCL_IMPORT_REF Osl_FileServer::~~Osl_FileServer ()

Destructor

7.47.2 Member Function Documentation

7.47.2.1 OSCL_IMPORT_REF int32 Osl_FileServer::Close ()

Closes a file server.

Returns:

returns 0 on success and a non-zero value otherwise

7.47.2.2 OSCL_IMPORT_REF int32 Osl_FileServer::Connect (bool *aShareSession* = false)

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

7.47.2.3 OSCL_IMPORT_REF int32 Osci_FileServer::Osci_DeleteFile (const [osci_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.

7.47.2.4 OSCL_IMPORT_REF int32 Osci_FileServer::Osci_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.

7.47.3 Friends And Related Function Documentation

7.47.3.1 friend class Osci_File [friend]

7.47.3.2 friend class OsciNativeFile [friend]

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

- [osci_file_server.h](#)

7.48 oscl_fsstat Struct Reference

```
#include <oscl_file_dir_utils.h>
```

Data Fields

- [uint64 freebytes](#)
- [uint64 totalbytes](#)

7.48.1 Field Documentation

7.48.1.1 [uint64 oscl_fsstat::freebytes](#)

7.48.1.2 [uint64 oscl_fsstat::totalbytes](#)

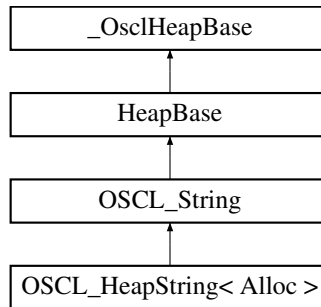
The documentation for this struct was generated from the following file:

- [oscl_file_dir_utils.h](#)

7.49 OSCL_HeapString< Alloc > Class Template Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL_HeapString< Alloc >::



Public Types

- typedef OSCL_String::chartype [chartype](#)
- typedef TOSCL_StringOp [optype](#)
- typedef OSCL_wString::chartype [other_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)
- void [set](#) (const other_chartype *buf, optype op)
- void [set](#) (const other_chartype *buf, uint32 length, optype op)

Friends

- class [OSCL_String](#)

7.49.1 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](#).

7.49.2 Member Typedef Documentation

7.49.2.1 template<class Alloc> typedef OSCL_String::chartype OSCL_HeapString< Alloc >::chartype

Reimplemented from [OSCL_String](#).

7.49.2.2 template<class Alloc> typedef **TOSCL_StringOp OSCL_HeapString< Alloc >::optype**

7.49.2.3 template<class Alloc> typedef **OSCL_wString::chartype OSCL_HeapString< Alloc >::other_chartype**

7.49.3 Friends And Related Function Documentation

7.49.3.1 template<class Alloc> friend class OSCL_String [friend]

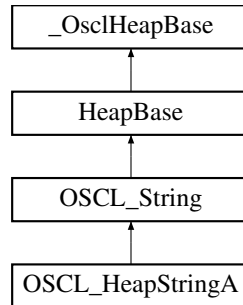
The documentation for this class was generated from the following file:

- [oscl_string_containers.h](#)

7.50 OSCL_HeapStringA Class Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL_HeapStringA::



Public Types

- typedef OSCL_String::chartype [chartype](#)
- typedef TOSCL_StringOp [optype](#)
- typedef OSCL_wString::chartype [other_chartype](#)

Public Methods

- OSCL_IMPORT_REF [OSCL_HeapStringA](#) ()
- OSCL_IMPORT_REF [OSCL_HeapStringA](#) (Osc_DefAlloc *alloc, OscRefCount *ref=NULL)
- OSCL_IMPORT_REF [OSCL_HeapStringA](#) (const OSCL_HeapStringA &src)
- OSCL_IMPORT_REF [OSCL_HeapStringA](#) (const OSCL_HeapStringA &src, Osc_DefAlloc *alloc, OscRefCount *ref=NULL)
- OSCL_IMPORT_REF [OSCL_HeapStringA](#) (const OSCL_String &src, Osc_DefAlloc *alloc=NULL, OscRefCount *ref=NULL)
- OSCL_IMPORT_REF [OSCL_HeapStringA](#) (const [chartype](#) *cstr, Osc_DefAlloc *alloc=NULL, OscRefCount *ref=NULL)
- OSCL_IMPORT_REF [OSCL_HeapStringA](#) (const [chartype](#) *buf, uint32 length, Osc_DefAlloc *alloc=NULL, OscRefCount *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)
- OSCL_IMPORT_REF void [set](#) (const [other_chartype](#) *buf, [optype](#) op)
- OSCL_IMPORT_REF void [set](#) (const [other_chartype](#) *buf, uint32 length, [optype](#) op)

Friends

- class [OSCL_String](#)

7.50.1 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 [OscMemAllocator](#) 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.

7.50.2 Member Typedef Documentation

7.50.2.1 typedef OSCL_String::chartype OSCL_HeapStringA::chartype

Reimplemented from [OSCL_String](#).

7.50.2.2 typedef OSCL_StringOp OSCL_HeapStringA::optype

7.50.2.3 typedef OSCL_wString::chartype OSCL_HeapStringA::other_chartype

7.50.3 Constructor & Destructor Documentation

7.50.3.1 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 this object will use an [OscMemAllocator](#).

7.50.3.2 OSCL_IMPORT_REF OSCL_HeapStringA::OSCL_HeapStringA (Osc_DefAlloc * alloc, OscRefCounter * ref = NULL)

7.50.3.3 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 this object will use an [OscMemAllocator](#).

7.50.3.4 OSCL_IMPORT_REF OSCL_HeapStringA::OSCL_HeapStringA (const OSCL_HeapStringA & src, [OscDefAlloc](#) * alloc, [OscRefCount](#) * ref = NULL)

7.50.3.5 OSCL_IMPORT_REF OSCL_HeapStringA::OSCL_HeapStringA (const [OSCL_String](#) & src, [OscDefAlloc](#) * alloc = NULL, [OscRefCount](#) * ref = NULL)

7.50.3.6 OSCL_IMPORT_REF OSCL_HeapStringA::OSCL_HeapStringA (const [char](#)* cstr, [OscDefAlloc](#) * alloc = NULL, [OscRefCount](#) * 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 this object will use an [OscMemAllocator](#).

7.50.3.7 OSCL_IMPORT_REF OSCL_HeapStringA::OSCL_HeapStringA (const [char](#)* buf, [uint32](#) length, [OscDefAlloc](#) * alloc = NULL, [OscRefCount](#) * 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 this object will use an [OscMemAllocator](#).

7.50.3.8 OSCL_IMPORT_REF OSCL_HeapStringA::~OSCL_HeapStringA ()

7.50.4 Member Function Documentation

7.50.4.1 OSCL_IMPORT_REF const [char](#)* OSCL_HeapStringA::get_cstr () [virtual]

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

Implements [OSCL_String](#).

7.50.4.2 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](#).

7.50.4.3 OSCL_IMPORT_REF uint32 OSCL_HeapStringA::get_size () [virtual]

Pure virtuals from [OSCL_String](#)

Implements [OSCL_String](#).

7.50.4.4 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](#).

7.50.4.5 OSCL_IMPORT_REF OSCL_HeapStringA& OSCL_HeapStringA::operator= (const chartype * *cstr*)

Assignment operator

am: null-terminated string

Reimplemented from [OSCL_String](#).

7.50.4.6 OSCL_IMPORT_REF OSCL_HeapStringA& OSCL_HeapStringA::operator= (const OSCL_String & *src*)

Assignment operator

Reimplemented from [OSCL_String](#).

7.50.4.7 OSCL_IMPORT_REF OSCL_HeapStringA& OSCL_HeapStringA::operator= (const OSCL_HeapStringA & *src*)

Assignment operators

7.50.4.8 OSCL_IMPORT_REF void OSCL_HeapStringA::set (const other_chartype * *buf*, uint32 *length*, optype *op*)

Set the contents of this string to a new string or character array, with conversion operation.

Parameters:

buf: string or character array.

length: number of characters to copy.

op: conversion operation to apply

7.50.4.9 OSCL_IMPORT_REF void OSCL_HeapStringA::set (const [other_chartype](#) * *buf*, [optype](#) *op*)

Set the contents of this string to a new string, with conversion operation.

Parameters:

buf: NULL-terminated wide string.

op: conversion operation to apply

7.50.4.10 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.

7.50.5 Friends And Related Function Documentation**7.50.5.1 friend class OSCL_String [friend]**

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

- [oscl_string_containers.h](#)

7.51 Osl_Int64_Utils Class Reference

The Osl_Int64_Utils class provides a wrapper for commonly used int64/uint64 operations.

```
#include <osl_int64_utils.h>
```

Static Public Methods

- OSCL_IMPORT_REF void [set_int64](#) (int64 &input_value, const int32 upper, const uint32 lower)
- OSCL_IMPORT_REF int32 [get_int64_upper32](#) (const int64 &input_value)
- OSCL_IMPORT_REF uint32 [get_int64_lower32](#) (const int64 &input_value)
- OSCL_IMPORT_REF uint32 [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)

7.51.1 Detailed Description

The Osl_Int64_Utils class provides a wrapper for commonly used int64/uint64 operations.

The Osl_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.

7.51.2 Member Function Documentation

- 7.51.2.1 OSCL_IMPORT_REF uint32 Osl_Int64_Utils::get_int64_lower32 (const [int64](#) & *input_value*) [static]
- 7.51.2.2 OSCL_IMPORT_REF uint32 Osl_Int64_Utils::get_int64_middle32 (const [int64](#) & *input_value*) [static]
- 7.51.2.3 OSCL_IMPORT_REF int32 Osl_Int64_Utils::get_int64_upper32 (const [int64](#) & *input_value*) [static]
- 7.51.2.4 OSCL_IMPORT_REF uint32 Osl_Int64_Utils::get_uint64_lower32 (const [uint64](#) & *input_value*) [static]
- 7.51.2.5 OSCL_IMPORT_REF uint32 Osl_Int64_Utils::get_uint64_middle32 (const [uint64](#) & *input_value*) [static]
- 7.51.2.6 OSCL_IMPORT_REF uint32 Osl_Int64_Utils::get_uint64_upper32 (const [uint64](#) & *input_value*) [static]
- 7.51.2.7 OSCL_IMPORT_REF void Osl_Int64_Utils::set_int64 ([int64](#) & *input_value*, const int32 *upper*, const uint32 *lower*) [static]
- 7.51.2.8 OSCL_IMPORT_REF void Osl_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](#)

7.52 Osl_Less< T > Struct Template Reference

```
#include <oscl_map.h>
```

Public Methods

- `bool operator()` (const T &x, const T &y) const

```
template<class T> struct Osl_Less< T >
```

7.52.1 Member Function Documentation

7.52.1.1 `template<class T> bool Osl_Less< T >::operator()` (const T & x, const T & y) const
[inline]

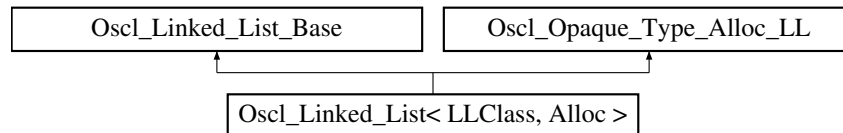
The documentation for this struct was generated from the following file:

- [oscl_map.h](#)

7.53 Osl_Linked_List< LLClass, Alloc > Class Template Reference

```
#include <osl_linked_list.h>
```

Inheritance diagram for Osl_Linked_List< LLClass, Alloc >::



Public Methods

- [Osl_Linked_List \(\)](#)
- [~Osl_Linked_List \(\)](#)
- [void clear \(\)](#)
- [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 insert_element \(const LLClass &new_element, int index\)](#)
- [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 \(const LLClass &data_to_move\)](#)

7.53.1 Detailed Description

```
template<class LLClass, class Alloc> class Osl_Linked_List< LLClass, Alloc >
```

Osl Linked List Class

7.53.2 Constructor & Destructor Documentation

7.53.2.1 `template<class LLClass, class Alloc> Osl_Linked_List< LLClass, Alloc >::Osl_Linked_List () [inline]`

Initialized the protected variables of list.

7.53.2.2 `template<class LLClass, class Alloc> OscL_Linked_List< LLClass, Alloc >::~~OscL_Linked_List () [inline]`

The destructor.

7.53.3 Member Function Documentation

7.53.3.1 `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 creates the list and adds the element as first element of list.

Parameters:

new_element the element to be added in the list.

Returns:

32-bit integer on success returns 1.

7.53.3.2 `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 already exists then it adds element at start of list otherwise it creates the list and adds the element as first element of list.

Parameters:

new_element the element to be added in the list.

Returns:

32-bit integer on success returns 1.

7.53.3.3 `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 nodes added to the list then returns 1 otherwise returns 0.

Reimplemented from [OscL_Linked_List_Base](#).

7.53.3.4 `template<class LLClass, class Alloc> void OscL_Linked_List< LLClass, Alloc >::clear () [inline]`

7.53.3.5 `template<class LLClass, class Alloc> int32 OscL_Linked_List< LLClass, Alloc >::dequeue_element (LLClass & element) [inline]`

7.53.3.6 `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.

7.53.3.7 `template<class LLClass, class Alloc> int32 Osci_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 integer, If first element found, it returns 1 otherwise it returns 0

7.53.3.8 `template<class LLClass, class Alloc> int32 Osci_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.

7.53.3.9 `template<class LLClass, class Alloc> int32 Osci_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 integer ,if next element is found in list, it returns 1 otherwise it returns 0

7.53.3.10 `template<class LLClass, class Alloc> int32 Osci_Linked_List< LLClass, Alloc >::get_num_elements () [inline]`

Get number of elements in the list.

Returns:

32-bit integer, number of elements in list.

7.53.3.11 `template<class LLClass, class Alloc> int32 Oslc_Linked_List< LLClass, Alloc >::insert_element (const LLClass & new_element, int index) [inline]`

Inserts new element in the list. If the index is past the end of the list it creates 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.

7.53.3.12 `template<class LLClass, class Alloc> int32 Oslc_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.

7.53.3.13 `template<class LLClass, class Alloc> int32 Oslc_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.

7.53.3.14 `template<class LLClass, class Alloc> int32 Oslc_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 [Oslc_Linked_List_Base](#).

7.53.3.15 `template<class LLClass, class Alloc> int32 Osci_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 found in the list returns 1 otherwise returns 0.

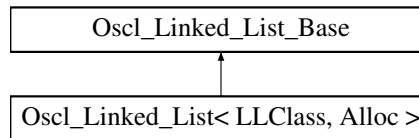
The documentation for this class was generated from the following file:

- [osci_linked_list.h](#)

7.54 Osl_Linked_List_Base Class Reference

```
#include <osl_linked_list.h>
```

Inheritance diagram for Osl_Linked_List_Base::



Protected Methods

- virtual `~Osl_Linked_List_Base ()`
- OSCL_IMPORT_REF void `construct (Osl_Opaque_Type_Alloc_LL *op)`
- OSCL_IMPORT_REF void `destroy ()`
- OSCL_IMPORT_REF int32 `get_first (OslAny *ele)`
- OSCL_IMPORT_REF int32 `get_next (OslAny *ele)`
- OSCL_IMPORT_REF int32 `check_list ()`
- OSCL_IMPORT_REF int32 `add_element (const OslAny *new_element)`
- OSCL_IMPORT_REF int32 `add_to_front (const OslAny *new_element)`
- OSCL_IMPORT_REF int32 `insert_element (const OslAny *new_element, int index)`
- OSCL_IMPORT_REF int32 `get_element (int32 index, OslAny *element)`
- OSCL_IMPORT_REF int32 `remove_element (const OslAny *data_to_remove)`
- OSCL_IMPORT_REF int32 `get_index (const OslAny *data)`
- OSCL_IMPORT_REF int32 `remove_element (const int32 index_to_remove)`
- OSCL_IMPORT_REF int32 `move_to_end (const OslAny *data_to_move)`
- OSCL_IMPORT_REF int32 `move_to_front (const OslAny *data_to_move)`

Protected Attributes

- OslAny * `head`
- OslAny * `tail`
- OslAny * `iterator`
- int32 `num_elements`
- uint32 `sizeof_T`

7.54.1 Detailed Description

Osl Linked List Base Class. A non-templated base class is used to avoid large inline functions in the [Osl_Linked_List](#) implementation.

7.54.2 Constructor & Destructor Documentation

7.54.2.1 `virtual Osl_Linked_List_Base::~~Osl_Linked_List_Base ()` [inline, protected, virtual]

7.54.3 Member Function Documentation

7.54.3.1 `OSCL_IMPORT_REF int32 Osl_Linked_List_Base::add_element (const OslAny * 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.

7.54.3.2 `OSCL_IMPORT_REF int32 Osl_Linked_List_Base::add_to_front (const OslAny * 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.

7.54.3.3 `OSCL_IMPORT_REF int32 Osl_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 [Osl_Linked_List<LLClass, Alloc >](#).

7.54.3.4 `OSCL_IMPORT_REF void Osl_Linked_List_Base::construct (Osl_Opaque_Type_Alloc_LL * op)` [protected]

7.54.3.5 `OSCL_IMPORT_REF void Osl_Linked_List_Base::destroy ()` [protected]

7.54.3.6 `OSCL_IMPORT_REF int32 Osl_Linked_List_Base::get_element (int32 index, OslAny * 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.

7.54.3.7 OSCL_IMPORT_REF int32 Osl_Linked_List_Base::get_first (OslAny * 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 integer, If first element found, it returns 1 otherwise it returns 0

7.54.3.8 OSCL_IMPORT_REF int32 Osl_Linked_List_Base::get_index (const OslAny * 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.

7.54.3.9 OSCL_IMPORT_REF int32 Osl_Linked_List_Base::get_next (OslAny * 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 integer ,if next element is found in list, it returns 1 otherwise it returns 0

7.54.3.10 OSCL_IMPORT_REF int32 Osl_Linked_List_Base::insert_element (const OslAny * new_element, int index) [protected]

Inserts new element in the list. If the index is past the end of the list it creates 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.

7.54.3.11 OSCL_IMPORT_REF int32 Osl_Linked_List_Base::move_to_end (const OslAny * *data_to_move*) [protected]

Moves the element to end of the list

Parameters:

data_to_move

Returns:

On success returns 1 otherwise returns 0.

7.54.3.12 OSCL_IMPORT_REF int32 Osl_Linked_List_Base::move_to_front (const OslAny * *data_to_move*) [protected]

Moves the element to front of the list

Parameters:

data_to_move

Returns:

On success returns 1 otherwise returns 0.

7.54.3.13 OSCL_IMPORT_REF int32 Osl_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 [Osl_Linked_List<LLClass, Alloc >](#).

7.54.3.14 OSCL_IMPORT_REF int32 Osl_Linked_List_Base::remove_element (const OslAny * *data_to_remove*) [protected]

Removes the element from the list.

Parameters:

data_to_remove

Returns:

32-bit integer on if element found in the list returns 1 otherwise returns 0.

7.54.4 Field Documentation

7.54.4.1 **OslAny*** Osl_Linked_List_Base::head [protected]

7.54.4.2 **OslAny*** Osl_Linked_List_Base::iterator [protected]

7.54.4.3 **int32** Osl_Linked_List_Base::num_elements [protected]

7.54.4.4 **uint32** Osl_Linked_List_Base::sizeof_T [protected]

7.54.4.5 **OslAny*** Osl_Linked_List_Base::tail [protected]

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

- [oscl_linked_list.h](#)

7.55 Osl_Map< Key, T, Alloc, Compare > Class Template Reference

```
#include <oscl_map.h>
```

Public Types

- typedef Key [key_type](#)
- typedef Compare [key_compare](#)
- typedef Osl_Pair< const Key, T > [value_type](#)
- typedef Osl_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 Osl_Pair< iterator, bool > [pair_iterator_bool](#)
- typedef Osl_Pair< iterator, iterator > [pair_iterator_iterator](#)
- typedef Osl_Pair< const_iterator, const_iterator > [pair_citerator_citerator](#)

Public Methods

- [Osl_Map](#) (const Compare &comp=Compare())
- [Osl_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

7.55.1 Detailed Description

template<class **Key**, class **T**, class **Alloc**, class **Compare** = **Osl_Less**<**Key**>> **class Osl_Map**< **Key**, **T**, **Alloc**, **Compare** >

Osl_Map Class. A subset of STL::Map methods. Osl_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. Osl_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.

7.55.2 Member Typedef Documentation

- 7.55.2.1 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef rep_type::const_iterator Osl_Map< Key, T, Alloc, Compare >::const_iterator`
- 7.55.2.2 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef rep_type::const_reference Osl_Map< Key, T, Alloc, Compare >::const_reference`
- 7.55.2.3 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef rep_type::iterator Osl_Map< Key, T, Alloc, Compare >::iterator`
- 7.55.2.4 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef Compare Osl_Map< Key, T, Alloc, Compare >::key_compare`
- 7.55.2.5 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef Key Osl_Map< Key, T, Alloc, Compare >::key_type`
- 7.55.2.6 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef Osl_Pair<const_iterator, const_iterator> Osl_Map< Key, T, Alloc, Compare >::pair_citerator_citerator`
- 7.55.2.7 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef Osl_Pair<iterator, bool> Osl_Map< Key, T, Alloc, Compare >::pair_iterator_bool`
- 7.55.2.8 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef Osl_Pair<iterator, iterator> Osl_Map< Key, T, Alloc, Compare >::pair_iterator_iterator`
- 7.55.2.9 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef rep_type::pointer Osl_Map< Key, T, Alloc, Compare >::pointer`
- 7.55.2.10 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef rep_type::reference Osl_Map< Key, T, Alloc, Compare >::reference`
- 7.55.2.11 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef Osl_Map<Key, T, Alloc, Compare> Osl_Map< Key, T, Alloc, Compare >::self`
- 7.55.2.12 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef rep_type::size_type Osl_Map< Key, T, Alloc, Compare >::size_type`
- 7.55.2.13 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef Osl_Pair<const Key, T> Osl_Map< Key, T, Alloc, Compare >::value_type`

7.55.3 Constructor & Destructor Documentation

- 7.55.3.1 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> Osl_Map< Key, T, Alloc, Compare >::Osl_Map (const Compare & comp = Compare()) [inline]`

Creates an empty map using comp as the key compare object

7.55.3.2 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> Osl_Map<Key, T, Alloc, Compare >::Osl_Map (const self & x) [inline]`

Osl_Map copy constructor

7.55.4 Member Function Documentation

7.55.4.1 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> const_iterator Osl_Map< Key, T, Alloc, Compare >::begin () const [inline]`

Returns a const iterator pointing to the beginning of the map

7.55.4.2 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> iterator Osl_Map< Key, T, Alloc, Compare >::begin () [inline]`

Returns an iterator pointing to the beginning of the map

7.55.4.3 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> void Osl_Map< Key, T, Alloc, Compare >::clear () [inline]`

Erases all elements

7.55.4.4 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> size_type Osl_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.

7.55.4.5 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> bool Osl_Map< Key, T, Alloc, Compare >::empty () const [inline]`

Returns true if map size is 0

7.55.4.6 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> const_iterator Osl_Map< Key, T, Alloc, Compare >::end () const [inline]`

Returns a const iterator pointing to the end of the map.

7.55.4.7 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> iterator Osl_Map< Key, T, Alloc, Compare >::end () [inline]`

Returns an iterator pointing to the end of the map.

7.55.4.8 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> pair_citerator_citerator Osl_Map< Key, T, Alloc, Compare >::equal_range (const key_type & x) const [inline]`

Finds a range containing all elements whose key is x

7.55.4.9 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> pair_iterator_iterator Osl_Map< Key, T, Alloc, Compare >::equal_range (const key_type & x) [inline]`

Finds a range containing all elements whose key is x

7.55.4.10 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> void Osl_Map< Key, T, Alloc, Compare >::erase (iterator first, iterator last) [inline]`

Erases all elements in the range [first,last)

7.55.4.11 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> size_type Osl_Map< Key, T, Alloc, Compare >::erase (const key_type & x) [inline]`

Erases the element with key x

7.55.4.12 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> void Osl_Map< Key, T, Alloc, Compare >::erase (iterator position) [inline]`

Erases the element pointed to by position

7.55.4.13 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> const_iterator Osl_Map< Key, T, Alloc, Compare >::find (const key_type & x) const [inline]`

Finds an element whose key is x

7.55.4.14 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> iterator Osl_Map< Key, T, Alloc, Compare >::find (const key_type & x) [inline]`

Finds an element whose key is x

7.55.4.15 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> void Osl_Map< Key, T, Alloc, Compare >::insert (const value_type * first, const value_type * last) [inline]`

Inserts the range [first,last) into the map

7.55.4.16 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> iterator Osl_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

7.55.4.17 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>>`
`pair_iterator_bool Osl_Map< Key, T, Alloc, Compare >::insert (const value_type & x)`
`[inline]`

Inserts x into the map

7.55.4.18 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>>`
`key_compare Osl_Map< Key, T, Alloc, Compare >::key_comp () const [inline]`

Returns the key compare object used by the map

7.55.4.19 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>>`
`const_iterator Osl_Map< Key, T, Alloc, Compare >::lower_bound (const key_type &`
`x) const [inline]`

Finds the first element whose key is not less than x

7.55.4.20 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>>` `iterator`
`Osl_Map< Key, T, Alloc, Compare >::lower_bound (const key_type & x) [inline]`

Finds the first element whose key is not less than x

7.55.4.21 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>>` `size_type`
`Osl_Map< Key, T, Alloc, Compare >::max_size () const [inline]`

Returns the maximum possible size of the map

7.55.4.22 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>>` `self&`
`Osl_Map< Key, T, Alloc, Compare >::operator= (const self & x) [inline]`

Osl_Map assignment operator

7.55.4.23]

`template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>>` `T& Osl_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().

7.55.4.24 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>>` `size_type`
`Osl_Map< Key, T, Alloc, Compare >::size () const [inline]`

Returns the size of the map

7.55.4.25 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>>`
`const_iterator Osl_Map< Key, T, Alloc, Compare >::upper_bound (const key_type &`
`x) const [inline]`

Finds the first element whose key is not greater than x

7.55.4.26 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>>` `iterator`
`Osl_Map< Key, T, Alloc, Compare >::upper_bound (const key_type & x) [inline]`

Finds the first element whose key is not greater than x

7.55.4.27 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>>`
`value_compare Osl_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](#)

7.56 `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
```

7.56.1 Constructor & Destructor Documentation

7.56.1.1 `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]

7.56.2 Member Function Documentation

7.56.2.1 `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]

7.56.3 Friends And Related Function Documentation

7.56.3.1 `template<class Key, class T, class Alloc, class Compare = OscL_Less<Key>> friend class OscL_Map< Key, T, Alloc, Compare >` [friend]

7.56.4 Field Documentation

7.56.4.1 `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:



7.56 `OscL_Map< Key, T, Alloc, Compare >::value_compare` Class Reference

- [oscl_map.h](#)

7.57 Osl_MTLinked_List< LLClass, Alloc, TheLock > Class Template Reference

```
#include <osl_linked_list.h>
```

Public Methods

- [Osl_MTLinked_List \(\)](#)
- [~Osl_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

- [Osl_Linked_List< LLClass, Alloc > the_list](#)

7.57.1 Detailed Description

template<class LLClass, class Alloc, class TheLock> class Osl_MTLinked_List< LLClass, Alloc, TheLock >

Osl_MTLinked_List is a multi-threaded version of the LinkedList. It has mutex protection to allow access by different threads.

7.57.2 Constructor & Destructor Documentation

7.57.2.1 template<class LLClass, class Alloc, class TheLock> Osl_MTLinked_List< LLClass, Alloc, TheLock >::Osl_MTLinked_List () [inline]

Constructor for Osl_MTLinked_List

7.57.2.2 template<class LLClass, class Alloc, class TheLock> Osl_MTLinked_List< LLClass, Alloc, TheLock >::~~Osl_MTLinked_List () [inline]

Destructor for Osl_MTLinked_List

7.57.3 Member Function Documentation

7.57.3.1 `template<class LLClass, class Alloc, class TheLock> int32 Osci_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 creates the list and adds the element as first element of list.

Parameters:

new_element the element to be added in the list.

Returns:

32-bit integer on success returns 1.

7.57.3.2 `template<class LLClass, class Alloc, class TheLock> int32 Osci_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 already exists then it adds element at start of list otherwise it creates the list and adds the element as first element of list.

Parameters:

new_element the element to be added in the list.

Returns:

32-bit integer on success returns 1.

7.57.3.3 `template<class LLClass, class Alloc, class TheLock> int32 Osci_MTLinked_List< LLClass, Alloc, TheLock >::dequeue_element (LLClass & element) [inline]`

7.57.3.4 `template<class LLClass, class Alloc, class TheLock> uint32 Osci_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.

7.57.3.5 `template<class LLClass, class Alloc, class TheLock> int32 Osci_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 returned.

Returns:

32-bit integer if data is found in the list it returns index otherwise it returns -1.

7.57.3.6 `template<class LLClass, class Alloc, class TheLock> int32 Osci_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.

7.57.3.7 `template<class LLClass, class Alloc, class TheLock> int32 Osci_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.

7.57.3.8 `template<class LLClass, class Alloc, class TheLock> int32 Osci_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.

7.57.3.9 `template<class LLClass, class Alloc, class TheLock> int32 Osci_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 found in the list returns 1 otherwise returns 0.

7.57.4 Field Documentation

7.57.4.1 `template<class LLClass, class Alloc, class TheLock> Osci_Linked_List<LLClass, Alloc> Osci_MTLinked_List< LLClass, Alloc, TheLock >::the_list [protected]`

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



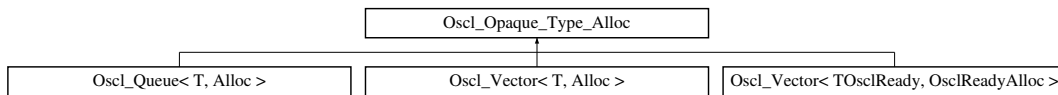
7.57 `OscL_MTLINKED_List< LLClass, Alloc, TheLock >` Class Template Reference

- [oscl_linked_list.h](#)

7.58 Osl_Opaque_Type_Alloc Class Reference

```
#include <osl_opaque_type.h>
```

Inheritance diagram for Osl_Opaque_Type_Alloc::



Public Methods

- virtual void `construct` (`OslAny *p`, const `OslAny *init_val`)=0
- virtual void `destroy` (`OslAny *p`)=0
- virtual `OslAny * allocate` (const uint32 size)=0
- virtual void `deallocate` (`OslAny *p`)=0

7.58.1 Detailed Description

This class combines opaque type operations with memory allocation operations.

7.58.2 Member Function Documentation

7.58.2.1 virtual `OslAny*` `Osl_Opaque_Type_Alloc::allocate` (const uint32 *size*) [pure virtual]

Allocate "size" bytes

7.58.2.2 virtual void `Osl_Opaque_Type_Alloc::construct` (`OslAny * p`, const `OslAny * init_val`) [pure virtual]

Construct element at p using element at init_val as the initial value. Both pointers must be non-NULL.

7.58.2.3 virtual void `Osl_Opaque_Type_Alloc::deallocate` (`OslAny * p`) [pure virtual]

Deallocate memory previously allocated with "allocate"

7.58.2.4 virtual void `Osl_Opaque_Type_Alloc::destroy` (`OslAny * p`) [pure virtual]

Destroy element at p.

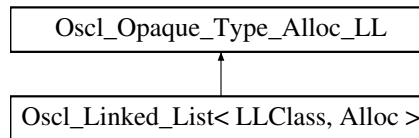
The documentation for this class was generated from the following file:

- [osl_opaque_type.h](#)

7.59 Oslc_Opaque_Type_Alloc_LL Class Reference

```
#include <oslc_opaque_type.h>
```

Inheritance diagram for Oslc_Opaque_Type_Alloc_LL::



Public Methods

- virtual void `construct` (`OslcAny *p`, const `OslcAny *init_val`)=0
- virtual void `destroy` (`OslcAny *p`)=0
- virtual `OslcAny *allocate` (const uint32 size)=0
- virtual void `deallocate` (`OslcAny *p`)=0
- virtual `OslcAny *get_next` (const `OslcAny *elem`) const=0
- virtual void `set_next` (`OslcAny *elem`, const `OslcAny *nextelem`)=0
- virtual void `get_data` (`OslcAny *elem`, `OslcAny *data_val`)=0
- virtual bool `compare_data` (const `OslcAny *elem`, const `OslcAny *data_val`) const=0

7.59.1 Detailed Description

This class combines opaque type operations with memory allocation operations and linked list support

7.59.2 Member Function Documentation

7.59.2.1 virtual `OslcAny*` `Oslc_Opaque_Type_Alloc_LL::allocate` (const uint32 *size*) [pure virtual]

Allocate "size" bytes

7.59.2.2 virtual bool `Oslc_Opaque_Type_Alloc_LL::compare_data` (const `OslcAny * elem`, const `OslcAny * data_val`) const [pure virtual]

Compare data.

7.59.2.3 virtual void `Oslc_Opaque_Type_Alloc_LL::construct` (`OslcAny * p`, const `OslcAny * init_val`) [pure virtual]

Construct element at p using element at init_val as the initial value. Both pointers must be non-NULL.

7.59.2.4 virtual void `Oslc_Opaque_Type_Alloc_LL::deallocate` (`OslcAny * p`) [pure virtual]

Deallocate memory previously allocated with "allocate"

7.59.2.5 virtual void Osci_Opaque_Type_Alloc_LL::destroy (OsciAny * *p*) [pure virtual]

Destroy element at *p*.

7.59.2.6 virtual void Osci_Opaque_Type_Alloc_LL::get_data (OsciAny * *elem*, OsciAny * *data_val*) [pure virtual]

Get data

7.59.2.7 virtual OsciAny* Osci_Opaque_Type_Alloc_LL::get_next (const OsciAny * *elem*) const [pure virtual]

Get next element in linked list.

7.59.2.8 virtual void Osci_Opaque_Type_Alloc_LL::set_next (OsciAny * *elem*, const OsciAny * *nextelem*) [pure virtual]

Set next element in linked list.

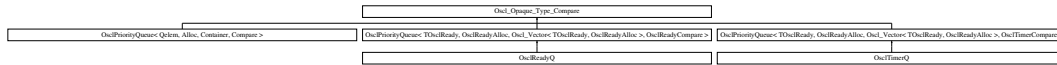
The documentation for this class was generated from the following file:

- [osci_opaque_type.h](#)

7.60 Osci_Opaque_Type_Compare Class Reference

```
#include <osci_opaque_type.h>
```

Inheritance diagram for Osci_Opaque_Type_Compare::



Public Methods

- virtual void `swap (OsciAny *a, const OsciAny *b)=0`
- virtual int `compare_LT (OsciAny *a, OsciAny *b) const=0`
- virtual int `compare_EQ (const OsciAny *a, const OsciAny *b) const=0`

7.60.1 Detailed Description

Opaque type operations with swap & comparisons.

7.60.2 Member Function Documentation

7.60.2.1 virtual int Osci_Opaque_Type_Compare::compare_EQ (const OsciAny * a, const OsciAny * b) const [pure virtual]

Return a==b.

Implemented in `OsciPriorityQueue< Qelem, Alloc, Container, Compare >`, `OsciPriorityQueue< TOsciReady, OsciReadyAlloc, Osci_Vector< TOsciReady, OsciReadyAlloc >, OsciReadyCompare >`, and `OsciPriorityQueue< TOsciReady, OsciReadyAlloc, Osci_Vector< TOsciReady, OsciReadyAlloc >, OsciTimerCompare >`.

7.60.2.2 virtual int Osci_Opaque_Type_Compare::compare_LT (OsciAny * a, OsciAny * b) const [pure virtual]

Return a<b.

Implemented in `OsciPriorityQueue< Qelem, Alloc, Container, Compare >`, `OsciPriorityQueue< TOsciReady, OsciReadyAlloc, Osci_Vector< TOsciReady, OsciReadyAlloc >, OsciReadyCompare >`, and `OsciPriorityQueue< TOsciReady, OsciReadyAlloc, Osci_Vector< TOsciReady, OsciReadyAlloc >, OsciTimerCompare >`.

7.60.2.3 virtual void Osci_Opaque_Type_Compare::swap (OsciAny * a, const OsciAny * b) [pure virtual]

Swap element at "a" with element at "b". Both pointers must be non-NULL.

Implemented in `OsciPriorityQueue< Qelem, Alloc, Container, Compare >`, `OsciPriorityQueue< TOsciReady, OsciReadyAlloc, Osci_Vector< TOsciReady, OsciReadyAlloc >, OsciReadyCompare >`, and `OsciPriorityQueue< TOsciReady, OsciReadyAlloc, Osci_Vector< TOsciReady, OsciReadyAlloc >, OsciTimerCompare >`.

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

- [oslc_opaque_type.h](#)

7.61 Osl_Pair< T1, T2 > Struct Template Reference

```
#include <oscl_tree.h>
```

Public Methods

- [Osl_Pair \(\)](#)
- [Osl_Pair \(const T1 &a, const T2 &b\)](#)

Data Fields

- T1 [first](#)
- T2 [second](#)

```
template<class T1, class T2> struct Osl_Pair< T1, T2 >
```

7.61.1 Constructor & Destructor Documentation

7.61.1.1 `template<class T1, class T2> Osl_Pair< T1, T2 >::Osl_Pair () [inline]`

7.61.1.2 `template<class T1, class T2> Osl_Pair< T1, T2 >::Osl_Pair (const T1 & a, const T2 & b) [inline]`

7.61.2 Field Documentation

7.61.2.1 `template<class T1, class T2> T1 Osl_Pair< T1, T2 >::first`

7.61.2.2 `template<class T1, class T2> T2 Osl_Pair< T1, T2 >::second`

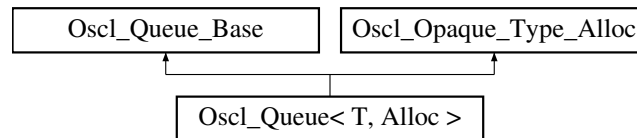
The documentation for this struct was generated from the following file:

- [oscl_tree.h](#)

7.62 `OscQueue< T, Alloc >` Class Template Reference

```
#include <oscl_queue.h>
```

Inheritance diagram for `OscQueue< 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

- `OscQueue` ()
- `OscQueue` (`uint32 n`)
- virtual `~OscQueue` ()
- void `push` (`const T &x`)
- [reference front](#) ()
- [const_reference front](#) () const
- void `pop` ()
- [reference back](#) ()
- [const_reference back](#) () const
- void `clear` ()

7.62.1 Detailed Description

```
template<class T, class Alloc> class OscQueue< T, Alloc >
```

`OscQueue` Class. A subset of `STL::Queue` methods. `OscQueue` 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.

7.62.2 Member Typedef Documentation

7.62.2.1 `template<class T, class Alloc> typedef const T& Osl_Queue< T, Alloc >::const_reference`

7.62.2.2 `template<class T, class Alloc> typedef T* Osl_Queue< T, Alloc >::pointer`

7.62.2.3 `template<class T, class Alloc> typedef T& Osl_Queue< T, Alloc >::reference`

7.62.2.4 `template<class T, class Alloc> typedef uint32 Osl_Queue< T, Alloc >::size_type`

7.62.2.5 `template<class T, class Alloc> typedef T Osl_Queue< T, Alloc >::value_type`

7.62.3 Constructor & Destructor Documentation

7.62.3.1 `template<class T, class Alloc> Osl_Queue< T, Alloc >::Osl_Queue () [inline]`

Creates an empty queue.

7.62.3.2 `template<class T, class Alloc> Osl_Queue< T, Alloc >::Osl_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.

7.62.3.3 `template<class T, class Alloc> virtual Osl_Queue< T, Alloc >::~~Osl_Queue () [inline, virtual]`

The destructor.

7.62.4 Member Function Documentation

7.62.4.1 `template<class T, class Alloc> const_reference Osl_Queue< T, Alloc >::back () const [inline]`

Returns the last element (const)

7.62.4.2 `template<class T, class Alloc> reference Osl_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)

7.62.4.3 `template<class T, class Alloc> void Osl_Queue< T, Alloc >::clear () [inline]`

Removes all elements.

Reimplemented from [Osl_Queue_Base](#).

7.62.4.4 `template<class T, class Alloc> const_reference Osl_Queue< T, Alloc >::front () const`
[inline]

Returns the first element (const)

7.62.4.5 `template<class T, class Alloc> reference Osl_Queue< T, Alloc >::front ()` [inline]

Returns the first element.

Reimplemented from [Osl_Queue_Base](#).

7.62.4.6 `template<class T, class Alloc> void Osl_Queue< T, Alloc >::pop ()` [inline]

Removes the first element

Reimplemented from [Osl_Queue_Base](#).

7.62.4.7 `template<class T, class Alloc> void Osl_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

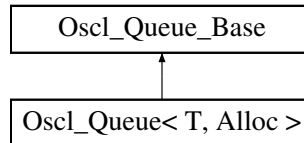
The documentation for this class was generated from the following file:

- [oscl_queue.h](#)

7.63 Osl_Queue_Base Class Reference

```
#include <osl_queue.h>
```

Inheritance diagram for Osl_Queue_Base::



Public Methods

- uint32 [size](#) () const
- uint32 [capacity](#) () const
- bool [empty](#) () const
- OSL_IMPORT_REF void [reserve](#) (uint32 n)

Protected Methods

- OSL_IMPORT_REF void [construct](#) (Osl_Opaque_Type_Alloc *aType)
- OSL_IMPORT_REF void [construct](#) (Osl_Opaque_Type_Alloc *aType, uint32 n)
- virtual [~Osl_Queue_Base](#) ()
- OSL_IMPORT_REF void [destroy](#) ()
- OSL_IMPORT_REF void [push](#) (const OslAny *x)
- OSL_IMPORT_REF void [pop](#) ()
- OSL_IMPORT_REF void [clear](#) ()

Protected Attributes

- uint32 [numelems](#)
- uint32 [bufsize](#)
- OslAny * [elems](#)
- uint32 [sizeof_T](#)
- uint32 [ifront](#)
- uint32 [irear](#)

7.63.1 Detailed Description

Osl_Queue_Base is a non-templated base class for [Osl_Queue](#). The purpose of this base class is to avoid large inline routines in the [Osl_Queue](#) implementation. This class is not intended for direct instantiation except by [Osl_Queue](#).

7.63.2 Constructor & Destructor Documentation

7.63.2.1 virtual Osl_Queue_Base::~~Osl_Queue_Base () [inline, protected, virtual]

The destructor.

7.63.3 Member Function Documentation

7.63.3.1 `uint32 Osl_Queue_Base::capacity () const` [inline]

Returns the allocated memory of the queue.

7.63.3.2 `OSCL_IMPORT_REF void Osl_Queue_Base::clear ()` [protected]

Removes all elements.

Reimplemented in [Osl_Queue< T, Alloc >](#).

7.63.3.3 `OSCL_IMPORT_REF void Osl_Queue_Base::construct (Osl_Opaque_Type_Alloc * aType, uint32 n)` [protected]

7.63.3.4 `OSCL_IMPORT_REF void Osl_Queue_Base::construct (Osl_Opaque_Type_Alloc * aType)` [protected]

7.63.3.5 `OSCL_IMPORT_REF void Osl_Queue_Base::destroy ()` [protected]

Like an explicit destructor call.

7.63.3.6 `bool Osl_Queue_Base::empty () const` [inline]

True if there are no elements in the queue

7.63.3.7 `OSCL_IMPORT_REF void Osl_Queue_Base::pop ()` [protected]

Removes the first element

Reimplemented in [Osl_Queue< T, Alloc >](#).

7.63.3.8 `OSCL_IMPORT_REF void Osl_Queue_Base::push (const OslAny * 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

7.63.3.9 `OSCL_IMPORT_REF void Osl_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

7.63.3.10 `uint32 Osl_Queue_Base::size () const` [inline]

Returns the size of the queue.

7.63.4 Field Documentation**7.63.4.1** `uint32 Osl_Queue_Base::bufsize` [protected]**7.63.4.2** `OslAny* Osl_Queue_Base::elems` [protected]**7.63.4.3** `uint32 Osl_Queue_Base::ifront` [protected]**7.63.4.4** `uint32 Osl_Queue_Base::irear` [protected]**7.63.4.5** `uint32 Osl_Queue_Base::numelems` [protected]**7.63.4.6** `uint32 Osl_Queue_Base::sizeof_T` [protected]

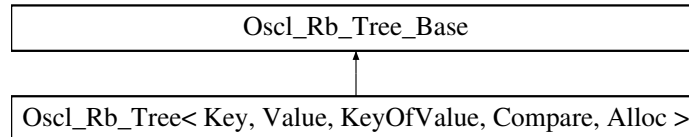
The documentation for this class was generated from the following file:

- [oscl_queue.h](#)

7.64 `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`)



7.64 `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 >`

7.64.1 Member Typedef Documentation

7.64.1.1 `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`

7.64.1.2 `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`

7.64.1.3 `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`

7.64.1.4 `template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef int32 OscL_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::difference_type`

7.64.1.5 `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`

7.64.1.6 `template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef Key OscL_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::key_type`

7.64.1.7 `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`

7.64.1.8 `template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef value_type* OscL_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::pointer`

7.64.1.9 `template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef value_type& OscL_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::reference`

7.64.1.10 `template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef uint32 OscL_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::size_type`

7.64.1.11 `template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef Value OscL_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::value_type`

7.64.2 Constructor & Destructor Documentation

7.64.2.1 `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]`

7.64.2.2 `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]`

7.64.2.3 `template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> OscL_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::~OscL_Rb_Tree () [inline]`



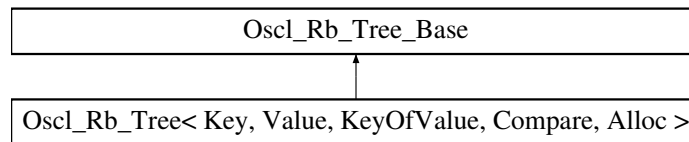
7.64 `OscL_Rb_Tree`< `Key`, `Value`, `KeyOfValue`, `Compare`, `Alloc` > Class Template Reference

- [oscl_tree.h](#)

7.65 Osl_Rb_Tree_Base Class Reference

```
#include <oscl_tree.h>
```

Inheritance diagram for Osl_Rb_Tree_Base::



Public Types

- typedef Osl_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)

7.65.1 Member Typedef Documentation

7.65.1.1 typedef Osl_Rb_Tree_Node_Base::base_link_type Osl_Rb_Tree_Base::base_link_type

7.65.2 Member Function Documentation

7.65.2.1 OSCL_IMPORT_REF void Osl_Rb_Tree_Base::rebalance ([base_link_type](#) x, [base_link_type](#) &root)

7.65.2.2 OSCL_IMPORT_REF [base_link_type](#) Osl_Rb_Tree_Base::rebalance_for_erase ([base_link_type](#) z, [base_link_type](#) &root, [base_link_type](#) &leftmost, [base_link_type](#) &rightmost)

7.65.2.3 OSCL_IMPORT_REF void Osl_Rb_Tree_Base::rotate_left ([base_link_type](#) x, [base_link_type](#) &root)

7.65.2.4 OSCL_IMPORT_REF void Osl_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](#)

7.66 `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 \rightarrow () 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 Oslc_Rb_Tree_Const_Iterator< Value >

7.66.1 Member Typedef Documentation

7.66.1.1 template<class Value> typedef [Oslc_Rb_Tree_Node_Base*](#)
Oslc_Rb_Tree_Const_Iterator< Value >::base_link_type

7.66.1.2 template<class Value> typedef Oslc_Rb_Tree_Const_Iterator<Value>
Oslc_Rb_Tree_Const_Iterator< Value >::const_iterator

7.66.1.3 template<class Value> typedef [Oslc_Rb_Tree_Node](#)<Value>*
Oslc_Rb_Tree_Const_Iterator< Value >::link_type

7.66.1.4 template<class Value> typedef const [value_type*](#) Oslc_Rb_Tree_Const_Iterator< Value
>::pointer

7.66.1.5 template<class Value> typedef const [value_type](#)& Oslc_Rb_Tree_Const_Iterator< Value
>::reference

7.66.1.6 template<class Value> typedef Oslc_Rb_Tree_Const_Iterator<Value>
Oslc_Rb_Tree_Const_Iterator< Value >::self

7.66.1.7 template<class Value> typedef Value Oslc_Rb_Tree_Const_Iterator< Value
>::value_type

7.66.2 Constructor & Destructor Documentation

7.66.2.1 template<class Value> Oslc_Rb_Tree_Const_Iterator< Value
>::Oslc_Rb_Tree_Const_Iterator() [inline]

7.66.2.2 template<class Value> Oslc_Rb_Tree_Const_Iterator< Value
>::Oslc_Rb_Tree_Const_Iterator([link_type](#) x) [inline]

7.66.2.3 template<class Value> Oslc_Rb_Tree_Const_Iterator< Value
>::Oslc_Rb_Tree_Const_Iterator(const [const_iterator](#) & it) [inline]

7.66.3 Member Function Documentation

7.66.3.1 template<class Value> [reference](#) Oslc_Rb_Tree_Const_Iterator< Value >::operator * ()
const [inline]

7.66.3.2 template<class Value> bool Oslc_Rb_Tree_Const_Iterator< Value >::operator!= (const
[self](#) & x) [inline]

7.66.3.3 template<class Value> [self](#) Oslc_Rb_Tree_Const_Iterator< Value >::operator++ (int)
[inline]

7.66.3.4 template<class Value> [self](#)& Oslc_Rb_Tree_Const_Iterator< Value >::operator++ ()
[inline]

7.66.3.5 template<class Value> [self](#) Oslc_Rb_Tree_Const_Iterator< Value >::operator-- (int)
[inline]

7.66.3.6 template<class Value> [self](#)& Oslc_Rb_Tree_Const_Iterator< Value >::operator-- ()
[inline]

- [oscl_tree.h](#)

7.67 Oslc_Rb_Tree_Iterator< Value > Struct Template Reference

```
#include <oslc_tree.h>
```

Public Types

- typedef Value [value_type](#)
- typedef [value_type](#) & [reference](#)
- typedef [value_type](#) * [pointer](#)
- typedef Oslc_Rb_Tree_Iterator< Value > [iterator](#)
- typedef Oslc_Rb_Tree_Iterator< Value > [self](#)
- typedef [Oslc_Rb_Tree_Node_Base](#) * [base_link_type](#)
- typedef [Oslc_Rb_Tree_Node](#)< Value > * [link_type](#)

Public Methods

- [Oslc_Rb_Tree_Iterator](#) ()
- [Oslc_Rb_Tree_Iterator](#) ([link_type](#) x)
- [Oslc_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 Oslc_Rb_Tree_Iterator< Value >

7.67.1 Member Typedef Documentation

7.67.1.1 `template<class Value> typedef Oslc_Rb_Tree_Node_Base* Oslc_Rb_Tree_Iterator< Value >::base_link_type`

7.67.1.2 `template<class Value> typedef Oslc_Rb_Tree_Iterator<Value> Oslc_Rb_Tree_Iterator< Value >::iterator`

7.67.1.3 `template<class Value> typedef Oslc_Rb_Tree_Node<Value>* Oslc_Rb_Tree_Iterator< Value >::link_type`

7.67.1.4 `template<class Value> typedef value_type* Oslc_Rb_Tree_Iterator< Value >::pointer`

7.67.1.5 `template<class Value> typedef value_type& Oslc_Rb_Tree_Iterator< Value >::reference`

7.67.1.6 `template<class Value> typedef Oslc_Rb_Tree_Iterator<Value> Oslc_Rb_Tree_Iterator< Value >::self`

7.67.1.7 `template<class Value> typedef Value Oslc_Rb_Tree_Iterator< Value >::value_type`

7.67.2 Constructor & Destructor Documentation

7.67.2.1 `template<class Value> Oslc_Rb_Tree_Iterator< Value >::Oslc_Rb_Tree_Iterator ()` `[inline]`

7.67.2.2 `template<class Value> Oslc_Rb_Tree_Iterator< Value >::Oslc_Rb_Tree_Iterator (link_type x)` `[inline]`

7.67.2.3 `template<class Value> Oslc_Rb_Tree_Iterator< Value >::Oslc_Rb_Tree_Iterator (const iterator & it)` `[inline]`

7.67.3 Member Function Documentation

7.67.3.1 `template<class Value> reference Oslc_Rb_Tree_Iterator< Value >::operator * () const` `[inline]`

7.67.3.2 `template<class Value> bool Oslc_Rb_Tree_Iterator< Value >::operator!= (const self & x)` `[inline]`

7.67.3.3 `template<class Value> self Oslc_Rb_Tree_Iterator< Value >::operator++ (int)` `[inline]`

7.67.3.4 `template<class Value> self& Oslc_Rb_Tree_Iterator< Value >::operator++ ()` `[inline]`

7.67.3.5 `template<class Value> self Oslc_Rb_Tree_Iterator< Value >::operator-- (int)` `[inline]`

7.67.3.6 `template<class Value> self& Oslc_Rb_Tree_Iterator< Value >::operator-- ()` `[inline]`

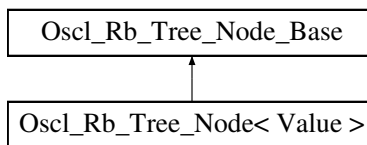
7.67.3.7 `template<class Value> pointer Oslc_Rb_Tree_Iterator< Value >::operator → () const` `[inline]`

- [oscl_tree.h](#)

7.68 Osci_Rb_Tree_Node< Value > Struct Template Reference

```
#include <osci_tree.h>
```

Inheritance diagram for Osci_Rb_Tree_Node< Value >::



Public Types

- typedef Value [value_type](#)
- typedef Osci_Rb_Tree_Node< Value > * [link_type](#)

Data Fields

- [value_type](#) value

```
template<class Value> struct Osci_Rb_Tree_Node< Value >
```

7.68.1 Member Typedef Documentation

7.68.1.1 `template<class Value> typedef Osci_Rb_Tree_Node<Value>* Osci_Rb_Tree_Node< Value >::link_type`

7.68.1.2 `template<class Value> typedef Value Osci_Rb_Tree_Node< Value >::value_type`

7.68.2 Field Documentation

7.68.2.1 `template<class Value> value_type Osci_Rb_Tree_Node< Value >::value`

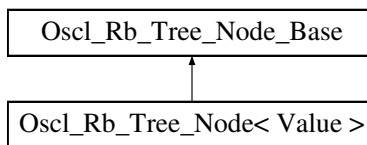
The documentation for this struct was generated from the following file:

- [osci_tree.h](#)

7.69 Osl_Rb_Tree_Node_Base Struct Reference

```
#include <osl_tree.h>
```

Inheritance diagram for Osl_Rb_Tree_Node_Base::



Public Types

- typedef Osl_Rb_Tree_Node_Base * [base_link_type](#)
- typedef enum [Osl_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](#)

7.69.1 Member Typedef Documentation

7.69.1.1 typedef Osl_Rb_Tree_Node_Base* Osl_Rb_Tree_Node_Base::base_link_type

7.69.1.2 typedef enum [Osl_Rb_Tree_Node_Base::RedBl](#) Osl_Rb_Tree_Node_Base::color_type

7.69.2 Member Enumeration Documentation

7.69.2.1 enum Osl_Rb_Tree_Node_Base::RedBl

Enumeration values:

red

black

7.69.3 Member Function Documentation

7.69.3.1 [base_link_type](#) `Oscl_Rb_Tree_Node_Base::maximum(base_link_type x)` [`inline`, `static`]

7.69.3.2 [base_link_type](#) `Oscl_Rb_Tree_Node_Base::minimum(base_link_type x)` [`inline`, `static`]

7.69.4 Field Documentation

7.69.4.1 [color_type](#) `Oscl_Rb_Tree_Node_Base::color`

7.69.4.2 [base_link_type](#) `Oscl_Rb_Tree_Node_Base::left`

7.69.4.3 [base_link_type](#) `Oscl_Rb_Tree_Node_Base::parent`

7.69.4.4 [base_link_type](#) `Oscl_Rb_Tree_Node_Base::right`

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

- [oscl_tree.h](#)

7.70 Osl_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 Osl_Select1st< V, U >
```

7.70.1 Member Function Documentation

7.70.1.1 `template<class V, class U> const U& Osl_Select1st< V, U >::operator() (const V & x)`
`const [inline]`

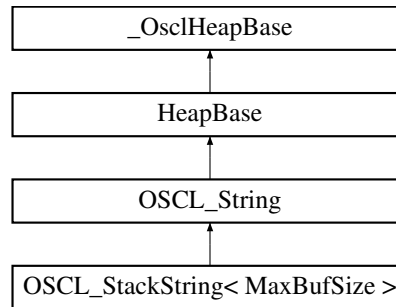
The documentation for this struct was generated from the following file:

- [oscl_map.h](#)

7.71 OSCL_StackString< MaxBufSize > Class Template Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL_StackString< MaxBufSize >::



Public Types

- typedef OSCL_String::chartype [chartype](#)
- typedef TOSCL_StringOp [optype](#)
- typedef OSCL_wString::chartype [other_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)
- void [set](#) (const other_chartype *buf, optype op)
- void [set](#) (const other_chartype *buf, uint32 length, optype op)

Friends

- class [OSCL_String](#)

7.71.1 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.

7.71.2 Member Typedef Documentation

7.71.2.1 `template<uint32 MaxBufSize> typedef OSCL_String::chartype OSCL_StackString< MaxBufSize >::chartype`

Reimplemented from [OSCL_String](#).

7.71.2.2 `template<uint32 MaxBufSize> typedef OSCL_StringOp OSCL_StackString< MaxBufSize >::optype`

7.71.2.3 `template<uint32 MaxBufSize> typedef OSCL_wString::chartype OSCL_StackString< MaxBufSize >::other_chartype`

7.71.3 Friends And Related Function Documentation

7.71.3.1 `template<uint32 MaxBufSize> friend class OSCL_String [friend]`

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

- [oscl_string_containers.h](#)

7.72 oscl_stat_buf Struct Reference

```
#include <oscl_file_dir_utils.h>
```

Data Fields

- uint32 [mode](#)
- uint32 [perms](#)

7.72.1 Field Documentation

7.72.1.1 uint32 oscl_stat_buf::mode

7.72.1.2 uint32 oscl_stat_buf::perms

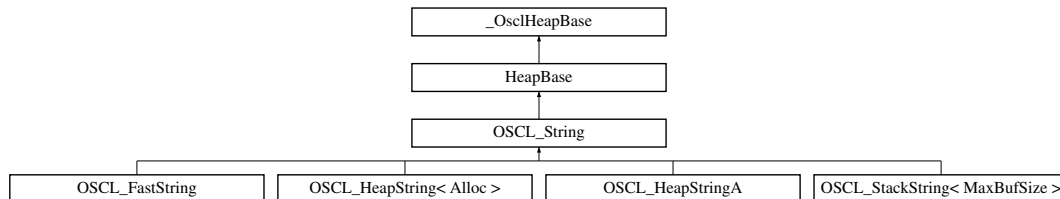
The documentation for this struct was generated from the following file:

- [oscl_file_dir_utils.h](#)

7.73 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 bool [is_writable](#) () const
- virtual [chartype](#) * [get_str](#) () const=0
- OSCL_String & [operator=](#) (const OSCL_String &src)
- OSCL_String & [operator=](#) (const [chartype](#) *cstr)
- OSCL_String & [operator+=](#) (const OSCL_String &src)
- OSCL_String & [operator+=](#) (const [chartype](#) *cstr)
- OSCL_String & [operator+=](#) (const [chartype](#) c)
- bool [operator==](#) (const OSCL_String &src) const
- bool [operator!=](#) (const OSCL_String &src) const
- bool [operator<](#) (const OSCL_String &src) const
- bool [operator<=](#) (const OSCL_String &src) const
- bool [operator>](#) (const OSCL_String &src) const
- bool [operator>=](#) (const OSCL_String &src) const
- bool [operator==](#) (const [chartype](#) *cstr) const
- [chartype](#) [operator\[\]](#) (uint32 index) const
- virtual [chartype](#) [read](#) (uint32 index) const
- virtual uint32 [setrep_to_char](#) (const [oscl_wchar](#) *src, uint32 len, [TOSCL_StringOp](#) op, [OscL_DefAlloc](#) *aAlloc)
- virtual int8 [hash](#) () const
- virtual void [write](#) (uint32 index, [chartype](#) c)
- virtual void [write](#) (uint32 offset, uint32 length, const [chartype](#) *buf)

Protected Methods

- [OSCL_String \(\)](#)
- virtual [~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](#)

7.73.1 Detailed Description

A common base class for string classes with "char" character format

7.73.2 Member Typedef Documentation

7.73.2.1 typedef char OSCL_String::chartype

Reimplemented in [OSCL_HeapString< Alloc >](#), [OSCL_HeapStringA](#), [OSCL_StackString< MaxBufSize >](#), [OSCL_FastString](#), and [OSCL_HeapString< OsciMemAllocator >](#).

7.73.3 Constructor & Destructor Documentation

7.73.3.1 OSCL_String::OSCL_String () [protected]

7.73.3.2 virtual OSCL_String::~~OSCL_String () [protected, virtual]

7.73.4 Member Function Documentation

7.73.4.1 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.

7.73.4.2 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.

7.73.4.3 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< OsciMemAllocator >](#).

7.73.4.4 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< OsciMemAllocator >](#).

7.73.4.5 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< OsciMemAllocator >](#).

7.73.4.6 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< OsciMemAllocator >](#).

7.73.4.7 virtual int8 OSCL_String::hash () [virtual]

This function performs a hash operation on the string. If the string is not writable, the function leaves.

7.73.4.8 virtual bool OSCL_String::is_writable () [virtual]

This function returns true if the string is writable.

7.73.4.9 bool OSCL_String::operator!= (const OSCL_String & src) const**7.73.4.10 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.

7.73.4.11 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

7.73.4.12 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.

7.73.4.13 `bool OSCL_String::operator< (const OSCL_String & src) const`

7.73.4.14 `bool OSCL_String::operator<= (const OSCL_String & src) const`

7.73.4.15 `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< OsciMemAllocator >](#).

7.73.4.16 `OSCL_String& OSCL_String::operator= (const OSCL_String & src)`

Assignment operator

Reimplemented in [OSCL_HeapString< Alloc >](#), [OSCL_HeapStringA](#), [OSCL_StackString< MaxBufSize >](#), and [OSCL_HeapString< OsciMemAllocator >](#).

7.73.4.17 `bool OSCL_String::operator== (const chartype * cstr) const`

Comparison operator

am: null-terminated string

7.73.4.18 `bool OSCL_String::operator== (const OSCL_String & src) const`

Comparison operators

7.73.4.19 `bool OSCL_String::operator> (const OSCL_String & src) const`

7.73.4.20 `bool OSCL_String::operator>= (const OSCL_String & src) const`

7.73.4.21 `]`

`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.

7.73.4.22 `virtual 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.

7.73.4.23 `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.

7.73.4.24 `virtual void OSCL_String::set_rep (const OSCL_String & src)` [protected, pure virtual]

Set string representation to input string.

7.73.4.25 `virtual void OSCL_String::set_rep (const chartype * cstr)` [protected, pure virtual]

Set string representation to input null-terminated string.

7.73.4.26 `virtual uint32 OSCL_String::setrep_to_char (const oscl_wchar * src, uint32 len, TOSCL_StringOp op, Oscl_DefAlloc * aAlloc)` [virtual]

This function allocates a temp storage for performing one of the following operations based on TOSCL_StringOp

- compress src string from `oscl_wchar` to utf8.
- convert src string from `oscl_wchar` to utf8. call parent `set_rep()` to copy resulting string.

Parameters:

- src*: reference input string
- len*: length of string to operate on
- op*: type operation mentioned above
- aAlloc*: optional, memory allocator if available

Returns:

length of compressed or converted string exclude terminated '\0'.

7.73.4.27 `virtual 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.

7.73.4.28 `virtual 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.

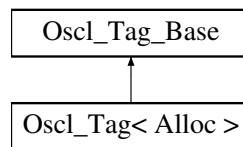
The documentation for this class was generated from the following file:

- [oscl_string.h](#)

7.74 Oslc_Tag< Alloc > Struct Template Reference

```
#include <oscl_tagtree.h>
```

Inheritance diagram for Oslc_Tag< Alloc >::



Public Methods

- [Oslc_Tag](#) (const Oslc_Tag< Alloc > &t)
- [Oslc_Tag](#) (const [tag_base_type](#) &t)
- [~Oslc_Tag](#) ()
- [bool operator<](#) (const Oslc_Tag< Alloc > &x) const

Data Fields

- [Oslc_TAlloc< tag_base_unit, Alloc > tagAllocator](#)
- [tag_base_type tag](#)

```
template<class Alloc> struct Oslc_Tag< Alloc >
```

7.74.1 Constructor & Destructor Documentation

7.74.1.1 `template<class Alloc> Oslc_Tag< Alloc >::Oslc_Tag (const Oslc_Tag< Alloc > & t)`
[inline]

7.74.1.2 `template<class Alloc> Oslc_Tag< Alloc >::Oslc_Tag (const tag_base_type & t)`
[inline]

7.74.1.3 `template<class Alloc> Oslc_Tag< Alloc >::~~Oslc_Tag ()` [inline]

7.74.2 Member Function Documentation

7.74.2.1 `template<class Alloc> bool Oslc_Tag< Alloc >::operator< (const Oslc_Tag< Alloc > & x) const` [inline]

7.74.3 Field Documentation

7.74.3.1 `template<class Alloc> tag_base_type Oslc_Tag< Alloc >::tag`

7.74.3.2 `template<class Alloc> Oslc_TAlloc<tag_base_unit, Alloc> Oslc_Tag< Alloc >::tagAllocator`

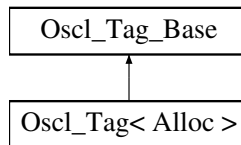
The documentation for this struct was generated from the following file:

- [osci_tagtree.h](#)

7.75 Oslc_Tag_Base Struct Reference

```
#include <oscl_tagtree.h>
```

Inheritance diagram for Oslc_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

7.75.1 Member Typedef Documentation

7.75.1.1 typedef uint32 Oslc_Tag_Base::size_type

7.75.1.2 typedef [tag_base_unit](#)* Oslc_Tag_Base::tag_base_type

7.75.1.3 typedef char Oslc_Tag_Base::tag_base_unit

7.75.2 Member Function Documentation

7.75.2.1 bool Oslc_Tag_Base::operator() (const [tag_base_type](#) & x, const [tag_base_type](#) & y) const
[inline]

7.75.2.2 OSCL_IMPORT_REF [tag_base_type](#) Oslc_Tag_Base::tag_ancestor ([tag_base_type](#) & dest, const [tag_base_type](#) & src) const

7.75.2.3 int32 Oslc_Tag_Base::tag_cmp (const [tag_base_type](#) & x, const [tag_base_type](#) & y) const
[inline]

7.75.2.4 [tag_base_type](#) Oslc_Tag_Base::tag_copy ([tag_base_type](#) & dest, const [tag_base_type](#) & src) const [inline]

7.75.2.5 OSCL_IMPORT_REF [size_type](#) Oslc_Tag_Base::tag_depth (const [tag_base_type](#) & t) const

7.75.2.6 [size_type](#) Oslc_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](#)

7.76 Oslc_TagTree< T, Alloc > Class Template Reference

```
#include <oscl_tagtree.h>
```

Public Types

- typedef [Oslc_Tag](#)< Alloc > [tag_type](#)
- typedef [tag_type](#)::[tag_base_type](#) [tag_base_type](#)
- typedef [Oslc_Vector](#)< [Node](#) *, Alloc > [children_type](#)
- typedef [Node](#) [node_type](#)
- typedef [node_type](#) * [node_ptr](#)
- typedef [Oslc_Map](#)< const [tag_base_type](#), [node_ptr](#), Alloc, [Oslc_Tag_Base](#) > [map_type](#)
- typedef [map_type](#)::[size_type](#) [size_type](#)
- typedef [map_type](#)::[value_type](#) [value_type](#)
- typedef [Oslc_Pair](#)< [iterator](#), bool > [pair_iterator_bool](#)

Public Methods

- [Oslc_TagTree](#) ([size_type](#) max_depth=0)
- [Oslc_TagTree](#) (const [Oslc_TagTree](#)< T, Alloc > &x)
- [Oslc_TagTree](#)< T, Alloc > & [operator=](#) (const [Oslc_TagTree](#)< T, Alloc > &x)
- [~Oslc_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

7.76.1 Detailed Description

```
template<class T, class Alloc> class Oslc_TagTree< T, Alloc >
```

Oslc_TagTree Class.

7.76.2 Member Typedef Documentation

- 7.76.2.1 `template<class T, class Alloc> typedef Oslc_Vector<Node*, Alloc> Oslc_TagTree< T, Alloc >::children_type`
- 7.76.2.2 `template<class T, class Alloc> typedef Oslc_Map<const tag_base_type, node_ptr, Alloc , Oslc_Tag_Base> Oslc_TagTree< T, Alloc >::map_type`
- 7.76.2.3 `template<class T, class Alloc> typedef node_type* Oslc_TagTree< T, Alloc >::node_ptr`
- 7.76.2.4 `template<class T, class Alloc> typedef Node Oslc_TagTree< T, Alloc >::node_type`
- 7.76.2.5 `template<class T, class Alloc> typedef Oslc_Pair<iterator, bool> Oslc_TagTree< T, Alloc >::pair_iterator_bool`
- 7.76.2.6 `template<class T, class Alloc> typedef map_type::size_type Oslc_TagTree< T, Alloc >::size_type`
- 7.76.2.7 `template<class T, class Alloc> typedef tag_type::tag_base_type Oslc_TagTree< T, Alloc >::tag_base_type`
- 7.76.2.8 `template<class T, class Alloc> typedef Oslc_Tag<Alloc> Oslc_TagTree< T, Alloc >::tag_type`
- 7.76.2.9 `template<class T, class Alloc> typedef map_type::value_type Oslc_TagTree< T, Alloc >::value_type`

7.76.3 Constructor & Destructor Documentation

- 7.76.3.1 `template<class T, class Alloc> Oslc_TagTree< T, Alloc >::Oslc_TagTree (size_type max_depth = 0) [inline]`

Creates a tag tree with only a root node with tag ""

- 7.76.3.2 `template<class T, class Alloc> Oslc_TagTree< T, Alloc >::Oslc_TagTree (const Oslc_TagTree< T, Alloc > & x) [inline]`

Copy constructor

- 7.76.3.3 `template<class T, class Alloc> Oslc_TagTree< T, Alloc >::~Oslc_TagTree () [inline]`

Destructor

7.76.4 Member Function Documentation

- 7.76.4.1 `template<class T, class Alloc> const_iterator Oslc_TagTree< T, Alloc >::begin () const [inline]`

Returns an iterator pointing to the first node in the tree.

7.76.4.2 `template<class T, class Alloc> iterator Oslc_TagTree< T, Alloc >::begin () [inline]`

Returns an iterator pointing to the first node in the tree.

7.76.4.3 `template<class T, class Alloc> void Oslc_TagTree< T, Alloc >::clear () [inline]`

Erases the entire tag tree.

7.76.4.4 `template<class T, class Alloc> size_type Oslc_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..

7.76.4.5 `template<class T, class Alloc> bool Oslc_TagTree< T, Alloc >::empty () const [inline]`

Returns true if tree size is 0

7.76.4.6 `template<class T, class Alloc> const_iterator Oslc_TagTree< T, Alloc >::end () const [inline]`

Returns a const iterator pointing to the end of the tree.

7.76.4.7 `template<class T, class Alloc> iterator Oslc_TagTree< T, Alloc >::end () [inline]`

Returns an iterator pointing to the end of the tree.

7.76.4.8 `template<class T, class Alloc> size_type Oslc_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

7.76.4.9 `template<class T, class Alloc> void Oslc_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

7.76.4.10 `template<class T, class Alloc> iterator Oslc_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()`

7.76.4.11 `template<class T, class Alloc> pair_iterator_bool Oslc_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.

7.76.4.12 `template<class T, class Alloc> Oslc_TagTree<T, Alloc>& Oslc_TagTree< T, Alloc >::operator= (const Oslc_TagTree< T, Alloc > & x) [inline]`

Assignment operator

7.76.4.13]

`template<class T, class Alloc> T& Oslc_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().

7.76.4.14 `template<class T, class Alloc> size_type Oslc_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](#)

7.77 Oslc_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

7.77.1 Member Typedef Documentation

7.77.1.1 template<class T, class Alloc> typedef [map_type::const_iterator](#) [Oscl_TagTree](#)< T, Alloc >::const_iterator::mapiter

7.77.1.2 template<class T, class Alloc> typedef const [node_type*](#) [Oscl_TagTree](#)< T, Alloc >::const_iterator::pointer

7.77.1.3 template<class T, class Alloc> typedef const [node_type&](#) [Oscl_TagTree](#)< T, Alloc >::const_iterator::reference

7.77.1.4 template<class T, class Alloc> typedef const_iterator [Oscl_TagTree](#)< T, Alloc >::const_iterator::self

7.77.2 Constructor & Destructor Documentation

7.77.2.1 template<class T, class Alloc> [Oscl_TagTree](#)< T, Alloc >::const_iterator::const_iterator () [inline]

7.77.2.2 template<class T, class Alloc> [Oscl_TagTree](#)< T, Alloc >::const_iterator::const_iterator ([mapiter](#) x) [inline]

7.77.2.3 template<class T, class Alloc> [Oscl_TagTree](#)< T, Alloc >::const_iterator::const_iterator (const const_iterator & it) [inline]

7.77.3 Member Function Documentation

7.77.3.1 template<class T, class Alloc> [reference](#) [Oscl_TagTree](#)< T, Alloc >::const_iterator::operator * () const [inline]

7.77.3.2 template<class T, class Alloc> bool [Oscl_TagTree](#)< T, Alloc >::const_iterator::operator!= (const [self](#) & x) [inline]

7.77.3.3 template<class T, class Alloc> [self](#) [Oscl_TagTree](#)< T, Alloc >::const_iterator::operator++ (int) [inline]

7.77.3.4 template<class T, class Alloc> [self&](#) [Oscl_TagTree](#)< T, Alloc >::const_iterator::operator++ () [inline]

7.77.3.5 template<class T, class Alloc> [self](#) [Oscl_TagTree](#)< T, Alloc >::const_iterator::operator-- (int) [inline]

7.77.3.6 template<class T, class Alloc> [self&](#) [Oscl_TagTree](#)< T, Alloc >::const_iterator::operator-- () [inline]

7.77.3.7 template<class T, class Alloc> [pointer](#) [Oscl_TagTree](#)< T, Alloc >::const_iterator::operator → () const [inline]

7.77.3.8 template<class T, class Alloc> bool [Oscl_TagTree](#)< T, Alloc >::const_iterator::operator== (const [self](#) & x) [inline]

7.77.4 Field Documentation

7.77.4.1 template<class T, class Alloc> [manifer](#) [Oscl_TagTree](#)< T, Alloc >::const_iterator::manif

- [oscl_tagtree.h](#)

7.78 Oslc_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`

7.78.1 Member Typedef Documentation

7.78.1.1 `template<class T, class Alloc> typedef map_type::iterator OscL_TagTree< T, Alloc >::iterator::mapiter`

7.78.1.2 `template<class T, class Alloc> typedef node_type* OscL_TagTree< T, Alloc >::iterator::pointer`

7.78.1.3 `template<class T, class Alloc> typedef node_type& OscL_TagTree< T, Alloc >::iterator::reference`

7.78.1.4 `template<class T, class Alloc> typedef iterator OscL_TagTree< T, Alloc >::iterator::self`

7.78.2 Constructor & Destructor Documentation

7.78.2.1 `template<class T, class Alloc> OscL_TagTree< T, Alloc >::iterator::iterator ()`
`[inline]`

7.78.2.2 `template<class T, class Alloc> OscL_TagTree< T, Alloc >::iterator::iterator (mapiter x)`
`[inline]`

7.78.2.3 `template<class T, class Alloc> OscL_TagTree< T, Alloc >::iterator::iterator (const iterator &it)` `[inline]`

7.78.3 Member Function Documentation

7.78.3.1 `template<class T, class Alloc> reference OscL_TagTree< T, Alloc >::iterator::operator * () const` `[inline]`

7.78.3.2 `template<class T, class Alloc> bool OscL_TagTree< T, Alloc >::iterator::operator != (const self &x)` `[inline]`

7.78.3.3 `template<class T, class Alloc> self OscL_TagTree< T, Alloc >::iterator::operator++ (int)`
`[inline]`

7.78.3.4 `template<class T, class Alloc> self& OscL_TagTree< T, Alloc >::iterator::operator++ ()`
`[inline]`

7.78.3.5 `template<class T, class Alloc> self OscL_TagTree< T, Alloc >::iterator::operator-- (int)`
`[inline]`

7.78.3.6 `template<class T, class Alloc> self& OscL_TagTree< T, Alloc >::iterator::operator-- ()`
`[inline]`

7.78.3.7 `template<class T, class Alloc> pointer OscL_TagTree< T, Alloc >::iterator::operator -> () const` `[inline]`

7.78.3.8 `template<class T, class Alloc> bool OscL_TagTree< T, Alloc >::iterator::operator == (const self &x)` `[inline]`

7.78.4 Field Documentation

7.78.4.1 `template<class T, class Alloc> mapiter OscL_TagTree< T, Alloc >::iterator::mapit`

- [oscl_tagtree.h](#)

7.79 Osci_TagTree< T, Alloc >::Node Struct Reference

```
#include <osci_tagtree.h>
```

Public Types

- typedef [Osci_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 Oslc_TagTree< T, Alloc >::Node

7.79.1 Member Typedef Documentation

7.79.1.1 `template<class T, class Alloc> typedef Oslc_Vector<Node*, Alloc> Oslc_TagTree< T, Alloc >::Node::children_type`

7.79.2 Constructor & Destructor Documentation

7.79.2.1 `template<class T, class Alloc> Oslc_TagTree< T, Alloc >::Node::Node () [inline]`

7.79.3 Member Function Documentation

7.79.3.1 `template<class T, class Alloc> tag_type::size_type Oslc_TagTree< T, Alloc >::Node::depth () [inline]`

7.79.3.2 `template<class T, class Alloc> void Oslc_TagTree< T, Alloc >::Node::sort_children () [inline]`

7.79.4 Field Documentation

7.79.4.1 `template<class T, class Alloc> children_type Oslc_TagTree< T, Alloc >::Node::children`

7.79.4.2 `template<class T, class Alloc> Node* Oslc_TagTree< T, Alloc >::Node::parent`

7.79.4.3 `template<class T, class Alloc> tag_type Oslc_TagTree< T, Alloc >::Node::tag`

7.79.4.4 `template<class T, class Alloc> T Oslc_TagTree< T, Alloc >::Node::value`

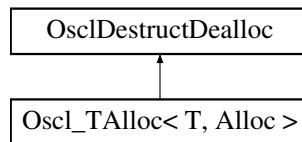
The documentation for this struct was generated from the following file:

- [oslc_tagtree.h](#)

7.80 Oslc_TAlloc< T, Alloc > Class Template Reference

```
#include <oscl_defalloc.h>
```

Inheritance diagram for Oslc_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 [~Oslc_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](#) ([OslcAny](#) *p)
- void [deallocate](#) ([OslcAny](#) *p, [size_type](#) n)
- void [destruct_and_dealloc](#) ([OslcAny](#) *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 Osci_TAlloc< T, Alloc >
```

7.80.1 Member Typedef Documentation

7.80.1.1 `template<class T, class Alloc> typedef const T* Osci_TAlloc< T, Alloc >::const_pointer`

7.80.1.2 `template<class T, class Alloc> typedef const T& Osci_TAlloc< T, Alloc >::const_reference`

7.80.1.3 `template<class T, class Alloc> typedef T* Osci_TAlloc< T, Alloc >::pointer`

7.80.1.4 `template<class T, class Alloc> typedef T& Osci_TAlloc< T, Alloc >::reference`

7.80.1.5 `template<class T, class Alloc> typedef uint32 Osci_TAlloc< T, Alloc >::size_type`

7.80.1.6 `template<class T, class Alloc> typedef T Osci_TAlloc< T, Alloc >::value_type`

7.80.2 Constructor & Destructor Documentation

7.80.2.1 `template<class T, class Alloc> virtual Osci_TAlloc< T, Alloc >::~Osci_TAlloc ()`
 [inline, virtual]

7.80.3 Member Function Documentation

7.80.3.1 `template<class T, class Alloc> const_pointer Osci_TAlloc< T, Alloc >::address`
 (`const_reference r`) const [inline]

7.80.3.2 `template<class T, class Alloc> pointer Osci_TAlloc< T, Alloc >::address` (`reference r`)
 [inline]

7.80.3.3 `template<class T, class Alloc> pointer Osci_TAlloc< T, Alloc >::alloc_and_construct`
 (`const_reference val`) [inline]

7.80.3.4 `template<class T, class Alloc> pointer Osci_TAlloc< T, Alloc >::alloc_and_construct_fl`
 (`const_reference val`, `const char *file_name`, `const int line_num`) [inline]

7.80.3.5 `template<class T, class Alloc> pointer Osci_TAlloc< T, Alloc >::allocate` (`uint32 size`)
 [inline]

7.80.3.6 `template<class T, class Alloc> pointer Osci_TAlloc< T, Alloc >::allocate_fl` (`uint32 size`,
`const char *file_name`, `const int line_num`) [inline]

7.80.3.7 `template<class T, class Alloc> void Osci_TAlloc< T, Alloc >::construct` (`pointer p`,
`const_reference val`) [inline]

7.80.3.8 `template<class T, class Alloc> void Osci_TAlloc< T, Alloc >::deallocate` (`OsciAny * p`,
`size_type n`) [inline]

7.80.3.9 `template<class T, class Alloc> void Osci_TAlloc< T, Alloc >::deallocate` (`OsciAny * p`)
 [inline]

7.80.3.10 `template<class T, class Alloc> void Osci_TAlloc< T, Alloc >::destroy` (`pointer p`)
 [inline]

7.80.3.11 `template<class T, class Alloc> void Osci_TAlloc< T, Alloc >::destruct_and_dealloc`
 (`OsciAny * p`) [inline, virtual]

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

- [oscl_defalloc.h](#)

7.81 `OscI_TAlloc< T, Alloc >::rebind< U, V >` Struct Template Reference

```
#include <oscl_defalloc.h>
```

Public Types

- typedef `OscI_TAlloc< U, V >` `other`

```
template<class T, class Alloc>template<class U, class V> struct OscI_TAlloc< T, Alloc >::rebind< U, V >
```

7.81.1 Member Typedef Documentation

7.81.1.1 `template<class T, class Alloc> template<class U, class V> typedef OscI_TAlloc<U, V> OscI_TAlloc< T, Alloc >::rebind< U, V >::other`

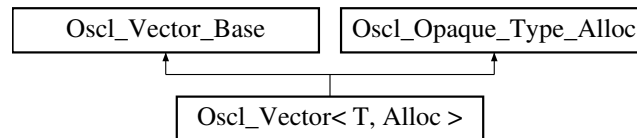
The documentation for this struct was generated from the following file:

- `oscl_defalloc.h`

7.82 Osl_Vector< T, Alloc > Class Template Reference

```
#include <osl_vector.h>
```

Inheritance diagram for Osl_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

- [Osl_Vector](#) ()
- [Osl_Vector](#) (uint32 n)
- [Osl_Vector](#) (const Osl_Vector< T, Alloc > &x)
- virtual [~Osl_Vector](#) ()
- Osl_Vector< T, Alloc > & [operator=](#) (const Osl_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)

7.82.1 Detailed Description

template<class T, class Alloc> class Osl_Vector< T, Alloc >

Osl_Vector Class. A subset of STL::Vector methods. Osl_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.

7.82.2 Member Typedef Documentation

7.82.2.1 template<class T, class Alloc> typedef const T* Osl_Vector< T, Alloc >::const_iterator

7.82.2.2 template<class T, class Alloc> typedef const T& Osl_Vector< T, Alloc >::const_reference

7.82.2.3 template<class T, class Alloc> typedef T* Osl_Vector< T, Alloc >::iterator

7.82.2.4 template<class T, class Alloc> typedef T* Osl_Vector< T, Alloc >::pointer

7.82.2.5 template<class T, class Alloc> typedef T& Osl_Vector< T, Alloc >::reference

7.82.2.6 template<class T, class Alloc> typedef T Osl_Vector< T, Alloc >::value_type

7.82.3 Constructor & Destructor Documentation

7.82.3.1 template<class T, class Alloc> Osl_Vector< T, Alloc >::Osl_Vector () [inline]

Creates an empty vector.

7.82.3.2 template<class T, class Alloc> Osl_Vector< T, Alloc >::Osl_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.

7.82.3.3 template<class T, class Alloc> Osl_Vector< T, Alloc >::Osl_Vector (const Osl_Vector< T, Alloc > &x) [inline]

Copy Constructor.

Parameters:

x vector class to copy.

7.82.3.4 `template<class T, class Alloc> virtual Osl_Vector< T, Alloc >::~~Osl_Vector ()`
`[inline, virtual]`

The destructor.

7.82.4 Member Function Documentation

7.82.4.1 `template<class T, class Alloc> const T& Osl_Vector< T, Alloc >::back () const`
`[inline]`

Returns the last element.

7.82.4.2 `template<class T, class Alloc> T& Osl_Vector< T, Alloc >::back ()` `[inline]`

Returns the last element.

7.82.4.3 `template<class T, class Alloc> iterator Osl_Vector< T, Alloc >::begin () const`
`[inline]`

Returns an iterator pointing to the beginning of the vector.

Reimplemented from [Osl_Vector_Base](#).

7.82.4.4 `template<class T, class Alloc> void Osl_Vector< T, Alloc >::clear ()` `[inline]`

Removes all elements.

7.82.4.5 `template<class T, class Alloc> void Osl_Vector< T, Alloc >::destroy ()` `[inline]`

Destroy – this is like an explicit destructor call.

Reimplemented from [Osl_Vector_Base](#).

7.82.4.6 `template<class T, class Alloc> iterator Osl_Vector< T, Alloc >::end () const`
`[inline]`

Returns an iterator pointing to the end of the vector..

Reimplemented from [Osl_Vector_Base](#).

7.82.4.7 `template<class T, class Alloc> iterator Osl_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

7.82.4.8 `template<class T, class Alloc> iterator Osl_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

7.82.4.9 `template<class T, class Alloc> const T& Osl_Vector< T, Alloc >::front () const`
`[inline]`

Returns the first element.

7.82.4.10 `template<class T, class Alloc> T& Osl_Vector< T, Alloc >::front ()` `[inline]`

Returns the first element.

7.82.4.11 `template<class T, class Alloc> iterator Osl_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

7.82.4.12 `template<class T, class Alloc> Osl_Vector<T, Alloc>& Osl_Vector< T, Alloc >::operator= (const Osl_Vector< T, Alloc > & x)` `[inline]`

The assignment operator

7.82.4.13]

`template<class T, class Alloc> const T& Osl_Vector< T, Alloc >::operator[] (uint32 n) const`
`[inline]`

Returns the n'th element.

Parameters:

n element position to return

7.82.4.14]

`template<class T, class Alloc> T& Osl_Vector< T, Alloc >::operator[] (uint32 n)` `[inline]`

Returns the n'th element.

Parameters:

n element position to return

7.82.4.15 `template<class T, class Alloc> void OscVector< T, Alloc >::pop_back ()` [inline]

Removes the last element.

Reimplemented from [OscVectorBase](#).

7.82.4.16 `template<class T, class Alloc> void OscVector< 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

7.82.4.17 `template<class T, class Alloc> void OscVector< 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

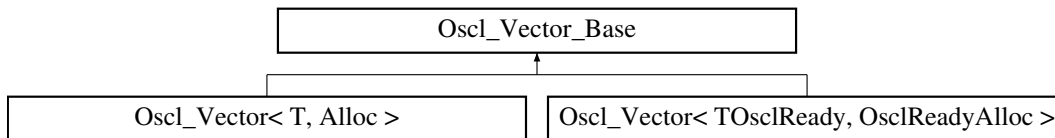
The documentation for this class was generated from the following file:

- [oscl_vector.h](#)

7.83 Osl_Vector_Base Class Reference

```
#include <osl_vector.h>
```

Inheritance diagram for Osl_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](#) (Osl_Opaque_Type_Alloc *aType)
- OSCL_IMPORT_REF void [construct](#) (Osl_Opaque_Type_Alloc *aType, uint32 n)
- OSCL_IMPORT_REF void [construct](#) (Osl_Opaque_Type_Alloc *aType, const Osl_Vector_Base &x)
- virtual [~Osl_Vector_Base](#) ()
- OSCL_IMPORT_REF void [push_back](#) (const OslAny *x)
- OSCL_IMPORT_REF void [pop_back](#) ()
- OSCL_IMPORT_REF void [push_front](#) (const OslAny *x)
- OSCL_IMPORT_REF OslAny * [insert](#) (OslAny *pos, const OslAny *x)
- OSCL_IMPORT_REF OslAny * [erase](#) (OslAny *pos)
- OSCL_IMPORT_REF OslAny * [erase](#) (OslAny *first, OslAny *last)
- OSCL_IMPORT_REF void [assign_vector](#) (const Osl_Vector_Base &x)
- OSCL_IMPORT_REF void [destroy](#) ()

Protected Attributes

- uint32 [numelems](#)
- uint32 [bufsize](#)
- OslAny * [elems](#)
- uint32 [sizeof_T](#)

Friends

- class [OslPriorityQueueBase](#)

7.83.1 Detailed Description

Osl_Vector_Base is a non-templated base class for [Osl_Vector](#). The purpose of this base class is to avoid large inline routines in the [Osl_Vector](#) implementation. This class is not intended for direct instantiation except by [Osl_Vector](#).

7.83.2 Constructor & Destructor Documentation

7.83.2.1 `virtual Osl_Vector_Base::~~Osl_Vector_Base () [inline, protected, virtual]`

The destructor.

7.83.3 Member Function Documentation

7.83.3.1 `OSCL_IMPORT_REF void Osl_Vector_Base::assign_vector (const Osl_Vector_Base & x) [protected]`

7.83.3.2 `uint32 Osl_Vector_Base::capacity () const [inline]`

Returns the allocated memory of the vector in units of number of elements.

7.83.3.3 `OSCL_IMPORT_REF void Osl_Vector_Base::construct (Osl_Opaque_Type_Alloc * aType, const Osl_Vector_Base & x) [protected]`

7.83.3.4 `OSCL_IMPORT_REF void Osl_Vector_Base::construct (Osl_Opaque_Type_Alloc * aType, uint32 n) [protected]`

7.83.3.5 `OSCL_IMPORT_REF void Osl_Vector_Base::construct (Osl_Opaque_Type_Alloc * aType) [protected]`

7.83.3.6 `OSCL_IMPORT_REF void Osl_Vector_Base::destroy () [protected]`

Reimplemented in [Osl_Vector< T, Alloc >](#), [Osl_Vector< OslComponentRegistryElement, OslMemAllocator >](#), [Osl_Vector< uint32, OslMemAllocator >](#), [Osl_Vector< OslSocketServRequestQElem, OslMemAllocator >](#), [Osl_Vector< Node *, Alloc >](#), [Osl_Vector< OslFixedCacheParam, OslMemAllocator >](#), [Osl_Vector< OslSocketRequest *, OslMemAllocator >](#), [Osl_Vector< entry_type *, Alloc >](#), [Osl_Vector< OSCL_HeapString< OslMemAllocator >, OslMemAllocator >](#), [Osl_Vector< OslAsyncFileBuffer *, OslMemAllocator >](#), [Osl_Vector< TOslFileOffset, OslMemAllocator >](#), [Osl_Vector< MemPoolBufferInfo *, OslMemAllocator >](#), [Osl_Vector< OslSharedPtr< PVLoggerFilter >, alloc_type >](#), [Osl_Vector< TOslReady, OslReadyAlloc >](#), [Osl_Vector< OslFileCacheBuffer, OslMemAllocator >](#), [Osl_Vector< OslSharedPtr< PVLoggerAppender >, alloc_type >](#), [Osl_Vector< OslAny *, OslMemAllocator >](#), and [Osl_Vector< OslNetworkAddress, OslMemAllocator >](#).

7.83.3.7 `bool Osl_Vector_Base::empty () const [inline]`

True if the vector's size is 0.

7.83.3.8 OSCL_IMPORT_REF OslAny* Osl_Vector_Base::erase (OslAny * *first*, OslAny * *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

7.83.3.9 OSCL_IMPORT_REF OslAny* Osl_Vector_Base::erase (OslAny * *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

7.83.3.10 OSCL_IMPORT_REF OslAny* Osl_Vector_Base::insert (OslAny * *pos*, const OslAny * *x*) [protected]

Inserts a new element at a specific position.

Parameters:

pos iterator at insert position.

x pointer to new element

7.83.3.11 OSCL_IMPORT_REF void Osl_Vector_Base::pop_back () [protected]

Removes the last element.

Reimplemented in [Osl_Vector< T, Alloc >](#), [Osl_Vector< OslComponentRegistryElement, OslMemAllocator >](#), [Osl_Vector< uint32, OslMemAllocator >](#), [Osl_Vector< OslSocketServRequestQElem, OslMemAllocator >](#), [Osl_Vector< Node *, Alloc >](#), [Osl_Vector< OslFixedCacheParam, OslMemAllocator >](#), [Osl_Vector< OslSocketRequest *, OslMemAllocator >](#), [Osl_Vector< entry_type *, Alloc >](#), [Osl_Vector< OSCL_HeapString< OslMemAllocator >, OslMemAllocator >](#), [Osl_Vector< OslAsyncFileBuffer *, OslMemAllocator >](#), [Osl_Vector< TOslFileOffset, OslMemAllocator >](#), [Osl_Vector< MemPoolBufferInfo *, OslMemAllocator >](#), [Osl_Vector< OslSharedPtr< PVLoggerFilter >, alloc_type >](#), [Osl_Vector< TOslReady, OslReadyAlloc >](#), [Osl_Vector< OslFileCacheBuffer, OslMemAllocator >](#), [Osl_Vector< OslSharedPtr< PVLoggerAppender >, alloc_type >](#), [Osl_Vector< OslAny *, OslMemAllocator >](#), and [Osl_Vector< OslNetworkAddress, OslMemAllocator >](#).

7.83.3.12 OSCL_IMPORT_REF void Osl_Vector_Base::push_back (const OslAny * *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

7.83.3.13 OSL_IMPORT_REF void Osl_Vector_Base::push_front (const OslAny * *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

7.83.3.14 OSL_IMPORT_REF void Osl_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

7.83.3.15 uint32 Osl_Vector_Base::size () const [inline]

Returns the size of the vector in units of number of elements.

7.83.4 Friends And Related Function Documentation**7.83.4.1 friend class OslPriorityQueueBase** [friend]**7.83.5 Field Documentation****7.83.5.1 uint32 Osl_Vector_Base::bufsize** [protected]**7.83.5.2 OslAny* Osl_Vector_Base::elems** [protected]**7.83.5.3 uint32 Osl_Vector_Base::numelems** [protected]**7.83.5.4 uint32 Osl_Vector_Base::sizeof_T** [protected]

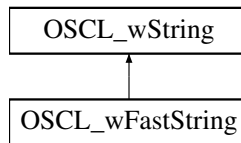
The documentation for this class was generated from the following file:

- [oscl_vector.h](#)

7.84 OSCL_wFastString Class Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL_wFastString::



Public Types

- typedef OSCL_wString::chartype [chartype](#)
- typedef [TOSCL_wStringOp](#) optype
- typedef [OSCL_String::chartype](#) other_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](#) (const [other_chartype](#) *buf, uint32 numofbyte, [optype](#) op)
- OSCL_IMPORT_REF void [set_length](#) ()

Friends

- class [OSCL_wString](#)

7.84.1 Detailed Description

[OSCL_wFastString](#) is identical to [OSCL_FastString](#) except that it uses wide-character format. For descriptions, see [OSCL_FastString](#).

7.84.2 Member Typedef Documentation

7.84.2.1 typedef OSCL_wString::chartype OSCL_wFastString::chartype

Reimplemented from [OSCL_wString](#).

7.84.2.2 typedef [TOSCL_wStringOp](#) OSCL_wFastString::optype

7.84.2.3 typedef [OSCL_String::chartype](#) OSCL_wFastString::other_chartype

7.84.3 Constructor & Destructor Documentation

7.84.3.1 OSCL_IMPORT_REF OSCL_wFastString::OSCL_wFastString ()

7.84.3.2 OSCL_IMPORT_REF OSCL_wFastString::OSCL_wFastString (const OSCL_wFastString & *src*)

7.84.3.3 OSCL_IMPORT_REF OSCL_wFastString::OSCL_wFastString (const [chartype](#) * *cstring*)

7.84.3.4 OSCL_IMPORT_REF OSCL_wFastString::OSCL_wFastString ([chartype](#) * *buf*, uint32 *maxlen*)

7.84.3.5 OSCL_IMPORT_REF OSCL_wFastString::~~OSCL_wFastString ()

7.84.4 Member Function Documentation

7.84.4.1 OSCL_IMPORT_REF const [chartype](#)* OSCL_wFastString::get_cstr () [virtual]

Implements [OSCL_wString](#).

7.84.4.2 OSCL_IMPORT_REF uint32 OSCL_wFastString::get_maxsize () [virtual]

Implements [OSCL_wString](#).

7.84.4.3 OSCL_IMPORT_REF uint32 OSCL_wFastString::get_size () [virtual]

Implements [OSCL_wString](#).

7.84.4.4 OSCL_IMPORT_REF [chartype](#)* OSCL_wFastString::get_str () [virtual]

Implements [OSCL_wString](#).

7.84.4.5 OSCL_IMPORT_REF OSCL_wFastString& OSCL_wFastString::operator= (const [chartype](#) * *cstring*)

Reimplemented from [OSCL_wString](#).

- 7.84.4.6 OSCL_IMPORT_REF OSCL_wFastString& OSCL_wFastString::operator= (const OSCL_wFastString & *src*)
- 7.84.4.7 OSCL_IMPORT_REF void OSCL_wFastString::set (const **other_chartype** * *buf*, uint32 *numofbyte*, **optype** *op*)
- 7.84.4.8 OSCL_IMPORT_REF void OSCL_wFastString::set (**chartype** * *cstring*, uint32 *maxlen*)
- 7.84.4.9 OSCL_IMPORT_REF void OSCL_wFastString::set_length ()

7.84.5 Friends And Related Function Documentation

- 7.84.5.1 friend class OSCL_wString [friend]

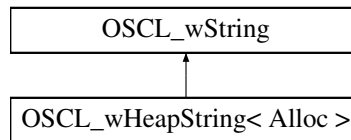
The documentation for this class was generated from the following file:

- [oscl_string_containers.h](#)

7.85 OSCL_wHeapString< Alloc > Class Template Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL_wHeapString< Alloc >::



Public Types

- typedef OSCL_wString::chartype [chartype](#)
- typedef TOSCL_wStringOp [optype](#)
- typedef OSCL_String::chartype [other_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)
- void [set](#) (const other_chartype *buf, optype op)
- void [set](#) (const other_chartype *buf, uint32 length, optype op)

Friends

- class [OSCL_wString](#)

7.85.1 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](#).

7.85.2 Member Typedef Documentation

7.85.2.1 `template<class Alloc> typedef OSCL_wString::chartype OSCL_wHeapString< Alloc >::chartype`

Reimplemented from [OSCL_wString](#).

7.85.2.2 `template<class Alloc> typedef TOSCL_wStringOp OSCL_wHeapString< Alloc >::optype`

7.85.2.3 `template<class Alloc> typedef OSCL_String::chartype OSCL_wHeapString< Alloc >::other_chartype`

7.85.3 Friends And Related Function Documentation

7.85.3.1 `template<class Alloc> friend class OSCL_wString [friend]`

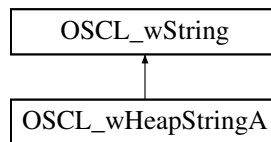
The documentation for this class was generated from the following file:

- [oscl_string_containers.h](#)

7.86 OSCL_wHeapStringA Class Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL_wHeapStringA::



Public Types

- typedef OSCL_wString::chartype [chartype](#)
- typedef [TOSCL_wStringOp](#) optype
- typedef [OSCL_String::chartype](#) other_chartype

Public Methods

- OSCL_IMPORT_REF [OSCL_wHeapStringA](#) ()
- OSCL_IMPORT_REF [OSCL_wHeapStringA](#) ([OscDefAlloc](#) *alloc, [OscRefCount](#) *ref=NULL)
- OSCL_IMPORT_REF [OSCL_wHeapStringA](#) (const [OSCL_wHeapStringA](#) &src)
- OSCL_IMPORT_REF [OSCL_wHeapStringA](#) (const [OSCL_wHeapStringA](#) &src, [OscDefAlloc](#) *alloc, [OscRefCount](#) *ref=NULL)
- OSCL_IMPORT_REF [OSCL_wHeapStringA](#) (const [OSCL_wString](#) &src, [OscDefAlloc](#) *alloc=NULL, [OscRefCount](#) *ref=NULL)
- OSCL_IMPORT_REF [OSCL_wHeapStringA](#) (const [chartype](#) *cstr, [OscDefAlloc](#) *alloc=NULL, [OscRefCount](#) *ref=NULL)
- OSCL_IMPORT_REF [OSCL_wHeapStringA](#) (const [chartype](#) *buf, uint32 length, [OscDefAlloc](#) *alloc=NULL, [OscRefCount](#) *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)
- OSCL_IMPORT_REF void [set](#) (const [other_chartype](#) *buf, [optype](#) op)
- OSCL_IMPORT_REF void [set](#) (const [other_chartype](#) *buf, uint32 length, [optype](#) op)

Friends

- class [OSCL_wString](#)

7.86.1 Detailed Description

OSCL_wHeapStringA is identical to [OSCL_HeapStringA](#) except that it uses wide-character format. For descriptions, see [OSCL_HeapStringA](#).

7.86.2 Member Typedef Documentation

7.86.2.1 typedef OSCL_wString::chartype OSCL_wHeapStringA::chartype

Reimplemented from [OSCL_wString](#).

7.86.2.2 typedef **TOSCL_wStringOp** OSCL_wHeapStringA::optype

7.86.2.3 typedef **OSCL_String::chartype** OSCL_wHeapStringA::other_chartype

7.86.3 Constructor & Destructor Documentation

7.86.3.1 OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA ()

7.86.3.2 OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA (**Oscl_DefAlloc** * *alloc*, **OsclRefCount** * *ref* = NULL)

7.86.3.3 OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA (const OSCL_wHeapStringA & *src*)

7.86.3.4 OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA (const OSCL_wHeapStringA & *src*, **Oscl_DefAlloc** * *alloc*, **OsclRefCount** * *ref* = NULL)

7.86.3.5 OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA (const **OSCL_wString** & *src*, **Oscl_DefAlloc** * *alloc* = NULL, **OsclRefCount** * *ref* = NULL)

7.86.3.6 OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA (const **chartype** * *cstr*, **Oscl_DefAlloc** * *alloc* = NULL, **OsclRefCount** * *ref* = NULL)

7.86.3.7 OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA (const **chartype** * *buf*, uint32 *length*, **Oscl_DefAlloc** * *alloc* = NULL, **OsclRefCount** * *ref* = NULL)

7.86.3.8 OSCL_IMPORT_REF OSCL_wHeapStringA::~~OSCL_wHeapStringA ()

7.86.4 Member Function Documentation

7.86.4.1 OSCL_IMPORT_REF const **chartype*** OSCL_wHeapStringA::get_cstr () [virtual]

Implements [OSCL_wString](#).

7.86.4.2 OSCL_IMPORT_REF uint32 OSCL_wHeapStringA::get_maxsize () [virtual]

Implements [OSCL_wString](#).

7.86.4.3 OSCL_IMPORT_REF uint32 OSCL_wHeapStringA::get_size () [virtual]

Implements [OSCL_wString](#).

7.86.4.4 OSCL_IMPORT_REF chartype* OSCL_wHeapStringA::get_str () [virtual]

Implements [OSCL_wString](#).

7.86.4.5 OSCL_IMPORT_REF OSCL_wHeapStringA& OSCL_wHeapStringA::operator= (const chartype * *cstr*)

Reimplemented from [OSCL_wString](#).

7.86.4.6 OSCL_IMPORT_REF OSCL_wHeapStringA& OSCL_wHeapStringA::operator= (const OSCL_wString & *src*)

Reimplemented from [OSCL_wString](#).

7.86.4.7 OSCL_IMPORT_REF OSCL_wHeapStringA& OSCL_wHeapStringA::operator= (const OSCL_wHeapStringA & *src*)**7.86.4.8 OSCL_IMPORT_REF void OSCL_wHeapStringA::set (const other_chartype * *buf*, uint32 *length*, optype *op*)****7.86.4.9 OSCL_IMPORT_REF void OSCL_wHeapStringA::set (const other_chartype * *buf*, optype *op*)****7.86.4.10 OSCL_IMPORT_REF void OSCL_wHeapStringA::set (const chartype * *buf*, uint32 *length*)****7.86.5 Friends And Related Function Documentation****7.86.5.1 friend class OSCL_wString [friend]**

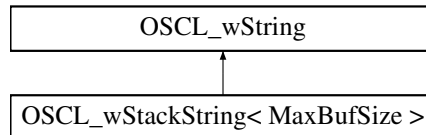
The documentation for this class was generated from the following file:

- [oscl_string_containers.h](#)

7.87 OSCL_wStackString< MaxBufSize > Class Template Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL_wStackString< MaxBufSize >::



Public Types

- typedef OSCL_wString::chartype [chartype](#)
- typedef TOSCL_wStringOp [optype](#)
- typedef OSCL_String::chartype [other_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)
- void set (const other_chartype *buf, optype op)
- void set (const other_chartype *buf, uint32 length, optype op)

Friends

- class OSCL_wString

7.87.1 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](#).

7.87.2 Member Typedef Documentation

7.87.2.1 `template<uint32 MaxBufSize> typedef OSCL_wString::chartype OSCL_wStackString< MaxBufSize >::chartype`

Reimplemented from [OSCL_wString](#).

7.87.2.2 `template<uint32 MaxBufSize> typedef TOSCL_wStringOp OSCL_wStackString< MaxBufSize >::optype`

7.87.2.3 `template<uint32 MaxBufSize> typedef OSCL_String::chartype OSCL_wStackString< MaxBufSize >::other_chartype`

7.87.3 Friends And Related Function Documentation

7.87.3.1 `template<uint32 MaxBufSize> friend class OSCL_wString [friend]`

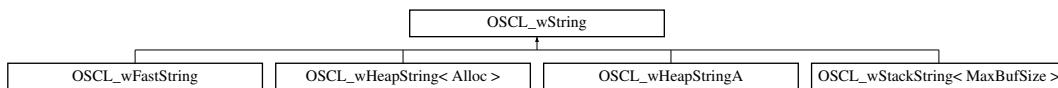
The documentation for this class was generated from the following file:

- [oscl_string_containers.h](#)

7.88 OSCL_wString Class Reference

```
#include <oscl_string.h>
```

Inheritance diagram for OSCL_wString::



Public Types

- typedef [oscl_wchar](#) [chartye](#)

Public Methods

- virtual uint32 [get_size](#) () const=0
- virtual uint32 [get_maxsize](#) () const=0
- virtual const [chartye](#) * [get_cstr](#) () const=0
- virtual bool [is_writable](#) () const
- virtual [chartye](#) * [get_str](#) () const=0
- OSCL_wString & [operator=](#) (const OSCL_wString &src)
- OSCL_wString & [operator=](#) (const [chartye](#) *cstr)
- OSCL_wString & [operator+=](#) (const OSCL_wString &src)
- OSCL_wString & [operator+=](#) (const [chartye](#) *cstr)
- OSCL_wString & [operator+=](#) (const [chartye](#) c)
- bool [operator==](#) (const OSCL_wString &src) const
- bool [operator!=](#) (const OSCL_wString &src) const
- bool [operator<](#) (const OSCL_wString &src) const
- bool [operator<=](#) (const OSCL_wString &src) const
- bool [operator>](#) (const OSCL_wString &src) const
- bool [operator>=](#) (const OSCL_wString &src) const
- bool [operator==](#) (const [chartye](#) *cstr) const
- [chartye](#) [operator\[\]](#) (uint32 index) const
- virtual [chartye](#) [read](#) (uint32 index) const
- virtual uint32 [setrep_to_wide_char](#) (const char *src, uint32 len, [TOSCL_wStringOp](#) op, [Oscl_DefAlloc](#) *aAlloc)
- virtual int8 [hash](#) () const
- virtual void [write](#) (uint32 index, [chartye](#) c)
- virtual void [write](#) (uint32 offset, uint32 length, const [chartye](#) *buf)

Protected Methods

- [OSCL_wString](#) ()
- virtual [~OSCL_wString](#) ()
- virtual void [set_rep](#) (const [chartye](#) *cstr)=0
- virtual void [append_rep](#) (const [chartye](#) *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

7.88.1 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](#).

7.88.2 Member Typedef Documentation

7.88.2.1 typedef `oscl_wchar` `OSCL_wString::chartype`

Reimplemented in [OSCL_wHeapString< Alloc >](#), [OSCL_wHeapStringA](#), [OSCL_wStackString< MaxBufSize >](#), and [OSCL_wFastString](#).

7.88.3 Constructor & Destructor Documentation

7.88.3.1 `OSCL_wString::OSCL_wString ()` [protected]

7.88.3.2 `virtual OSCL_wString::~~OSCL_wString ()` [protected, virtual]

7.88.4 Member Function Documentation

7.88.4.1 `virtual void OSCL_wString::append_rep (const OSCL_wString & src)` [protected, pure virtual]

7.88.4.2 `virtual void OSCL_wString::append_rep (const chartype * cstr)` [protected, pure virtual]

7.88.4.3 `virtual const chartype* OSCL_wString::get_cstr ()` [pure virtual]

Implemented in [OSCL_wHeapString< Alloc >](#), [OSCL_wHeapStringA](#), [OSCL_wStackString< MaxBufSize >](#), and [OSCL_wFastString](#).

7.88.4.4 `virtual uint32 OSCL_wString::get_maxsize ()` [pure virtual]

Implemented in [OSCL_wHeapString< Alloc >](#), [OSCL_wHeapStringA](#), [OSCL_wStackString< MaxBufSize >](#), and [OSCL_wFastString](#).

7.88.4.5 `virtual uint32 OSCL_wString::get_size ()` [pure virtual]

Implemented in [OSCL_wHeapString< Alloc >](#), [OSCL_wHeapStringA](#), [OSCL_wStackString< MaxBufSize >](#), and [OSCL_wFastString](#).

7.88.4.6 `virtual chartype* OSCL_wString::get_str ()` [pure virtual]

Implemented in [OSCL_wHeapString< Alloc >](#), [OSCL_wHeapStringA](#), [OSCL_wStackString< MaxBufSize >](#), and [OSCL_wFastString](#).

7.88.4.7 `virtual int8 OSCL_wString::hash () [virtual]`

7.88.4.8 `virtual bool OSCL_wString::is_writable () [virtual]`

7.88.4.9 `bool OSCL_wString::operator!= (const OSCL_wString & src) const`

7.88.4.10 `OSCL_wString& OSCL_wString::operator+= (const chartype c)`

7.88.4.11 `OSCL_wString& OSCL_wString::operator+= (const chartype * cstr)`

7.88.4.12 `OSCL_wString& OSCL_wString::operator+= (const OSCL_wString & src)`

7.88.4.13 `bool OSCL_wString::operator< (const OSCL_wString & src) const`

7.88.4.14 `bool OSCL_wString::operator<= (const OSCL_wString & src) const`

7.88.4.15 `OSCL_wString& OSCL_wString::operator= (const chartype * cstr)`

Reimplemented in [OSCL_wHeapString< Alloc >](#), [OSCL_wHeapStringA](#), [OSCL_wStackString< MaxBufSize >](#), and [OSCL_wFastString](#).

7.88.4.16 `OSCL_wString& OSCL_wString::operator= (const OSCL_wString & src)`

Reimplemented in [OSCL_wHeapString< Alloc >](#), [OSCL_wHeapStringA](#), and [OSCL_wStackString< MaxBufSize >](#).

7.88.4.17 `bool OSCL_wString::operator== (const chartype * cstr) const`

7.88.4.18 `bool OSCL_wString::operator== (const OSCL_wString & src) const`

7.88.4.19 `bool OSCL_wString::operator> (const OSCL_wString & src) const`

7.88.4.20 `bool OSCL_wString::operator>= (const OSCL_wString & src) const`

7.88.4.21 `]`

`chartype OSCL_wString::operator[] (uint32 index) const`

- 7.88.4.22 **virtual `char`** OSCL_wString::read (uint32 *index*) const [virtual]
- 7.88.4.23 **virtual void** OSCL_wString::set_len (uint32 *len*) [protected, pure virtual]
- 7.88.4.24 **virtual void** OSCL_wString::set_rep (const OSCL_wString & *src*) [protected, pure virtual]
- 7.88.4.25 **virtual void** OSCL_wString::set_rep (const **`char`** * *ctr*) [protected, pure virtual]
- 7.88.4.26 **virtual uint32** OSCL_wString::setrep_to_wide_char (const char * *src*, uint32 *len*, **`TO`**OSCL_wStringOp *op*, **`Oscl`**DefAlloc * *aAlloc*) [virtual]
- 7.88.4.27 **virtual void** OSCL_wString::write (uint32 *offset*, uint32 *length*, const **`char`** * *buf*) [virtual]
- 7.88.4.28 **virtual void** OSCL_wString::write (uint32 *index*, **`char`** *c*) [virtual]

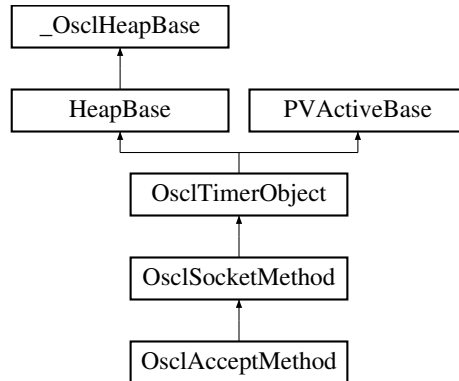
The documentation for this class was generated from the following file:

- [oscl_string.h](#)

7.89 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 \(OsclIPSocketI &c\)](#)

7.89.1 Constructor & Destructor Documentation

7.89.1.1 [OsclAcceptMethod::~OsclAcceptMethod \(\)](#)

7.89.2 Member Function Documentation

7.89.2.1 [TPVSocketEvent OsclAcceptMethod::Accept \(int32 aTimeout\)](#)

7.89.2.2 [OsclAcceptRequest* OsclAcceptMethod::AcceptRequest \(\) \[inline\]](#)

7.89.2.3 [void OsclAcceptMethod::DiscardAcceptedSocket \(\)](#)

7.89.2.4 [OsclSocketI* OsclAcceptMethod::GetAcceptedSocket \(\)](#)

7.89.2.5 [OsclAcceptMethod* OsclAcceptMethod::NewL \(OsclIPSocketI &c\) \[static\]](#)

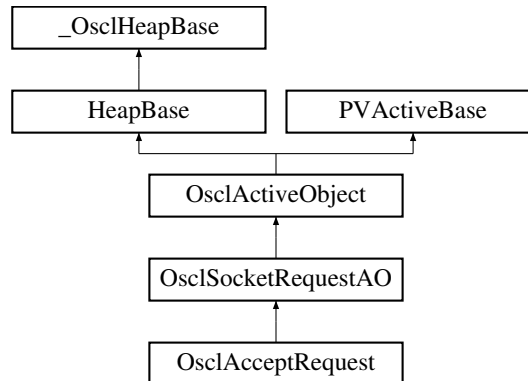
The documentation for this class was generated from the following file:

- [oscl_socket_accept.h](#)

7.90 OsclAcceptRequest Class Reference

```
#include <oscl_socket_accept.h>
```

Inheritance diagram for OsclAcceptRequest::



Public Methods

- [OsclAcceptRequest \(OsclSocketMethod &c\)](#)
- void [Accept \(OsclSocketI &aSocket\)](#)

7.90.1 Constructor & Destructor Documentation

7.90.1.1 [OsclAcceptRequest::OsclAcceptRequest \(OsclSocketMethod &c\)](#) [inline]

7.90.2 Member Function Documentation

7.90.2.1 void [OsclAcceptRequest::Accept \(OsclSocketI &aSocket\)](#)

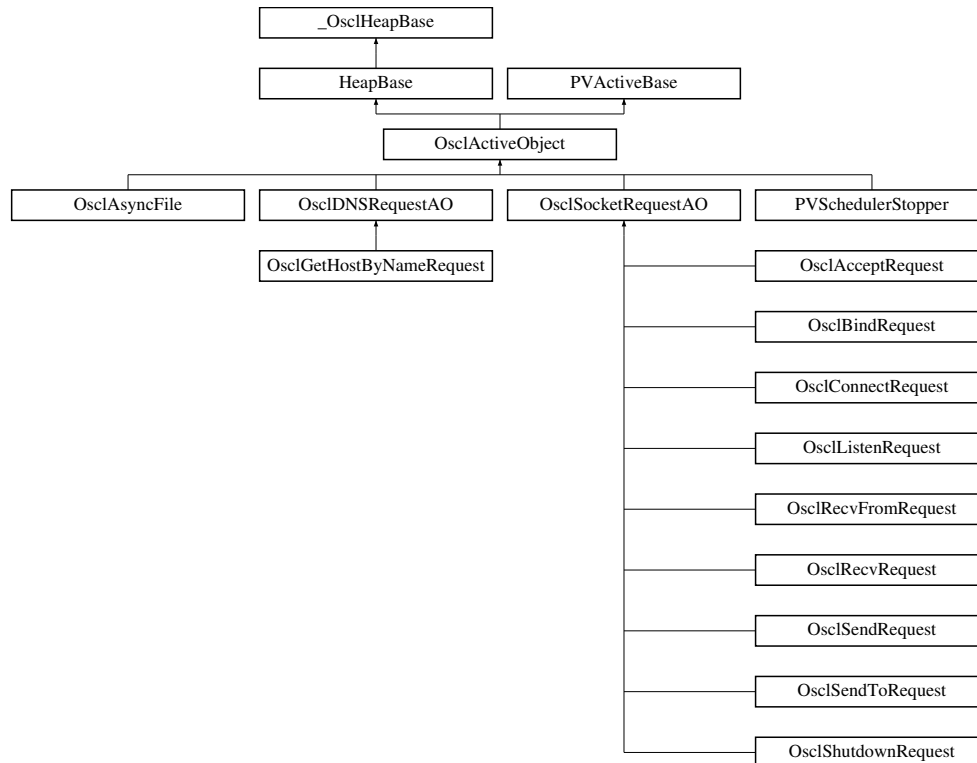
The documentation for this class was generated from the following file:

- [oscl_socket_accept.h](#)

7.91 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)

7.91.1 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.

7.91.2 Member Enumeration Documentation

7.91.2.1 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.

7.91.3 Constructor & Destructor Documentation

7.91.3.1 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.

7.91.3.2 virtual OSCL_IMPORT_REF OsclActiveObject::~OsclActiveObject () [virtual]

Destructor.

7.91.4 Member Function Documentation

7.91.4.1 OSCL_IMPORT_REF void OsclActiveObject::AddToScheduler ()

Add this exec object to the current thread's scheduler.

Reimplemented from [PVActiveBase](#).

7.91.4.2 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 [PActiveBase](#).

7.91.4.3 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 [PActiveBase](#).

Reimplemented in [OsclDNSRequestAO](#), and [OsclSocketRequestAO](#).

7.91.4.4 OSCL_IMPORT_REF bool OsclActiveObject::IsBusy ()

Return true if this AO is pending, false otherwise.

7.91.4.5 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.

7.91.4.6 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.

7.91.4.7 OSCL_IMPORT_REF int32 OsclActiveObject::Priority ()

Return scheduling priority of this exec object.

7.91.4.8 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 [PActiveBase](#).

7.91.4.9 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](#).

7.91.4.10 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.

7.91.4.11 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.

7.91.4.12 OSCL_IMPORT_REF void OsclActiveObject::SetStatus (int32)**7.91.4.13 OSCL_IMPORT_REF int32 OsclActiveObject::Status ()**

Request status access

7.91.4.14 OSCL_IMPORT_REF [OsclAOSatus&](#) OsclActiveObject::StatusRef ()

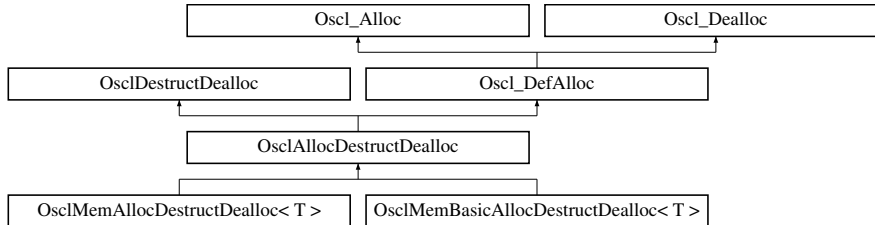
The documentation for this class was generated from the following file:

- [oscl_scheduler_ao.h](#)

7.92 OsciAllocDestructDealloc Class Reference

```
#include <osci_defalloc.h>
```

Inheritance diagram for OsciAllocDestructDealloc::



Public Methods

- virtual [~OsciAllocDestructDealloc \(\)](#)

7.92.1 Constructor & Destructor Documentation

7.92.1.1 `virtual OsciAllocDestructDealloc::~~OsciAllocDestructDealloc ()` [inline, virtual]

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

- [osci_defalloc.h](#)

7.93 OsclAOSStatus Class Reference

```
#include <oscl_aostatus.h>
```

Public Methods

- OSCL_INLINE [OsclAOSStatus](#) ()
- OSCL_INLINE [OsclAOSStatus](#) (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

7.93.1 Constructor & Destructor Documentation

7.93.1.1 OSCL_INLINE [OsclAOSStatus::OsclAOSStatus](#) ()

7.93.1.2 OSCL_INLINE [OsclAOSStatus::OsclAOSStatus](#) (int32 *aStatus*)

7.93.2 Member Function Documentation

7.93.2.1 OSCL_INLINE int32 [OsclAOSStatus::operator!=](#) (int32 *aStatus*) const

7.93.2.2 OSCL_INLINE int32 [OsclAOSStatus::operator<](#) (int32 *aStatus*) const

7.93.2.3 OSCL_INLINE int32 [OsclAOSStatus::operator<=](#) (int32 *aStatus*) const

7.93.2.4 OSCL_INLINE int32 [OsclAOSStatus::operator=](#) (int32 *aStatus*)

7.93.2.5 OSCL_INLINE int32 [OsclAOSStatus::operator==](#) (int32 *aStatus*) const

7.93.2.6 OSCL_INLINE int32 [OsclAOSStatus::operator>](#) (int32 *aStatus*) const

7.93.2.7 OSCL_INLINE int32 [OsclAOSStatus::operator>=](#) (int32 *aStatus*) const

7.93.2.8 OSCL_INLINE int32 [OsclAOSStatus::Value](#) ()

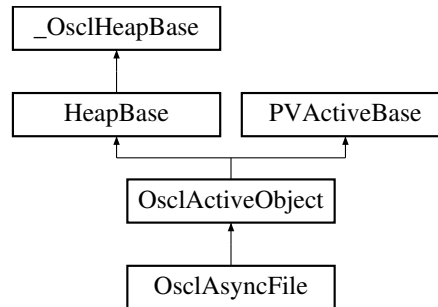
The documentation for this class was generated from the following file:

- [oscl_aostatus.h](#)

7.94 OsciAsyncFile Class Reference

```
#include <osci_file_async_read.h>
```

Inheritance diagram for OsciAsyncFile::



Public Methods

- [~OsciAsyncFile \(\)](#)
- [int32 Open \(const \[osci_wchar\]\(#\) *filename, uint32 mode, const \[OsciNativeFileParams\]\(#\) ¶ms, \[Osci_FileServer\]\(#\) &fileserv\)](#)
- [int32 Open \(const char *filename, uint32 mode, const \[OsciNativeFileParams\]\(#\) ¶ms, \[Osci_FileServer\]\(#\) &fileserv\)](#)
- [int32 Seek \(\[TOsciFileOffset\]\(#\) offset, \[Osci_File::seek_type\]\(#\) origin\)](#)
- [TOsciFileOffset Tell \(\)](#)
- [uint32 Read \(\[OsciAny\]\(#\) *aBuffer1, uint32 aDataSize, uint32 aNumElements\)](#)
- [int32 EndOfFile \(\)](#)
- [TOsciFileOffset Size \(\)](#)
- [int32 Close \(\)](#)
- [uint32 Write \(const \[OsciAny\]\(#\) *aBuffer1, uint32 aDataSize, uint32 aNumElements\)](#)
- [uint32 Flush \(\)](#)

Static Public Methods

- [OsciAsyncFile * NewL \(\[OsciNativeFile\]\(#\) &aAsyncFile, int32 aCacheSize, \[PVLogger\]\(#\) *\)](#)
- [void Delete \(OsciAsyncFile *\)](#)

Data Fields

- [uint32 iNumOfRun](#)
- [uint32 iNumOfRunErr](#)

7.94.1 Detailed Description

OsciAsyncFile

7.94.2 Constructor & Destructor Documentation

7.94.2.1 OsclAsyncFile::~OsclAsyncFile ()

Destructor.

7.94.3 Member Function Documentation

7.94.3.1 int32 OsclAsyncFile::Close ()

7.94.3.2 void OsclAsyncFile::Delete (OsclAsyncFile *) [static]

7.94.3.3 int32 OsclAsyncFile::EndOfFile ()

7.94.3.4 uint32 OsclAsyncFile::Flush () [inline]

7.94.3.5 OsclAsyncFile* OsclAsyncFile::NewL (OsclNativeFile & aAsyncFile, int32 aCacheSize, PVLogger *) [static]

Two-phased constructor.

Parameters:

aAsyncFile: 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.

- 7.94.3.6 `int32 OsclAsyncFile::Open (const char * filename, uint32 mode, const OsclNativeFileParams & params, Oscl_FileServer & fileserv)`
- 7.94.3.7 `int32 OsclAsyncFile::Open (const oscl_wchar * filename, uint32 mode, const OsclNativeFileParams & params, Oscl_FileServer & fileserv)`
- 7.94.3.8 `uint32 OsclAsyncFile::Read (OsclAny * aBuffer1, uint32 aDataSize, uint32 aNumElements)`
- 7.94.3.9 `int32 OsclAsyncFile::Seek (TOsclFileOffset offset, Oscl_File::seek_type origin)`
- 7.94.3.10 `TOsclFileOffset OsclAsyncFile::Size ()`
- 7.94.3.11 `TOsclFileOffset OsclAsyncFile::Tell ()`
- 7.94.3.12 `uint32 OsclAsyncFile::Write (const OsclAny * aBuffer1, uint32 aDataSize, uint32 aNumElements) [inline]`

7.94.4 Field Documentation

- 7.94.4.1 `uint32 OsclAsyncFile::iNumOfRun`
- 7.94.4.2 `uint32 OsclAsyncFile::iNumOfRunErr`

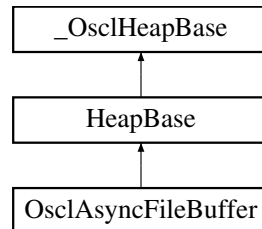
The documentation for this class was generated from the following file:

- [oscl_file_async_read.h](#)

7.95 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)

7.95.1 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.

7.95.2 Constructor & Destructor Documentation

7.95.2.1 `OsclAsyncFileBuffer::~OsclAsyncFileBuffer ()`

7.95.3 Member Function Documentation

7.95.3.1 `OsclBuf* OsclAsyncFileBuffer::Buffer ()`

7.95.3.2 `void OsclAsyncFileBuffer::CleanInUse () [inline]`

7.95.3.3 `bool OsclAsyncFileBuffer::HasThisOffset (TOsclFileOffset aOffset)`

7.95.3.4 `int32 OsclAsyncFileBuffer::Id () [inline]`

7.95.3.5 `bool OsclAsyncFileBuffer::IsInUse () [inline]`

7.95.3.6 `bool OsclAsyncFileBuffer::IsValid () [inline]`

7.95.3.7 `int32 OsclAsyncFileBuffer::Length () [inline]`

7.95.3.8 `OsclAsyncFileBuffer* OsclAsyncFileBuffer::NewL (int32 aBufferSize, int32 aId)`
[static]

7.95.3.9 `TOsclFileOffset OsclAsyncFileBuffer::Offset () [inline]`

7.95.3.10 `void OsclAsyncFileBuffer::SetInUse () [inline]`

7.95.3.11 `void OsclAsyncFileBuffer::SetOffset (TOsclFileOffset aOffset) [inline]`

7.95.3.12 `void OsclAsyncFileBuffer::StartAsyncRead (bool aStartAsyncRead)`

7.95.3.13 `void OsclAsyncFileBuffer::UpdateData ()`

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

- [oscl_file_async_read.h](#)

7.96 OsclAuditCB Class Reference

```
#include <oscl_mem.h>
```

Public Methods

- [OsclAuditCB \(\)](#)
- [OsclAuditCB \(const \[OsclMemStatsNode\]\(#\) *myStatsNode, \[OsclMemAudit\]\(#\) *ptr\)](#)

Data Fields

- const [OsclMemStatsNode](#) * pStatsNode
- [OsclMemAudit](#) * pAudit

7.96.1 Constructor & Destructor Documentation

7.96.1.1 [OsclAuditCB::OsclAuditCB \(\)](#) [inline]

7.96.1.2 [OsclAuditCB::OsclAuditCB \(const \[OsclMemStatsNode\]\(#\) * myStatsNode, \[OsclMemAudit\]\(#\) * ptr\)](#) [inline]

7.96.2 Field Documentation

7.96.2.1 [OsclMemAudit*](#) [OsclAuditCB::pAudit](#)

7.96.2.2 const [OsclMemStatsNode*](#) [OsclAuditCB::pStatsNode](#)

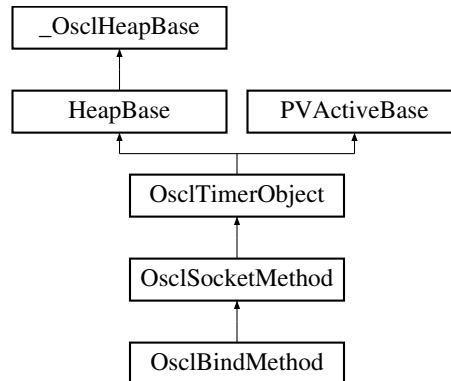
The documentation for this class was generated from the following file:

- [oscl_mem.h](#)

7.97 OsciBindMethod Class Reference

```
#include <osci_socket_bind.h>
```

Inheritance diagram for OsciBindMethod::



Public Methods

- [~OsciBindMethod \(\)](#)
- [TPVSocketEvent Bind \(OsciNetworkAddress &aAddress, int32 aTimeout\)](#)
- [OsciBindRequest * BindRequest \(\)](#)

Static Public Methods

- [OsciBindMethod * NewL \(OsciIPSocketI &c\)](#)

7.97.1 Constructor & Destructor Documentation

7.97.1.1 [OsciBindMethod::~~OsciBindMethod \(\)](#)

7.97.2 Member Function Documentation

7.97.2.1 [TPVSocketEvent OsciBindMethod::Bind \(OsciNetworkAddress &aAddress, int32 aTimeout\)](#)

7.97.2.2 [OsciBindRequest* OsciBindMethod::BindRequest \(\)](#) [inline]

7.97.2.3 [OsciBindMethod* OsciBindMethod::NewL \(OsciIPSocketI &c\)](#) [static]

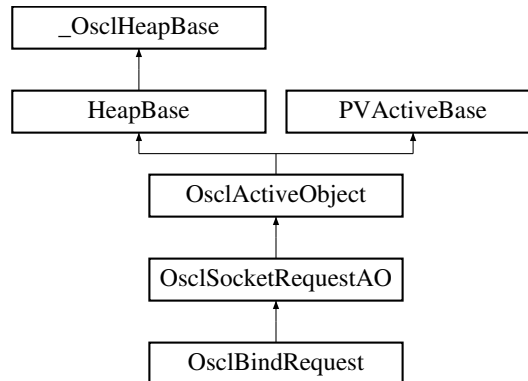
The documentation for this class was generated from the following file:

- [osci_socket_bind.h](#)

7.98 OsciBindRequest Class Reference

```
#include <osci_socket_bind.h>
```

Inheritance diagram for OsciBindRequest::



Public Methods

- [OsciBindRequest](#) ([OsciSocketMethod](#) &c)
- void [Bind](#) ([OsciNetworkAddress](#) &aAddress)

7.98.1 Detailed Description

This is the AO that interacts with the socket server

7.98.2 Constructor & Destructor Documentation

7.98.2.1 [OsciBindRequest::OsciBindRequest](#) ([OsciSocketMethod](#) & c) [inline]

7.98.3 Member Function Documentation

7.98.3.1 void [OsciBindRequest::Bind](#) ([OsciNetworkAddress](#) & aAddress)

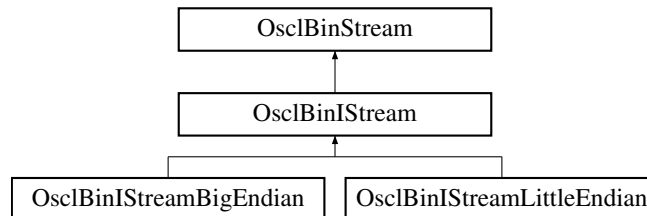
The documentation for this class was generated from the following file:

- [osci_socket_bind.h](#)

7.99 OsciBinIStream Class Reference

```
#include <osci_bin_stream.h>
```

Inheritance diagram for OsciBinIStream::



Public Methods

- [OsciBinIStream \(\)](#)
- [~OsciBinIStream \(\)](#)
- `uint8` [Read_uint8 \(\)](#)

This method reads an unsigned short from the stream.

- `OsciBinIStream &` [get \(int8 *data, int32 size\)](#)

This method reads 'length' number of bytes from the stream and places them in 'data'.

7.99.1 Constructor & Destructor Documentation

7.99.1.1 `OsciBinIStream::OsciBinIStream ()` [inline]

7.99.1.2 `OsciBinIStream::~~OsciBinIStream ()` [inline]

7.99.2 Member Function Documentation

7.99.2.1 `OsciBinIStream& OsciBinIStream::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

7.99.2.2 `uint8 OsciBinIStream::Read_uint8 ()`

This method reads an unsigned short from the stream.

Returns:

Unsigned short read from the stream.

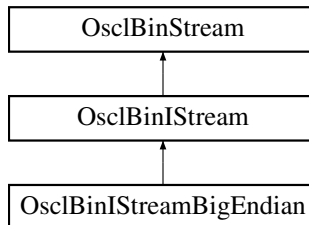
The documentation for this class was generated from the following file:

- [osci_bin_stream.h](#)

7.100 OsciBinIStreamBigEndian Class Reference

```
#include <osci_bin_stream.h>
```

Inheritance diagram for OsciBinIStreamBigEndian::



Public Methods

- [OsciBinIStreamBigEndian](#) ()
- void [Read](#) (int8 &data)
- void [Read](#) (uint8 &data)
- void [Read](#) (int16 &data)
- void [Read](#) (uint16 &data)
- void [Read](#) (int32 &data)
- void [Read](#) (uint32 &data)
- OsciBinIStreamBigEndian & [operator>>](#) (int8 &data)
This method reads a int8 from the stream and stores it in 'data'.
- OsciBinIStream & [operator>>](#) (uint8 &data)
This method reads a uint8 from the stream and stores it in 'data'.
- OsciBinIStreamBigEndian & [operator>>](#) (int16 &data)
This method reads a int16 from the stream and stores it in 'data'.
- OsciBinIStreamBigEndian & [operator>>](#) (uint16 &data)
This method reads a uint16 from the stream and stores it in 'data'.
- OsciBinIStreamBigEndian & [operator>>](#) (int32 &data)
This method reads a int32 from the stream and stores it in 'data'.
- OsciBinIStreamBigEndian & [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.

7.100.1 Constructor & Destructor Documentation

7.100.1.1 `OslBinIStreamBigEndian::OslBinIStreamBigEndian ()` [inline]

7.100.2 Member Function Documentation

7.100.2.1 `OslBinIStreamBigEndian& OslBinIStreamBigEndian::operator>> (uint32 & data)`

This method reads a uint32 from the stream and stores it in 'data'.

7.100.2.2 `OslBinIStreamBigEndian& OslBinIStreamBigEndian::operator>> (int32 & data)`

This method reads a int32 from the stream and stores it in 'data'.

7.100.2.3 `OslBinIStreamBigEndian& OslBinIStreamBigEndian::operator>> (uint16 & data)`

This method reads a uint16 from the stream and stores it in 'data'.

7.100.2.4 `OslBinIStreamBigEndian& OslBinIStreamBigEndian::operator>> (int16 & data)`

This method reads a int16 from the stream and stores it in 'data'.

7.100.2.5 `OslBinIStream& OslBinIStreamBigEndian::operator>> (uint8 & data)`

This method reads a uint8 from the stream and stores it in 'data'.

7.100.2.6 `OslBinIStreamBigEndian& OslBinIStreamBigEndian::operator>> (int8 & data)`

This method reads a int8 from the stream and stores it in 'data'.

7.100.2.7 `void OslBinIStreamBigEndian::Read (uint32 & data)`

7.100.2.8 `void OslBinIStreamBigEndian::Read (int32 & data)`

7.100.2.9 `void OslBinIStreamBigEndian::Read (uint16 & data)`

7.100.2.10 `void OslBinIStreamBigEndian::Read (int16 & data)`

7.100.2.11 `void OslBinIStreamBigEndian::Read (uint8 & data)`

7.100.2.12 `void OslBinIStreamBigEndian::Read (int8 & data)`

7.100.2.13 `uint16 OslBinIStreamBigEndian::Read_uint16 ()`

This method reads an unsigned short from the stream.

Returns:

Unsigned short read from the stream.

7.100.2.14 uint32 OslBinIStreamBigEndian::Read_uint32 ()

This method reads an unsigned long from the stream.

Returns:

unsigned long read from the stream.

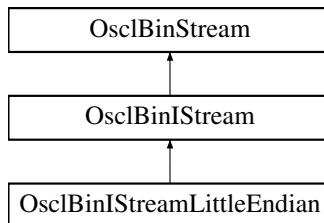
The documentation for this class was generated from the following file:

- [oscl_bin_stream.h](#)

7.101 OsciBinInputStreamLittleEndian Class Reference

```
#include <osci_bin_stream.h>
```

Inheritance diagram for OsciBinInputStreamLittleEndian::



Public Methods

- [OsciBinInputStreamLittleEndian \(\)](#)
- [OsciBinInputStreamLittleEndian & operator>> \(int8 &data\)](#)
This method reads a int8 from the stream and stores it in 'data'.
- [OsciBinInputStreamLittleEndian & operator>> \(uint8 &data\)](#)
This method reads a uint8 from the stream and stores it in 'data'.
- [OsciBinInputStreamLittleEndian & operator>> \(int16 &data\)](#)
This method reads a int16 from the stream and stores it in 'data'.
- [OsciBinInputStreamLittleEndian & operator>> \(uint16 &data\)](#)
This method reads a uint16 from the stream and stores it in 'data'.
- [OsciBinInputStreamLittleEndian & operator>> \(int32 &data\)](#)
This method reads a int32 from the stream and stores it in 'data'.
- [OsciBinInputStreamLittleEndian & operator>> \(uint32 &data\)](#)
This method reads a uint32 from the stream and stores it in 'data'.

Protected Methods

- uint16 [Read_uint16 \(\)](#)
- uint32 [Read_uint32 \(\)](#)

7.101.1 Constructor & Destructor Documentation

7.101.1.1 `OsciBinInputStreamLittleEndian::OsciBinInputStreamLittleEndian ()` [inline]

7.101.2 Member Function Documentation

7.101.2.1 `OsciBinInputStreamLittleEndian& OsciBinInputStreamLittleEndian::operator>> (uint32 & data)`

This method reads a uint32 from the stream and stores it in 'data'.

7.101.2.2 `OsciBinInputStreamLittleEndian& OsciBinInputStreamLittleEndian::operator>> (int32 & data)`

This method reads a int32 from the stream and stores it in 'data'.

7.101.2.3 `OsciBinInputStreamLittleEndian& OsciBinInputStreamLittleEndian::operator>> (uint16 & data)`

This method reads a uint16 from the stream and stores it in 'data'.

7.101.2.4 `OsciBinInputStreamLittleEndian& OsciBinInputStreamLittleEndian::operator>> (int16 & data)`

This method reads a int16 from the stream and stores it in 'data'.

7.101.2.5 `OsciBinInputStreamLittleEndian& OsciBinInputStreamLittleEndian::operator>> (uint8 & data)`

This method reads a uint8 from the stream and stores it in 'data'.

7.101.2.6 `OsciBinInputStreamLittleEndian& OsciBinInputStreamLittleEndian::operator>> (int8 & data)`

This method reads a int8 from the stream and stores it in 'data'.

7.101.2.7 `uint16 OsciBinInputStreamLittleEndian::Read_uint16 ()` [protected]

7.101.2.8 `uint32 OsciBinInputStreamLittleEndian::Read_uint32 ()` [protected]

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

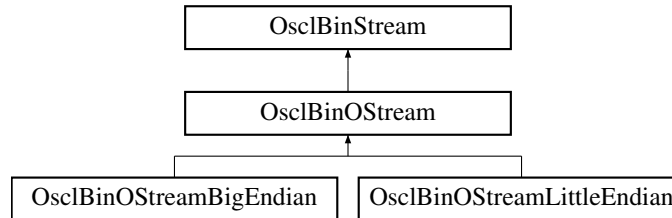
- [osci_bin_stream.h](#)

7.102 OsciBinOStream Class Reference

Class OsciBinOStream implements the basic stream functions for an output stream.

```
#include <osci_bin_stream.h>
```

Inheritance diagram for OsciBinOStream::



Public Methods

- [OsciBinOStream \(\)](#)
- virtual [~OsciBinOStream \(\)](#)
- OsciBinOStream & [write](#) (const int8 *data, int32 size)

This method writes 'length' number of bytes stored in 'data' to the stream.

7.102.1 Detailed Description

Class OsciBinOStream implements the basic stream functions for an output stream.

7.102.2 Constructor & Destructor Documentation

7.102.2.1 `OsciBinOStream::OsciBinOStream () [inline]`

7.102.2.2 `virtual OsciBinOStream::~~OsciBinOStream () [inline, virtual]`

7.102.3 Member Function Documentation

7.102.3.1 `OsciBinOStream& OsciBinOStream::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:

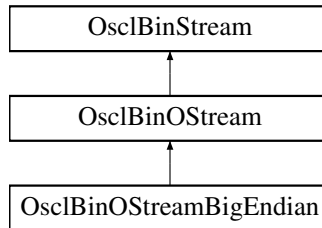
- [osci_bin_stream.h](#)

7.103 OslBinOStreamBigEndian Class Reference

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

```
#include <osl_bin_stream.h>
```

Inheritance diagram for OslBinOStreamBigEndian::



Public Methods

- [OslBinOStreamBigEndian \(\)](#)
- [OslBinOStreamBigEndian & operator<< \(const int8 &data\)](#)
This method writes a int8 from 'data' to the stream.
- [OslBinOStreamBigEndian & operator<< \(const uint8 &data\)](#)
This method writes a uint8 from 'data' to the stream.
- [OslBinOStreamBigEndian & operator<< \(const int16 &data\)](#)
This method writes a int16 from 'data' to the stream.
- [OslBinOStreamBigEndian & operator<< \(const uint16 &data\)](#)
This method writes a uint16 from 'data' to the stream.
- [OslBinOStreamBigEndian & operator<< \(const int32 &data\)](#)
This method writes a int32 from 'data' to the stream.
- [OslBinOStreamBigEndian & 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)

7.103.1 Detailed Description

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

7.103.2 Constructor & Destructor Documentation

7.103.2.1 `OsclBinOStreamBigEndian::OsclBinOStreamBigEndian ()` [`inline`]

7.103.3 Member Function Documentation

7.103.3.1 `OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const uint32 & data)`

This method writes a uint32 from 'data' to the stream.

7.103.3.2 `OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const int32 & data)`

This method writes a int32 from 'data' to the stream.

7.103.3.3 `OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const uint16 & data)`

This method writes a uint16 from 'data' to the stream.

7.103.3.4 `OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const int16 & data)`

This method writes a int16 from 'data' to the stream.

7.103.3.5 `OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const uint8 & data)`

This method writes a uint8 from 'data' to the stream.

7.103.3.6 `OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const int8 & data)`

This method writes a int8 from 'data' to the stream.

7.103.3.7 `void OsclBinOStreamBigEndian::WriteUnsignedLong (const uint32 data)`
[protected]

7.103.3.8 `void OsclBinOStreamBigEndian::WriteUnsignedShort (const uint16 data)`
[protected]

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

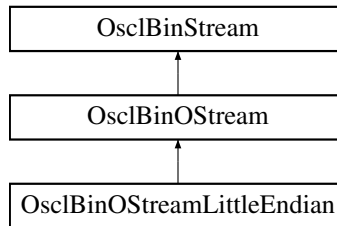
- [oscl_bin_stream.h](#)

7.104 OsciBinOutputStreamLittleEndian Class Reference

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

```
#include <osci_bin_stream.h>
```

Inheritance diagram for OsciBinOutputStreamLittleEndian::



Public Methods

- [OsciBinOutputStreamLittleEndian \(\)](#)
- [OsciBinOutputStreamLittleEndian & operator<< \(const int8 &data\)](#)
This method writes a int8 from 'data' to the stream.
- [OsciBinOutputStreamLittleEndian & operator<< \(const uint8 &data\)](#)
This method writes a uint8 from 'data' to the stream.
- [OsciBinOutputStreamLittleEndian & operator<< \(const int16 &data\)](#)
This method writes a int16 from 'data' to the stream.
- [OsciBinOutputStreamLittleEndian & operator<< \(const uint16 &data\)](#)
This method writes a uint16 from 'data' to the stream.
- [OsciBinOutputStreamLittleEndian & operator<< \(const int32 &data\)](#)
This method writes a int32 from 'data' to the stream.
- [OsciBinOutputStreamLittleEndian & 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.

7.104.1 Detailed Description

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

7.104.2 Constructor & Destructor Documentation

7.104.2.1 `OsciBinOStreamLittleEndian::OsciBinOStreamLittleEndian ()` `[inline]`

7.104.3 Member Function Documentation

7.104.3.1 `OsciBinOStreamLittleEndian& OsciBinOStreamLittleEndian::operator<< (const uint32 & data)`

This method writes a uint32 from 'data' to the stream.

7.104.3.2 `OsciBinOStreamLittleEndian& OsciBinOStreamLittleEndian::operator<< (const int32 & data)`

This method writes a int32 from 'data' to the stream.

7.104.3.3 `OsciBinOStreamLittleEndian& OsciBinOStreamLittleEndian::operator<< (const uint16 & data)`

This method writes a uint16 from 'data' to the stream.

7.104.3.4 `OsciBinOStreamLittleEndian& OsciBinOStreamLittleEndian::operator<< (const int16 & data)`

This method writes a int16 from 'data' to the stream.

7.104.3.5 `OsciBinOStreamLittleEndian& OsciBinOStreamLittleEndian::operator<< (const uint8 & data)`

This method writes a uint8 from 'data' to the stream.

7.104.3.6 `OsciBinOStreamLittleEndian& OsciBinOStreamLittleEndian::operator<< (const int8 & data)`

This method writes a int8 from 'data' to the stream.

7.104.3.7 `void OsciBinOStreamLittleEndian::WriteUnsignedLong (const uint32 data)`
`[protected]`

This method writes 'data' (unsigned long) to the stream.

7.104.3.8 `void OsciBinOStreamLittleEndian::WriteUnsignedShort (const uint16 data)`
`[protected]`

This method writes 'data' (unsigned short) to the stream.

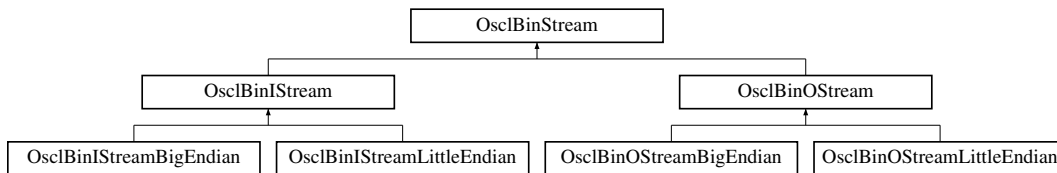
The documentation for this class was generated from the following file:

- [osci_bin_stream.h](#)

7.105 OsciBinStream Class Reference

```
#include <osci_bin_stream.h>
```

Inheritance diagram for OsciBinStream::



Public Methods

- [OsciBinStream](#) ()
- 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 methods specifies the data buffer to attach to the stream.
- void [Attach](#) (const uint32 numFragments, const [OsciMemoryFragment](#) *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 [OsciMemoryFragment](#) * *nextFragPtr*
- [int](#) *fragsLeft*
- const [OsciMemoryFragment](#) * *firstFragPtr*
- [int](#) *numFrag*s
- [OsciMemoryFragment](#) *specialFragBuffer*

7.105.1 Member Enumeration Documentation

7.105.1.1 enum [OsciBinStream::state_t](#) [protected]

Enumeration values:

[GOOD_STATE](#)

[EOF_STATE](#)

[FAIL_STATE](#)

7.105.2 Constructor & Destructor Documentation

7.105.2.1 [OsciBinStream::OsciBinStream\(\)](#) [inline]

7.105.3 Member Function Documentation

7.105.3.1 void [OsciBinStream::Attach](#) (const [uint32](#) *numFragments*, const [OsciMemoryFragment](#) * *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

7.105.3.2 void [OsciBinStream::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

7.105.3.3 bool OsciBinStream::eof ()

This method determines if end of stream has been reached.

Returns:

true if end of stream has been reached.

7.105.3.4 bool OsciBinStream::fail ()

This method determines if an error has occurred in the stream.

Returns:

true if an error occurred in the stream.

7.105.3.5 bool OsciBinStream::good ()

This method determines if the stream is ok.

Returns:

true if stream is ok.

7.105.3.6 bool OsciBinStream::HaveRoomInCurrentBlock (uint32 *size*) [protected]**7.105.3.7 uint32 OsciBinStream::PositionInBlock ()**

This method returns the current stream position.

Returns:

stream position.

7.105.3.8 bool OsciBinStream::ReserveSpace (uint32 *size*) [protected]**7.105.3.9 void OsciBinStream::Seek (uint32 *absPosition*)**

This method seeks to the specified stream position.

Returns:

Stream position.

7.105.3.10 void OsciBinStream::seekFromCurrentPosition (int32 *offset*)

This method seeks to the specified offset from the current location.

Parameters:

offset from current stream location

7.105.3.11 uint32 OsciBinStream::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.

7.105.4 Field Documentation

7.105.4.1 **const OsciMemoryFragment*** OsciBinStream::firstFragPtr [protected]

7.105.4.2 **int** OsciBinStream::fragsLeft [protected]

7.105.4.3 **uint32** OsciBinStream::length [protected]

7.105.4.4 **const OsciMemoryFragment*** OsciBinStream::nextFragPtr [protected]

7.105.4.5 **int** OsciBinStream::numFrag [protected]

7.105.4.6 **uint8*** OsciBinStream::pBasePosition [protected]

7.105.4.7 **uint8*** OsciBinStream::pPosition [protected]

7.105.4.8 **OsciMemoryFragment** OsciBinStream::specialFragBuffer [protected]

7.105.4.9 **state_t** OsciBinStream::state [protected]

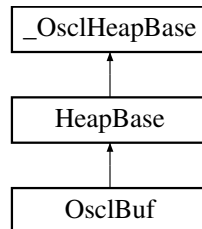
The documentation for this class was generated from the following file:

- [osci_bin_stream.h](#)

7.106 OsciBuf Class Reference

```
#include <osci_file_async_read.h>
```

Inheritance diagram for OsciBuf::



Public Methods

- [OsciBuf](#) (int32 size)
- int32 [Length](#) ()
- [OsciPtr Des](#) ()
- [OsciPtrC DesC](#) ()

Static Public Methods

- OsciBuf * [NewL](#) (int32 size)
- void [Delete](#) (OsciBuf *a)

Data Fields

- uint8 * [iBuffer](#)
- int32 [iMaxLength](#)
- int32 [iLength](#)

7.106.1 Constructor & Destructor Documentation

7.106.1.1 `OsciBuf::OsciBuf (int32 size)` [inline]

7.106.2 Member Function Documentation

7.106.2.1 `void OsciBuf::Delete (OsciBuf * a)` [inline, static]

7.106.2.2 `OsciPtr OsciBuf::Des ()` [inline]

7.106.2.3 `OsciPtrC OsciBuf::DesC ()` [inline]

7.106.2.4 `int32 OsciBuf::Length ()` [inline]

7.106.2.5 `OsciBuf* OsciBuf::NewL (int32 size)` [inline, static]

7.106.3 Field Documentation

7.106.3.1 `uint8* OsciBuf::iBuffer`

7.106.3.2 `int32 OsciBuf::iLength`

7.106.3.3 `int32 OsciBuf::iMaxLength`

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

- [osci_file_async_read.h](#)

7.107 OslCompareLess< T > Class Template Reference

```
#include <oscl_priqueue.h>
```

Public Methods

- int [compare](#) (T &a, T &b) const

```
template<class T> class OslCompareLess< T >
```

7.107.1 Member Function Documentation

7.107.1.1 `template<class T> int OslCompareLess< T >::compare (T & a, T & b) const`
[inline]

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

- [oscl_priqueue.h](#)

7.108 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 \(OSCL_String &, Oscl_Vector< OsclRegistryAccessElement, OsclMem-Allocator > &\)](#)
- [void OpenSession \(\)](#)
- [void CloseSession \(\)](#)

Data Fields

- [OsclComponentRegistryData iData](#)
- [OsclMutex iMutex](#)
- [uint32 iComponentIdCounter](#)
- [uint32 iNumSessions](#)

7.108.1 Detailed Description

Thread-safe singleton registry object.

7.108.2 Constructor & Destructor Documentation

7.108.2.1 `OslComponentRegistry::OslComponentRegistry ()`

7.108.2.2 `OslComponentRegistry::~~OslComponentRegistry ()`

7.108.3 Member Function Documentation

7.108.3.1 `void OslComponentRegistry::CloseSession ()`

7.108.3.2 `OslComponentFactory OslComponentRegistry::FindExact (OSCL_String &)`

7.108.3.3 `void OslComponentRegistry::FindHierarchical (OSCL_String &, Osl_Vector< OslRegistryAccessElement, OslMemAllocator > &)`

7.108.3.4 `void OslComponentRegistry::OpenSession ()`

7.108.3.5 `int32 OslComponentRegistry::Register (uint32 & aId, OSCL_String &, OslComponentFactory)`

7.108.3.6 `int32 OslComponentRegistry::Unregister (uint32)`

7.108.3.7 `int32 OslComponentRegistry::Unregister (OSCL_String &)`

7.108.4 Field Documentation

7.108.4.1 `uint32 OslComponentRegistry::iComponentIdCounter`

7.108.4.2 `OslComponentRegistryData OslComponentRegistry::iData`

7.108.4.3 `OslMutex OslComponentRegistry::iMutex`

7.108.4.4 `uint32 OslComponentRegistry::iNumSessions`

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

- [oscl_registry_serv_impl.h](#)

7.109 OslComponentRegistryData Class Reference

```
#include <oscl_registry_serv_impl.h>
```

Public Methods

- [OslComponentRegistryElement](#) * Find (OSCL_String &, bool aExact)

Data Fields

- [Osl_Vector](#)< [OslComponentRegistryElement](#), [OslMemAllocator](#) > iVec

7.109.1 Detailed Description

Registry

7.109.2 Member Function Documentation

- 7.109.2.1 [OslComponentRegistryElement](#)* [OslComponentRegistryData::Find](#) ([OSCL_String](#) &, bool *aExact*)

7.109.3 Field Documentation

- 7.109.3.1 [Osl_Vector](#)<[OslComponentRegistryElement](#), [OslMemAllocator](#)> [OslComponentRegistryData::iVec](#)

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

- [oscl_registry_serv_impl.h](#)

7.110 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

7.110.1 Detailed Description

Data for each registered component.

7.110.2 Constructor & Destructor Documentation

7.110.2.1 [OsclComponentRegistryElement::OsclComponentRegistryElement](#) ([OSCL_String](#) &, [OsclComponentFactory](#))

7.110.2.2 [OsclComponentRegistryElement::OsclComponentRegistryElement](#) (const [OsclComponentRegistryElement](#) &)

7.110.2.3 [OsclComponentRegistryElement::~~OsclComponentRegistryElement](#) ()

7.110.3 Member Function Documentation

7.110.3.1 bool [OsclComponentRegistryElement::Match](#) ([OSCL_String](#) & aStr, bool aExact)

7.110.3.2 [OsclComponentRegistryElement&](#) [OsclComponentRegistryElement::operator=](#) (const [OsclComponentRegistryElement](#) & src)

7.110.4 Field Documentation

7.110.4.1 uint32 [OsclComponentRegistryElement::iComponentId](#)

7.110.4.2 [OsclComponentFactory](#) [OsclComponentRegistryElement::iFactory](#)

7.110.4.3 [OSCL_String*](#) [OsclComponentRegistryElement::iId](#)

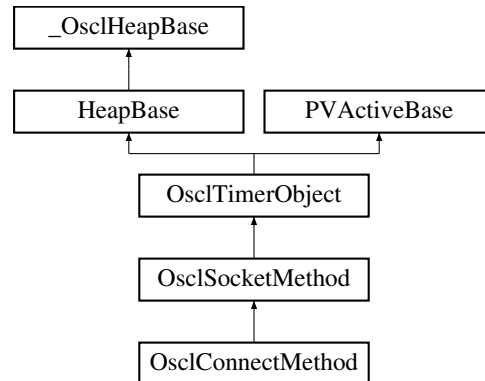
The documentation for this class was generated from the following file:

- [oscl_registry_serv_impl.h](#)

7.111 OsciConnectMethod Class Reference

```
#include <osci_socket_connect.h>
```

Inheritance diagram for OsciConnectMethod::



Public Methods

- [~OsciConnectMethod \(\)](#)
- [TPVSocketEvent Connect \(OsciNetworkAddress &aAddress, int32 aTimeout\)](#)
- [OsciConnectRequest * ConnectRequest \(\)](#)

Static Public Methods

- [OsciConnectMethod * NewL \(OsciIPSocketI &c\)](#)

7.111.1 Constructor & Destructor Documentation

7.111.1.1 [OsciConnectMethod::~~OsciConnectMethod \(\)](#)

7.111.2 Member Function Documentation

7.111.2.1 [TPVSocketEvent OsciConnectMethod::Connect \(OsciNetworkAddress &aAddress, int32 aTimeout\)](#)

7.111.2.2 [OsciConnectRequest* OsciConnectMethod::ConnectRequest \(\) \[inline\]](#)

7.111.2.3 [OsciConnectMethod* OsciConnectMethod::NewL \(OsciIPSocketI &c\) \[static\]](#)

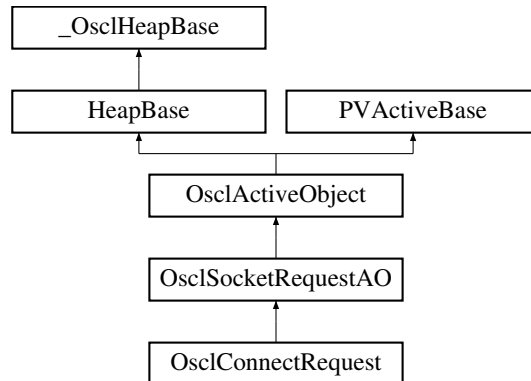
The documentation for this class was generated from the following file:

- [osci_socket_connect.h](#)

7.112 OsciConnectRequest Class Reference

```
#include <osci_socket_connect.h>
```

Inheritance diagram for OsciConnectRequest::



Public Methods

- [OsciConnectRequest](#) ([OsciSocketMethod](#) &c)
- void [Connect](#) ([OsciNetworkAddress](#) &aAddress)

7.112.1 Detailed Description

This is the AO that interacts with the socket server

7.112.2 Constructor & Destructor Documentation

7.112.2.1 [OsciConnectRequest::OsciConnectRequest](#) ([OsciSocketMethod](#) & c) [inline]

7.112.3 Member Function Documentation

7.112.3.1 void [OsciConnectRequest::Connect](#) ([OsciNetworkAddress](#) & aAddress)

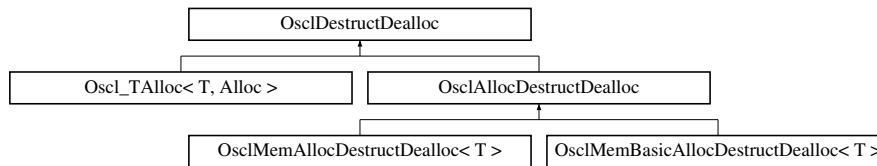
The documentation for this class was generated from the following file:

- [osci_socket_connect.h](#)

7.113 OsciDestructDealloc Class Reference

```
#include <osci_defalloc.h>
```

Inheritance diagram for OsciDestructDealloc::



Public Methods

- virtual void [destruct_and_dealloc](#) ([OsciAny](#) *ptr)=0

7.113.1 Member Function Documentation

7.113.1.1 virtual void [OsciDestructDealloc::destruct_and_dealloc](#) ([OsciAny](#) * ptr) [pure virtual]

Implemented in [Osci_TAlloc< T, Alloc >](#), [OsciMemAllocDestructDealloc< T >](#), [OsciMemBasicAllocDestructDealloc< T >](#), [Osci_TAlloc< entry_type, Alloc >](#), [Osci_TAlloc< node_type, TagTree_Allocator >](#), [Osci_TAlloc< node_type, alloc_type >](#), [Osci_TAlloc< MM_StatsNodeTagTreeType, OsciMemBasicAllocator >](#), [Osci_TAlloc< char, alloc_type >](#), [Osci_TAlloc< tag_base_unit, Alloc >](#), [Osci_TAlloc< PVLogger, alloc_type >](#), and [Osci_TAlloc< node_type, Alloc >](#).

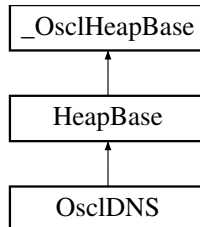
The documentation for this class was generated from the following file:

- [osci_defalloc.h](#)

7.114 OsciDNS Class Reference

```
#include <osci_dns.h>
```

Inheritance diagram for OsciDNS::



Public Methods

- OSCI_IMPORT_REF [~OsciDNS \(\)](#)
- OSCI_IMPORT_REF [TPVDNSEvent GetHostByName](#) (char *name, [OsciNetworkAddress](#) &addr, int32 aTimeoutMsec=-1, [Osci_Vector](#)< [OsciNetworkAddress](#), [OsciMemAllocator](#) > *aAddressList=NULL)
- OSCI_IMPORT_REF void [CancelGetHostByName \(\)](#)

Static Public Methods

- OSCI_IMPORT_REF [OsciDNS * NewL](#) ([Osci_DefAlloc](#) &alloc, [OsciSocketServ](#) &aServ, [OsciDNSObserver](#) &aObserver, uint32 aId)

Friends

- class [OsciDNSRequestAO](#)

7.114.1 Detailed Description

The DNS class

7.114.2 Constructor & Destructor Documentation

7.114.2.1 OSCI_IMPORT_REF OsciDNS::~~OsciDNS ()

Destructor.

Note: the application must de-allocate the DNS object using the same allocator that was passed in the NewL object creation call.

7.114.3 Member Function Documentation

7.114.3.1 OSCI_IMPORT_REF void OsciDNS::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.

7.114.3.2 OSCL_IMPORT_REF TPVDNSEvent OsciDNS::GetHostByName (char * name, OsciNetworkAddress & addr, int32 aTimeoutMsec = -1, Osci_Vector< OsciNetworkAddress, OsciMemAllocator > * aAddressList = NULL)

GetHostByName. This is an asynchronous method.

Parameters:

- name*: Null-terminated string containing the host name.
- addr*: The output address corresponding to the host. 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.
- aAddressList* : A list of addresses for the host. @returns: EPVDNSPending for success, EPVDNS-Failure for failure.

7.114.3.3 OSCL_IMPORT_REF OsciDNS* OsciDNS::NewL (Osci_DefAlloc & alloc, OsciSocketServ & aServ, OsciDNSObserver & 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.

7.114.4 Friends And Related Function Documentation

7.114.4.1 friend class OsciDNSRequestAO [friend]

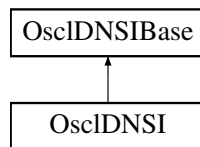
The documentation for this class was generated from the following file:

- [oscl_dns.h](#)

7.115 OsciDNSI Class Reference

```
#include <osci_dns_imp_pv.h>
```

Inheritance diagram for OsciDNSI::



Public Methods

- [~OsciDNSI \(\)](#)
- [int32 Open \(OsciSocketServI &aServer\)](#)
- [int32 Close \(\)](#)
- [void GetHostByName \(GetHostByNameParam &, OsciDNSRequestAO &\)](#)
- [void GetHostByNameSuccess \(GetHostByNameParam &\)](#)
- [void GetNextHost \(OsciDNSRequestAO &\)](#)
- [void GetNextHostSuccess \(GetHostByNameParam &\)](#)
- [bool GetHostByNameResponseContainsAliasInfo \(\)](#)

Static Public Methods

- [OsciDNSI * NewL \(Osci_DefAlloc &a\)](#)

Friends

- class [OsciDNSRequest](#)
- class [OsciGetHostByNameRequest](#)
- class [DNSRequestParam](#)

7.115.1 Detailed Description

OsciDNSI, non-Symbian implementation

7.115.2 Constructor & Destructor Documentation

7.115.2.1 OsciDNSI::~~OsciDNSI ()

7.115.3 Member Function Documentation

7.115.3.1 int32 OsciDNSI::Close () [virtual]

Implements [OsciDNSIBase](#).

7.115.3.2 void OsciDNSI::GetHostByName (GetHostByNameParam &, OsciDNSRequestAO &) [virtual]

Implements [OsciDNSIBase](#).

7.115.3.3 bool OsciDNSI::GetHostByNameResponseContainsAliasInfo () [virtual]

Implements [OsciDNSIBase](#).

7.115.3.4 void OsciDNSI::GetHostByNameSuccess (GetHostByNameParam &) [virtual]

Implements [OsciDNSIBase](#).

7.115.3.5 void OsciDNSI::GetNextHost (OsciDNSRequestAO &) [virtual]

Implements [OsciDNSIBase](#).

7.115.3.6 void OsciDNSI::GetNextHostSuccess (GetHostByNameParam &) [virtual]

Implements [OsciDNSIBase](#).

7.115.3.7 OsciDNSI* OsciDNSI::NewL (Osci_DefAlloc & a) [static]

7.115.3.8 int32 OsciDNSI::Open (OsciSocketServI & aServer) [virtual]

Implements [OsciDNSIBase](#).

7.115.4 Friends And Related Function Documentation

7.115.4.1 friend class DNSRequestParam [friend]

7.115.4.2 friend class OsciDNSRequest [friend]

Reimplemented from [OsciDNSIBase](#).

7.115.4.3 friend class OsciGetHostByNameRequest [friend]

Reimplemented from [OsciDNSIBase](#).

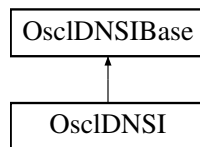
The documentation for this class was generated from the following file:

- [osci_dns_imp_pv.h](#)

7.116 OsciDNSIBase Class Reference

```
#include <osci_dns_imp_base.h>
```

Inheritance diagram for OsciDNSIBase::



Public Methods

- virtual `~OsciDNSIBase ()`
- virtual `int32 Open (OsciSocketServI &aServer)=0`
- virtual `int32 Close ()=0`
- virtual `void GetHostByName (GetHostByNameParam &, OsciDNSRequestAO &)=0`
- virtual `void GetHostByNameSuccess (GetHostByNameParam &)=0`
- virtual `bool GetHostByNameResponseContainsAliasInfo ()=0`
- virtual `void GetNextHost (OsciDNSRequestAO &)=0`
- virtual `void GetNextHostSuccess (GetHostByNameParam &)=0`
- `void CancelFxn (TPVDNSFxn)`

Protected Methods

- `OsciDNSIBase (Osci_DefAlloc &a)`
- virtual `bool IsReady (OsciDNSRequestAO &aObject)=0`
- virtual `void CancelGetHostByName ()=0`

Protected Attributes

- `Osci_DefAlloc & iAlloc`
- `OsciSocketServI * iSocketServ`

Friends

- class `OsciDNSRequest`
- class `OsciGetHostByNameRequest`

7.116.1 Detailed Description

OsciDNSIBase is a common base class for all implementations.

7.116.2 Constructor & Destructor Documentation

7.116.2.1 **virtual** OsciDNSIBase::~OsciDNSIBase () [virtual]

7.116.2.2 OsciDNSIBase::OsciDNSIBase (**Osci_DefAlloc** & *a*) [protected]

7.116.3 Member Function Documentation

7.116.3.1 **void** OsciDNSIBase::CancelFxn (**TPVDNSFxn**)

7.116.3.2 **virtual void** OsciDNSIBase::CancelGetHostByName () [protected, pure virtual]

7.116.3.3 **virtual int32** OsciDNSIBase::Close () [pure virtual]

Implemented in [OsciDNSI](#).

7.116.3.4 **virtual void** OsciDNSIBase::GetHostByName (**GetHostByNameParam** &, **OsciDNSRequestAO** &) [pure virtual]

Implemented in [OsciDNSI](#).

7.116.3.5 **virtual bool** OsciDNSIBase::GetHostByNameResponseContainsAliasInfo () [pure virtual]

Implemented in [OsciDNSI](#).

7.116.3.6 **virtual void** OsciDNSIBase::GetHostByNameSuccess (**GetHostByNameParam** &) [pure virtual]

Implemented in [OsciDNSI](#).

7.116.3.7 **virtual void** OsciDNSIBase::GetNextHost (**OsciDNSRequestAO** &) [pure virtual]

Implemented in [OsciDNSI](#).

7.116.3.8 **virtual void** OsciDNSIBase::GetNextHostSuccess (**GetHostByNameParam** &) [pure virtual]

Implemented in [OsciDNSI](#).

7.116.3.9 **virtual bool** OsciDNSIBase::IsReady (**OsciDNSRequestAO** & *aObject*) [protected, pure virtual]

7.116.3.10 **virtual int32** OsciDNSIBase::Open (**OsciSocketServI** & *aServer*) [pure virtual]

Implemented in [OsciDNSI](#).

7.116.4 Friends And Related Function Documentation

7.116.4.1 friend class OsciDNSRequest [friend]

Reimplemented in [OsciDNSI](#).

7.116.4.2 friend class OsciGetHostByNameRequest [friend]

Reimplemented in [OsciDNSI](#).

7.116.5 Field Documentation

7.116.5.1 [OsciDefAlloc](#)& OsciDNSIBase::iAlloc [protected]

7.116.5.2 [OsciSocketServI*](#) OsciDNSIBase::iSocketServ [protected]

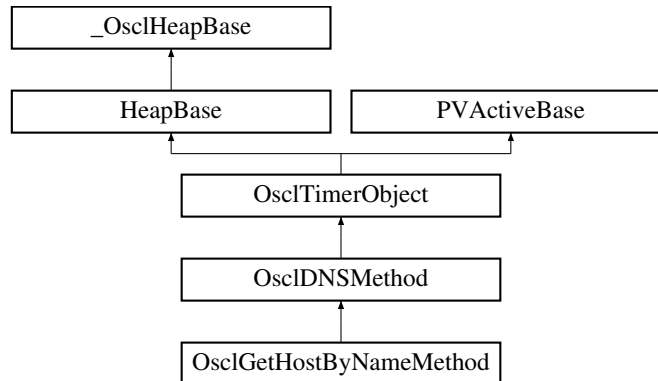
The documentation for this class was generated from the following file:

- [osci_dns_imp_base.h](#)

7.117 OsciDNSMethod Class Reference

```
#include <osci_dns_method.h>
```

Inheritance diagram for OsciDNSMethod::



Public Methods

- [OsciDNSMethod](#) ([Osci_DefAlloc](#) &a, const char *name, [TPVDNSFxn](#) fxn)
- void [Abort](#) ()
- void [AbortAll](#) ()
- void [CancelMethod](#) ()
- void [Run](#) ()

Data Fields

- [OsciDNSObserver](#) * [iDNSObserver](#)
- uint32 [iId](#)
- [Osci_DefAlloc](#) & [iAlloc](#)
- [TPVDNSFxn](#) [iDNSFxn](#)
- [PVLogger](#) * [iLogger](#)

Protected Methods

- void [ConstructL](#) ([OsciDNSObserver](#) *aObserver, [OsciDNSRequestAO](#) *aAO, uint32 aId)
- bool [StartMethod](#) (int32 aTimeoutMsec)
- void [MethodDone](#) ()

Protected Attributes

- [OsciDNSRequestAO](#) * [iDNSRequestAO](#)

7.117.1 Detailed Description

This is the base class for all socket methods. It provides the timeout on socket requests.

7.117.2 Constructor & Destructor Documentation

7.117.2.1 `OsciDNSMethod::OsciDNSMethod (Osci_DefAlloc & a, const char * name, TPVDNSFxn fn)` [inline]

7.117.3 Member Function Documentation

7.117.3.1 `void OsciDNSMethod::Abort ()`

7.117.3.2 `void OsciDNSMethod::AbortAll ()`

7.117.3.3 `void OsciDNSMethod::CancelMethod ()`

7.117.3.4 `void OsciDNSMethod::ConstructL (OsciDNSObserver * aObserver, OsciDNSRequestAO * aAO, uint32 aId)` [protected]

7.117.3.5 `void OsciDNSMethod::MethodDone ()` [protected]

7.117.3.6 `void OsciDNSMethod::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 [PActiveBase](#).

7.117.3.7 `bool OsciDNSMethod::StartMethod (int32 aTimeoutMsec)` [protected]

7.117.4 Field Documentation

7.117.4.1 `Osci_DefAlloc& OsciDNSMethod::iAlloc`

7.117.4.2 `TPVDNSFxn OsciDNSMethod::iDNSFxn`

7.117.4.3 `OsciDNSObserver* OsciDNSMethod::iDNSObserver`

7.117.4.4 `OsciDNSRequestAO* OsciDNSMethod::iDNSRequestAO` [protected]

7.117.4.5 `uint32 OsciDNSMethod::iId`

7.117.4.6 `PVLogger* OsciDNSMethod::iLogger`

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

- [osci_dns_method.h](#)

7.118 OsciDNSObserver Class Reference

```
#include <osci_dns.h>
```

Public Methods

- virtual OSCL_IMPORT_REF void [HandleDNSEvent](#) (int32 aId, [TPVDNSFxn](#) aFxn, [TPVDNSEvent](#) aEvent, int32 aError)=0
- virtual [~OsciDNSObserver](#) ()

7.118.1 Detailed Description

DNS event observer. The client implements this to get asynchronous command completion.

7.118.2 Constructor & Destructor Documentation

7.118.2.1 virtual [OsciDNSObserver::~OsciDNSObserver](#) () [inline, virtual]

7.118.3 Member Function Documentation

7.118.3.1 virtual OSCL_IMPORT_REF void [OsciDNSObserver::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:

- [osci_dns.h](#)

7.119 OsciDNSRequest Class Reference

```
#include <osci_dns_request.h>
```

Public Methods

- [OsciDNSRequest \(\)](#)
- [~OsciDNSRequest \(\)](#)
- void [CancelRequest \(\)](#)
- void [Complete](#) (bool, int32 aStatus, int32 aSockErr)
- void [Activate](#) ([DNSRequestParam](#) *iParam, [OsciDNSRequestAO](#) &a)

Data Fields

- [OsciDNSRequestAO](#) * [iDNSRequestAO](#)
- [DNSRequestParam](#) * [iDNSRequestParam](#)
- bool [iActive](#)

7.119.1 Detailed Description

This class defines the interface to the dns implementation threads.

7.119.2 Constructor & Destructor Documentation

7.119.2.1 [OsciDNSRequest::OsciDNSRequest \(\)](#) [[inline](#)]

7.119.2.2 [OsciDNSRequest::~~OsciDNSRequest \(\)](#) [[inline](#)]

7.119.3 Member Function Documentation

7.119.3.1 void [OsciDNSRequest::Activate](#) ([DNSRequestParam](#) * *iParam*, [OsciDNSRequestAO](#) & *a*)

7.119.3.2 void [OsciDNSRequest::CancelRequest \(\)](#)

7.119.3.3 void [OsciDNSRequest::Complete](#) (bool, int32 *aStatus*, int32 *aSockErr*)

7.119.4 Field Documentation

7.119.4.1 bool [OsciDNSRequest::iActive](#)

7.119.4.2 [OsciDNSRequestAO](#)* [OsciDNSRequest::iDNSRequestAO](#)

7.119.4.3 [DNSRequestParam](#)* [OsciDNSRequest::iDNSRequestParam](#)

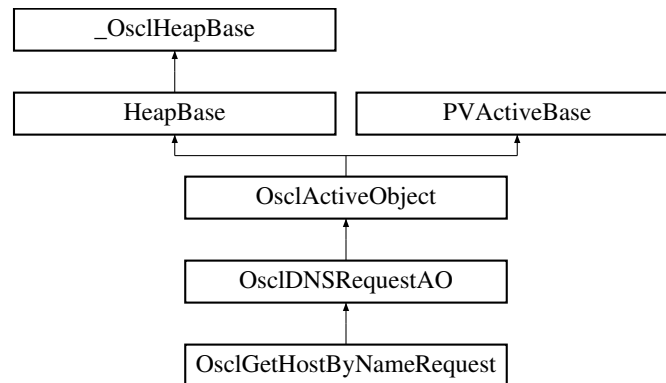
The documentation for this class was generated from the following file:

- [osci_dns_request.h](#)

7.120 OsciDNSRequestAO Class Reference

```
#include <osci_dns_method.h>
```

Inheritance diagram for OsciDNSRequestAO::



Protected Methods

- `OsciDNSRequestAO` (const char *name)
- void `ConstructL` (OsciDNSI *aDNS, OsciDNSMethod *aMethod)
- void `Abort` ()
- void `NewRequest` ()
- void `RequestDone` ()
- int `GetSocketError` ()
- `OsciSocketServI` * `Serv` ()
- void `DoCancel` ()
- void `Run` ()
- virtual void `Success` ()
- virtual void `Failure` ()
- virtual void `Cancelled` ()

Protected Attributes

- `OsciDNSI` * `iDNSI`
- `OsciDNSMethod` * `iDNSMethod`
- int32 `iSocketError`
- `PVLogger` * `iLogger`

Friends

- class `OsciDNSI`
- class `OsciDNSMethod`
- class `OsciDNSRequest`
- class `GetHostByNameParam`

7.120.1 Detailed Description

This is the base class for all requests to the socket server.

7.120.2 Constructor & Destructor Documentation

7.120.2.1 `OsciDNSRequestAO::OsciDNSRequestAO (const char * name)` [inline, protected]

7.120.3 Member Function Documentation

7.120.3.1 `void OsciDNSRequestAO::Abort ()` [inline, protected]

7.120.3.2 `virtual void OsciDNSRequestAO::Cancelled ()` [inline, protected, virtual]

7.120.3.3 `void OsciDNSRequestAO::ConstructL (OsciDNSI * aDNS, OsciDNSMethod * aMethod)` [inline, protected]

7.120.3.4 `void OsciDNSRequestAO::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 [OsciActiveObject](#).

7.120.3.5 `virtual void OsciDNSRequestAO::Failure ()` [inline, protected, virtual]

7.120.3.6 `int OsciDNSRequestAO::GetSocketError ()` [protected]

7.120.3.7 `void OsciDNSRequestAO::NewRequest ()` [protected]

7.120.3.8 `void OsciDNSRequestAO::RequestDone ()` [protected]

7.120.3.9 `void OsciDNSRequestAO::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](#).

7.120.3.10 [OsciSocketServI*](#) `OsciDNSRequestAO::Serv ()` [protected]

7.120.3.11 `virtual void OsciDNSRequestAO::Success ()` [inline, protected, virtual]

7.120.4 Friends And Related Function Documentation

7.120.4.1 friend class `GetHostByNameParam` [friend]

7.120.4.2 friend class `OsciDNSI` [friend]

7.120.4.3 friend class `OsciDNSMethod` [friend]

7.120.4.4 friend class `OsciDNSRequest` [friend]

7.120.5 Field Documentation

7.120.5.1 [OsciDNSI*](#) `OsciDNSRequestAO::iDNSI` [protected]

7.120.5.2 [OsciDNSMethod*](#) `OsciDNSRequestAO::iDNSMethod` [protected]

7.120.5.3 [PVLogger*](#) `OsciDNSRequestAO::iLogger` [protected]

7.120.5.4 `int32 OsciDNSRequestAO::iSocketError` [protected]

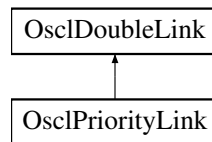
The documentation for this class was generated from the following file:

- [oscl_dns_method.h](#)

7.121 OsciDoubleLink Class Reference

```
#include <osci_double_list.h>
```

Inheritance diagram for OsciDoubleLink::



Public Methods

- [OsciDoubleLink \(\)](#)
- void [Remove \(\)](#)
- void [InsertAfter \(OsciDoubleLink *aLink\)](#)
- void [InsertBefore \(OsciDoubleLink *aLink\)](#)

Data Fields

- OsciDoubleLink * [iNext](#)
- OsciDoubleLink * [iPrev](#)

7.121.1 Constructor & Destructor Documentation

7.121.1.1 [OsciDoubleLink::OsciDoubleLink \(\) \[inline\]](#)

7.121.2 Member Function Documentation

7.121.2.1 [void OsciDoubleLink::InsertAfter \(OsciDoubleLink * *aLink*\)](#)

7.121.2.2 [void OsciDoubleLink::InsertBefore \(OsciDoubleLink * *aLink*\)](#)

7.121.2.3 [void OsciDoubleLink::Remove \(\)](#)

7.121.3 Field Documentation

7.121.3.1 [OsciDoubleLink* OsciDoubleLink::iNext](#)

7.121.3.2 [OsciDoubleLink* OsciDoubleLink::iPrev](#)

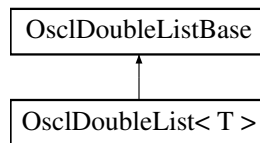
The documentation for this class was generated from the following file:

- [osci_double_list.h](#)

7.122 OsciDoubleList< T > Class Template Reference

```
#include <osci_double_list.h>
```

Inheritance diagram for OsciDoubleList< T >::



Public Methods

- OSCI_INLINE [OsciDoubleList](#) ()
- OSCI_INLINE [OsciDoubleList](#) (int32 anOffset)
- OSCI_INLINE void [InsertHead](#) (T &aRef)
- OSCI_INLINE void [InsertTail](#) (T &aRef)
- OSCI_INLINE bool [IsHead](#) (const T *aPtr) const
- OSCI_INLINE bool [IsTail](#) (const T *aPtr) const
- OSCI_INLINE T * [Head](#) () const
- OSCI_INLINE T * [Tail](#) () const

```
template<class T> class OsciDoubleList< T >
```

7.122.1 Constructor & Destructor Documentation

7.122.1.1 `template<class T> OSCI_INLINE OsciDoubleList< T >::OsciDoubleList ()`

7.122.1.2 `template<class T> OSCI_INLINE OsciDoubleList< T >::OsciDoubleList (int32 anOffset)`

7.122.2 Member Function Documentation

7.122.2.1 `template<class T> OSCI_INLINE T* OsciDoubleList< T >::Head ()`

7.122.2.2 `template<class T> OSCI_INLINE void OsciDoubleList< T >::InsertHead (T &aRef)`

7.122.2.3 `template<class T> OSCI_INLINE void OsciDoubleList< T >::InsertTail (T &aRef)`

7.122.2.4 `template<class T> OSCI_INLINE bool OsciDoubleList< T >::IsHead (const T * aPtr) const`

7.122.2.5 `template<class T> OSCI_INLINE bool OsciDoubleList< T >::IsTail (const T * aPtr) const`

7.122.2.6 `template<class T> OSCI_INLINE T* OsciDoubleList< T >::Tail ()`

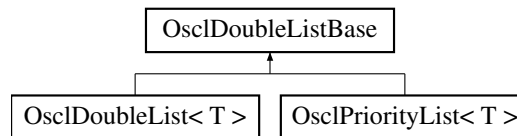
The documentation for this class was generated from the following file:

- [osci_double_list.h](#)

7.123 OsciDoubleListBase Class Reference

```
#include <osci_double_list.h>
```

Inheritance diagram for OsciDoubleListBase::



Public Methods

- bool [IsEmpty](#) () const
- void [SetOffset](#) (int32 anOffset)
- void [Reset](#) ()
- [OsciDoubleLink](#) * [getHead](#) ()
- int32 [getOffset](#) ()

Protected Methods

- [OsciDoubleListBase](#) ()
- [OsciDoubleListBase](#) (int32 anOffset)
- void [InsertHead](#) ([OsciAny](#) *aPtr)
- void [InsertTail](#) ([OsciAny](#) *aPtr)
- void [Insert](#) ([OsciAny](#) *aPtr)

Protected Attributes

- [OsciDoubleLink](#) iHead
- int32 iOffset

7.123.1 Constructor & Destructor Documentation

7.123.1.1 `OsciDoubleListBase::OsciDoubleListBase ()` [protected]

7.123.1.2 `OsciDoubleListBase::OsciDoubleListBase (int32 anOffset)` [protected]

7.123.2 Member Function Documentation

7.123.2.1 `OsciDoubleLink* OsciDoubleListBase::getHead ()` [inline]

7.123.2.2 `int32 OsciDoubleListBase::getOffset ()` [inline]

7.123.2.3 `void OsciDoubleListBase::Insert (OsciAny * aPtr)` [protected]

7.123.2.4 `void OsciDoubleListBase::InsertHead (OsciAny * aPtr)` [protected]

7.123.2.5 `void OsciDoubleListBase::InsertTail (OsciAny * aPtr)` [protected]

7.123.2.6 `bool OsciDoubleListBase::IsEmpty ()`

7.123.2.7 `void OsciDoubleListBase::Reset ()`

7.123.2.8 `void OsciDoubleListBase::SetOffset (int32 anOffset)`

7.123.3 Field Documentation

7.123.3.1 `OsciDoubleLink OsciDoubleListBase::iHead` [protected]

7.123.3.2 `int32 OsciDoubleListBase::iOffset` [protected]

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

- [osci_double_list.h](#)

7.124 OsciDoubleRunner< T > Class Template Reference

```
#include <osci_double_list.h>
```

Public Methods

- [OsciDoubleRunner](#) ([OsciDoubleListBase](#) &aQue)
- void [Set](#) (T &aLink)
- [operator T * \(\)](#)
- T * [operator++](#) (int)
- T * [operator--](#) (int)
- void [SetToHead](#) ()
- void [SetToTail](#) ()

Protected Attributes

- int32 [iOffset](#)
- [OsciDoubleLink](#) * [iHead](#)
- [OsciDoubleLink](#) * [iNext](#)

```
template<class T> class OsciDoubleRunner< T >
```

7.124.1 Constructor & Destructor Documentation

7.124.1.1 `template<class T> OsciDoubleRunner< T >::OsciDoubleRunner (OsciDoubleListBase &aQue) [inline]`

7.124.2 Member Function Documentation

7.124.2.1 `template<class T> OsciDoubleRunner< T >::operator T * () [inline]`

7.124.2.2 `template<class T> T* OsciDoubleRunner< T >::operator++ (int) [inline]`

7.124.2.3 `template<class T> T* OsciDoubleRunner< T >::operator-- (int)`

7.124.2.4 `template<class T> void OsciDoubleRunner< T >::Set (T &aLink) [inline]`

7.124.2.5 `template<class T> void OsciDoubleRunner< T >::SetToHead () [inline]`

7.124.2.6 `template<class T> void OsciDoubleRunner< T >::SetToTail () [inline]`

7.124.3 Field Documentation

7.124.3.1 `template<class T> OsciDoubleLink* OsciDoubleRunner< T >::iHead [protected]`

7.124.3.2 `template<class T> OsciDoubleLink* OsciDoubleRunner< T >::iNext [protected]`

7.124.3.3 `template<class T> int32 OsciDoubleRunner< T >::iOffset [protected]`

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

- [osci_double_list.h](#)

7.125 OslError Class Reference

```
#include <oscl_error.h>
```

Static Public Methods

- OSCL_IMPORT_REF void [PushL](#) ([_OslHeapBase](#) *aPtr)
- OSCL_IMPORT_REF void [PushL](#) ([OslAny](#) *aPtr)
- OSCL_IMPORT_REF void [PushL](#) ([OslTrapItem](#) 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](#) ([OslAny](#) *a)
- OSCL_IMPORT_REF void [LeaveIfError](#) (int32 aReason)

7.125.1 Detailed Description

User Error class

7.125.2 Member Function Documentation

7.125.2.1 OSCL_IMPORT_REF void OslError::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.

7.125.2.2 OSCL_IMPORT_REF void OslError::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.

7.125.2.3 OSCL_IMPORT_REF void OslError::LeaveIfNull (OslAny * a) [static]

Evaluate the input parameter, and if it is null, do a Leave with OslErrNoMemory reason code.

7.125.2.4 OSCL_IMPORT_REF void OslError::Pop (int32 aCount) [static]

Pop the cleanup stack N times

7.125.2.5 OSCL_IMPORT_REF void OslError::Pop () [static]

Pop the cleanup stack

7.125.2.6 OSCL_IMPORT_REF void OsclError::PopDealloc (int32 *aCount*) [static]

PopDealloc N times

7.125.2.7 OSCL_IMPORT_REF void OsclError::PopDealloc () [static]

Destroy the item on the top of the cleanup stack and pop it

7.125.2.8 OSCL_IMPORT_REF void OsclError::PushL (OsclTrapItem *anItem*) [static]

Push an [OsclTrapItem](#) onto the cleanup stack

7.125.2.9 OSCL_IMPORT_REF void OsclError::PushL (OsclAny * *aPtr*) [static]

Push an OsclAny item onto the cleanup stack.

7.125.2.10 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](#)

7.126 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

7.126.1 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

7.126.2 Constructor & Destructor Documentation

7.126.2.1 [OsclErrorAllocator::OsclErrorAllocator](#) ([Oscl_DefAlloc](#) * allocator) [inline]

constructor method

Parameters:

allocator - a pointer to the concrete object that provides the allocator/deallocator

7.126.3 Member Function Documentation

7.126.3.1 [OsclAny*](#) [OsclErrorAllocator::allocate](#) (uint32 aSize) [inline, static]

static method to allocate a block of memory on heap

Parameters:

aSize - number of bytes to allocate

7.126.3.2 [OsciAny](#) `OsciErrorAllocator::deallocate (OsciAny * aPointer)` [inline, static]

static method to deallocate a block of memory on heap

Parameters:

aPointer - pointer to block of memory to be deallocated

7.126.3.3 `void OsciErrorAllocator::operator delete (OsciAny * aPtr, OsciAny * aPtr2)`
[inline]

delete operator that doesn't do anything, user has to deallocate manually

7.126.3.4 `void* OsciErrorAllocator::operator new (uint32 size, OsciAny * aPtr)` [inline]

placement new operator that allocates memory using the user defined methods

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

- [osci_error_allocator.h](#)

7.127 OslErrorTrap Class Reference

```
#include <oscl_error.h>
```

Static Public Methods

- OSL_IMPORT_REF int32 [Init](#) ([Osl_DefAlloc](#) *aAlloc=NULL)
- OSL_IMPORT_REF int32 [Cleanup](#) ()
- OSL_IMPORT_REF [OslErrorTrapImp](#) * [GetErrorTrapImp](#) ()

7.127.1 Member Function Documentation

7.127.1.1 OSL_IMPORT_REF int32 OslErrorTrap::Cleanup () [static]

Cleanup and destroy error trap for the calling thread.

Returns:

0 for success, or an error

7.127.1.2 OSL_IMPORT_REF [OslErrorTrapImp](#)* OslErrorTrap::GetErrorTrapImp () [static]

Get the ErrorTrapImp for the current thread. Leaves on error.

7.127.1.3 OSL_IMPORT_REF int32 OslErrorTrap::Init ([Osl_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](#)

7.128 OslErrorTrapImp Class Reference

```
#include <oscl_error_trapcleanup.h>
```

Public Methods

- OSL_IMPORT_REF void [UnTrap](#) ()

Static Public Methods

- OSL_IMPORT_REF OslErrorTrapImp * [Trap](#) ()
- OSL_IMPORT_REF OslErrorTrapImp * [TrapNoTls](#) (OslErrorTrapImp *)

Data Fields

- OslJump * [iJumpData](#)
- int32 [iLeave](#)
- OslTrapStack * [iTrapStack](#)

Friends

- class [OslErrorTrap](#)
- class [OslError](#)
- class [OslExecScheduler](#)
- class [OslExecSchedulerCommonBase](#)
- class [OslJump](#)
- class [OslJumpMark](#)
- class [OslTrapStack](#)
- class [CPVInterfaceProxy](#)
- class [OslScheduler](#)

7.128.1 Detailed Description

A per-thread cleanup stack with nested trap support.

7.128.2 Member Function Documentation

7.128.2.1 OSL_IMPORT_REF OslErrorTrapImp* OslErrorTrapImp::Trap () [static]

PV trap cleanup. Public for use in macros only.

7.128.2.2 OSL_IMPORT_REF OslErrorTrapImp* OslErrorTrapImp::TrapNoTls (OslErrorTrapImp *) [static]

7.128.2.3 OSL_IMPORT_REF void OslErrorTrapImp::UnTrap ()

these are used in public macros, but aren't intended as public methods or members.

7.128.3 Friends And Related Function Documentation

7.128.3.1 friend class CPVInterfaceProxy [friend]

7.128.3.2 friend class OsciError [friend]

7.128.3.3 friend class OsciErrorTrap [friend]

7.128.3.4 friend class OsciExecScheduler [friend]

7.128.3.5 friend class OsciExecSchedulerCommonBase [friend]

7.128.3.6 friend class OsciJump [friend]

7.128.3.7 friend class OsciJumpMark [friend]

7.128.3.8 friend class OsciScheduler [friend]

7.128.3.9 friend class OsciTrapStack [friend]

7.128.4 Field Documentation

7.128.4.1 [OsciJump*](#) OsciErrorTrapImp::iJumpData

7.128.4.2 int32 OsciErrorTrapImp::iLeave

7.128.4.3 [OsciTrapStack*](#) OsciErrorTrapImp::iTrapStack

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

- [osci_error_trapcleanup.h](#)

7.129 OslException< 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

- [OslException \(\)](#)

Static Public Methods

- int [getLeaveCode \(\)](#)

7.129.1 Detailed Description

```
template<int LeaveCode> class OslException< 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 OslException 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

7.129.2 Constructor & Destructor Documentation

7.129.2.1 `template<int LeaveCode> OslException< LeaveCode >::OslException ()`
[inline]

7.129.3 Member Function Documentation

7.129.3.1 `template<int LeaveCode> int OslException< LeaveCode >::getLeaveCode ()`
[inline, static]

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

- [oscl_exception.h](#)

7.130 OsciExclusiveArrayPtr< T > Class Template Reference

The OsciExclusiveArrayPtr 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 OsciExclusiveArrayPtr expires, its destructor uses delete to free the memory.

```
#include <osci_exclusive_ptr.h>
```

Public Methods

- [OsciExclusiveArrayPtr](#) (T *inPtr=0)
Default constructor Initializes the pointer and takes ownership.
- [OsciExclusiveArrayPtr](#) (OsciExclusiveArrayPtr< T > &_Y)
Copy constructor.
- OsciExclusiveArrayPtr< T > & [operator=](#) (OsciExclusiveArrayPtr< T > &_Y)
Assignment operator from an another OsciExclusiveArrayPtr.
- virtual [~OsciExclusiveArrayPtr](#) ()
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](#)

7.130.1 Detailed Description

```
template<class T> class OsciExclusiveArrayPtr< T >
```

The OsciExclusiveArrayPtr 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 OsciExclusiveArrayPtr 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 [OsciExclusivePtr](#) 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 [OsciExclusivePtr](#) 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

7.130.2 Constructor & Destructor Documentation

7.130.2.1 `template<class T> OsciExclusiveArrayPtr< T >::OsciExclusiveArrayPtr (T * inPtr = 0) [inline, explicit]`

Default constructor Initializes the pointer and takes ownership.

7.130.2.2 `template<class T> OsciExclusiveArrayPtr< T >::OsciExclusiveArrayPtr (OsciExclusiveArrayPtr< T > & _Y) [inline]`

Copy constructor.

Initializes the pointer and takes ownership from another OsciExclusiveArrayPtr. Note that the other class does NOT own the pointer any longer, and hence it is NOT its responsibility to free it.

7.130.2.3 `template<class T> virtual OsciExclusiveArrayPtr< T >::~~OsciExclusiveArrayPtr () [inline, virtual]`

Destructor.

The pointer is deleted in case this class still has ownership

7.130.3 Member Function Documentation

7.130.3.1 `template<class T> T* OsciExclusiveArrayPtr< T >::get () const [inline]`

[get\(\)](#) method returns the pointer, currently owned by the class.

7.130.3.2 `template<class T> T& OsciExclusiveArrayPtr< T >::operator * () const [inline]`

The indirection operator (*) accesses a value indirectly, through a pointer.

This operator ensures that the OsciExclusiveArrayPtr can be used like the regular pointer that it was initialized with.

7.130.3.3 `template<class T> T* OsciExclusiveArrayPtr< T >::operator -> () const [inline]`

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the OsciExclusiveArrayPtr can be used like the regular pointer that it was initialized with.

7.130.3.4 `template<class T> OsciExclusiveArrayPtr<T>& OsciExclusiveArrayPtr< T >::operator= (OsciExclusiveArrayPtr< T > &_Y) [inline]`

Assignment operator from an another OsciExclusiveArrayPtr.

Parameters:

`_Y` The value parameter should be another OsciExclusiveArrayPtr

Returns:

Returns a reference to this OsciExclusiveArrayPtr 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 OsciExclusiveArrayPtr given as the input parameter. The ownership of the pointer is transferred.

7.130.3.5 `template<class T> T* OsciExclusiveArrayPtr< T >::release () [inline]`

`release()` method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.

7.130.3.6 `template<class T> bool OsciExclusiveArrayPtr< 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.

7.130.4 Field Documentation

7.130.4.1 `template<class T> T* OsciExclusiveArrayPtr< T >::_Ptr [protected]`

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

- [osci_exclusive_ptr.h](#)

7.131 OsciExclusivePtr< T > Class Template Reference

The OsciExclusivePtr 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 OsciExclusivePtr expires, its destructor uses delete to free the memory.

```
#include <osci_exclusive_ptr.h>
```

Public Methods

- [OsciExclusivePtr](#) (T *inPtr=0)
Default constructor Initializes the pointer and takes ownership.
- [OsciExclusivePtr](#) (OsciExclusivePtr< T > &_Y)
Copy constructor.
- OsciExclusivePtr< T > & [operator=](#) (OsciExclusivePtr< T > &_Y)
Assignment operator from an another OsciExclusivePtr.
- virtual [~OsciExclusivePtr](#) ()
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](#)

7.131.1 Detailed Description

```
template<class T> class OsciExclusivePtr< T >
```

The OsciExclusivePtr 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 OsciExclusivePtr 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 OsciExclusivePtr 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 OsciExclusivePtr 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

7.131.2 Constructor & Destructor Documentation

7.131.2.1 `template<class T> OsciExclusivePtr< T >::OsciExclusivePtr (T * inPtr = 0)`
`[inline, explicit]`

Default constructor Initializes the pointer and takes ownership.

7.131.2.2 `template<class T> OsciExclusivePtr< T >::OsciExclusivePtr (OsciExclusivePtr< T > & _Y)` `[inline]`

Copy constructor.

Initializes the pointer and takes ownership from another OsciExclusivePtr. Note that the other class does NOT own the pointer any longer, and hence it is NOT its responsibility to free it.

7.131.2.3 `template<class T> virtual OsciExclusivePtr< T >::~~OsciExclusivePtr ()` `[inline, virtual]`

Destructor.

The pointer is deleted in case this class still has ownership

7.131.3 Member Function Documentation

7.131.3.1 `template<class T> T* OsciExclusivePtr< T >::get () const` `[inline]`

[get\(\)](#) method returns the pointer, currently owned by the class.

7.131.3.2 `template<class T> T& OsciExclusivePtr< T >::operator * () const` `[inline]`

The indirection operator (*) accesses a value indirectly, through a pointer.

This operator ensures that the OsciExclusivePtr can be used like the regular pointer that it was initialized with.

7.131.3.3 `template<class T> T* OsciExclusivePtr< T >::operator -> () const` `[inline]`

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the OsciExclusivePtr can be used like the regular pointer that it was initialized with.

7.131.3.4 `template<class T> OsciExclusivePtr<T>& OsciExclusivePtr< T >::operator=(OsciExclusivePtr< T > &_Y) [inline]`

Assignment operator from an another OsciExclusivePtr.

Parameters:

`_Y` The value parameter should be another OsciExclusivePtr

Returns:

Returns a reference to this OsciExclusivePtr 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 OsciExclusivePtr given as the input parameter. The ownership of the pointer is transferred.

7.131.3.5 `template<class T> T* OsciExclusivePtr< T >::release () [inline]`

`release()` method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.

7.131.3.6 `template<class T> bool OsciExclusivePtr< 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.

7.131.4 Field Documentation**7.131.4.1** `template<class T> T* OsciExclusivePtr< T >::_Ptr [protected]`

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

- [oscl_exclusive_ptr.h](#)

7.132 OsciExclusivePtrA< T, Alloc > Class Template Reference

The OsciExclusivePtrA 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 OsciExclusivePtrA expires, Alloc is used to free the memory.

```
#include <osci_exclusive_ptr.h>
```

Public Methods

- [OsciExclusivePtrA](#) (T *inPtr=0)
Default constructor Initializes the pointer and takes ownership.
- [OsciExclusivePtrA](#) (OsciExclusivePtrA< T, Alloc > &_Y)
Copy constructor.
- OsciExclusivePtrA< T, Alloc > & [operator=](#) (OsciExclusivePtrA< T, Alloc > &_Y)
Assignment operator from an another [OsciExclusiveArrayPtr](#).
- virtual [~OsciExclusivePtrA](#) ()
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](#)

7.132.1 Detailed Description

```
template<class T, class Alloc> class OsciExclusivePtrA< T, Alloc >
```

The OsciExclusivePtrA 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 OsciExclusivePtrA 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 [OsciExclusivePtr](#) 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 [OsciExclusivePtr](#) 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

7.132.2 Constructor & Destructor Documentation

7.132.2.1 `template<class T, class Alloc> OsciExclusivePtrA< T, Alloc >::OsciExclusivePtrA (T * inPtr = 0) [inline, explicit]`

Default constructor Initializes the pointer and takes ownership.

7.132.2.2 `template<class T, class Alloc> OsciExclusivePtrA< T, Alloc >::OsciExclusivePtrA (OsciExclusivePtrA< T, Alloc > & _Y) [inline]`

Copy constructor.

Initializes the pointer and takes ownership from another [OsciExclusiveArrayPtr](#). Note that the other class does NOT own the pointer any longer, and hence it is NOT its responsibility to free it.

7.132.2.3 `template<class T, class Alloc> virtual OsciExclusivePtrA< T, Alloc >::~~OsciExclusivePtrA () [inline, virtual]`

Destructor.

The pointer is deleted in case this class still has ownership

7.132.3 Member Function Documentation

7.132.3.1 `template<class T, class Alloc> T* OsciExclusivePtrA< T, Alloc >::get () const [inline]`

[get\(\)](#) method returns the pointer, currently owned by the class.

7.132.3.2 `template<class T, class Alloc> T& OsciExclusivePtrA< T, Alloc >::operator * () const [inline]`

The indirection operator (*) accesses a value indirectly, through a pointer.

This operator ensures that the [OsciExclusiveArrayPtr](#) can be used like the regular pointer that it was initialized with.

7.132.3.3 `template<class T, class Alloc> T* OsciExclusivePtrA< T, Alloc >::operator -> () const [inline]`

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the [OsciExclusiveArrayPtr](#) can be used like the regular pointer that it was initialized with.

7.132.3.4 `template<class T, class Alloc> OsciExclusivePtrA<T, Alloc>& OsciExclusivePtrA< T, Alloc >::operator= (OsciExclusivePtrA< T, Alloc > & _Y) [inline]`

Assignment operator from an another [OsciExclusiveArrayPtr](#).

Parameters:

`_Y` The value parameter should be another [OsciExclusiveArrayPtr](#)

Returns:

Returns a reference to this [OsciExclusiveArrayPtr](#) 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 [OsciExclusiveArrayPtr](#) given as the input parameter. The ownership of the pointer is transferred.

7.132.3.5 `template<class T, class Alloc> T* OsciExclusivePtrA< T, Alloc >::release () [inline]`

[release\(\)](#) method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.

7.132.3.6 `template<class T, class Alloc> bool OsciExclusivePtrA< 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.

7.132.4 Field Documentation

7.132.4.1 `template<class T, class Alloc> T* OsciExclusivePtrA< T, Alloc >::_Ptr [protected]`

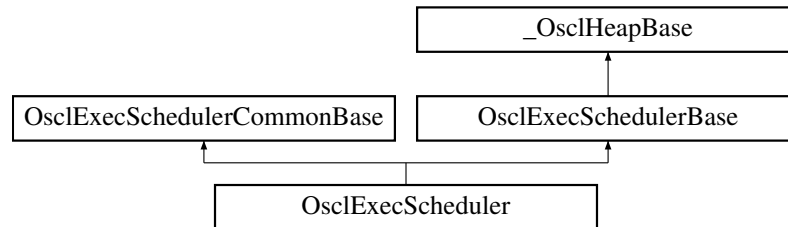
The documentation for this class was generated from the following file:

- [osci_exclusive_ptr.h](#)

7.133 OsciExecScheduler Class Reference

```
#include <osci_scheduler.h>
```

Inheritance diagram for OsciExecScheduler::



Public Methods

- OSL_IMPORT_REF void [RunSchedulerNonBlocking](#) (int32 aTargetCount, int32 &aReady, uint32 &aDelayMsec)
- OSL_IMPORT_REF void [RegisterForCallback](#) ([OsciSchedulerObserver](#) *aCallback, [OsciAny](#) *aCallbackContext)

Static Public Methods

- OSL_IMPORT_REF [OsciExecScheduler](#) * [Current](#) ()

Friends

- class [OsciScheduler](#)

7.133.1 Member Function Documentation

7.133.1.1 OSL_IMPORT_REF [OsciExecScheduler](#)* [OsciExecScheduler::Current](#) () [static]

Get currently installed scheduler for calling thread, or NULL if no scheduler is installed.

7.133.1.2 OSL_IMPORT_REF void [OsciExecScheduler::RegisterForCallback](#) ([OsciSchedulerObserver](#) * aCallback, [OsciAny](#) * 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.

7.133.1.3 OSL_IMPORT_REF void [OsciExecScheduler::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.

7.133.2 Friends And Related Function Documentation

7.133.2.1 friend class OslScheduler [friend]

Reimplemented from [OslExecSchedulerCommonBase](#).

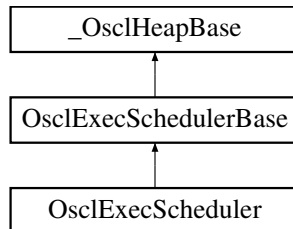
The documentation for this class was generated from the following file:

- [oscl_scheduler.h](#)

7.134 OsciExecSchedulerBase Class Reference

```
#include <osci_scheduler_types.h>
```

Inheritance diagram for OsciExecSchedulerBase::



Friends

- class [OsciExecScheduler](#)
- class [OsciCoeActiveScheduler](#)
- class [PVActiveBase](#)

7.134.1 Detailed Description

OsciActiveSchedulerBase is the base for [OsciExecScheduler](#). The non-Symbian OsciActiveSchedulerBase class is functionally similar to a subset of Symbian CActiveScheduler class.

7.134.2 Friends And Related Function Documentation

7.134.2.1 friend class OsciCoeActiveScheduler [friend]

7.134.2.2 friend class OsciExecScheduler [friend]

7.134.2.3 friend class PVActiveBase [friend]

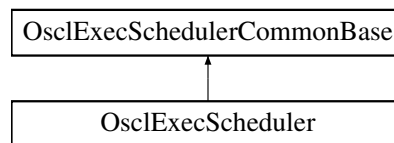
The documentation for this class was generated from the following file:

- [osci_scheduler_types.h](#)

7.135 OsciExecSchedulerCommonBase Class Reference

```
#include <osci_scheduler.h>
```

Inheritance diagram for OsciExecSchedulerCommonBase::



Public Methods

- OSCL_IMPORT_REF void [StartScheduler](#) (OsciSemaphore *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 [OsciNameString](#)< PVSCHEDNAMELEN > * [GetName](#) ()
- OSCL_IMPORT_REF uint32 [GetId](#) ()

Protected Types

- enum [TOtherExecStats](#) { [EOtherExecStats_WaitTime](#), [EOtherExecStats_QueueTime](#), [EOtherExecStats_NativeOS](#), [EOtherExecStats_ReleaseTime](#), [EOtherExecStats_Last](#) }

Protected Methods

- virtual [~OsciExecSchedulerCommonBase](#) ()
- void [InstallScheduler](#) ()
- void [UninstallScheduler](#) ()
- void [Error](#) (int32 anError) const
- [OsciExecSchedulerCommonBase](#) (Osci_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](#) (PActiveBase *)
- void [ConstructStatQ](#) ()
- void [BeginStats](#) ()
- void [EndStats](#) ()
- void [CleanupStatQ](#) ()
- PActiveBase * [FindPVBase](#) (PActiveBase *active, OsciDoubleList< PActiveBase > &)
- void [CleanupExecQ](#) ()
- void [InitExecQ](#) (int)
- void [ResetLogPerf](#) ()
- void [IncLogPerf](#) (uint32)

Static Protected Methods

- OsciExecSchedulerCommonBase * [GetScheduler](#) ()
- OsciExecSchedulerCommonBase * [SetScheduler](#) (OsciExecSchedulerCommonBase *)
- void [ShowStats](#) (PActiveStats *active)
- void [ShowSummaryStats](#) (PActiveStats *active, PVLogger *, int64, int64 &, float &)

Protected Attributes

- bool [iBlockingMode](#)
- bool [iNativeMode](#)
- PVSchedulerStopper * [iStopper](#)
- OsciNoYieldMutex [iStopperCrit](#)
- PVThreadContext [iThreadContext](#)
- OsciNameString< PVSCHEDNAMELEN > [iName](#)
- bool [iDoStop](#)
- bool [iDoSuspend](#)
- bool [iSuspended](#)
- OsciSemaphore [iResumeSem](#)
- OsciErrorTrapImp * [iErrorTrapImp](#)
- OsciReadyQ [iReadyQ](#)
- OsciTimerQ [iExecTimerQ](#)
- uint32 [iNumAOAdded](#)
- OsciDoubleList< PActiveStats > [iPVStatQ](#)
- PActiveStats * [iOtherExecStats](#) [EOtherExecStats_Last]
- uint8 * [iTotalTicksTemp](#)
- int64 [iGrandTotalTicks](#)
- float [iTotalPercent](#)
- uint32 [iTime](#)
- int32 [iDelta](#)
- PActiveStats * [iPVStats](#)
- PVLogger * [iLogger](#)
- PVLogger * [iDebugLogger](#)
- char * [iLogPerfIndentStr](#)
- int32 [iLogPerfIndentStrLen](#)
- uint32 [iLogPerfTotal](#)
- Osci_DefAlloc * [iAlloc](#)
- OsciMemAllocator [iDefAlloc](#)

Static Protected Attributes

- const uint32 [iTimeCompareThreshold](#)

Friends

- class [OsciScheduler](#)
- class [PVThreadContext](#)
- class [OsciCoeActiveScheduler](#)
- class [OsciTimerCompare](#)
- class [OsciReadyQ](#)
- class [OsciError](#)
- class [PVActiveStats](#)
- class [OsciActiveObject](#)
- class [OsciTimerObject](#)
- class [PVActiveBase](#)
- class [PVSchedulerStopper](#)
- class [OsciExecScheduler](#)

7.135.1 Member Enumeration Documentation

7.135.1.1 enum [OsciExecSchedulerCommonBase::TOtherExecStats](#) [protected]

Enumeration values:

[EOtherExecStats_WaitTime](#)

[EOtherExecStats_QueueTime](#)

[EOtherExecStats_NativeOS](#)

[EOtherExecStats_ReleaseTime](#)

[EOtherExecStats_Last](#)

7.135.2 Constructor & Destructor Documentation

- 7.135.2.1 `virtual OsciExecSchedulerCommonBase::~OsciExecSchedulerCommonBase ()`
[protected, virtual]
- 7.135.2.2 `OsciExecSchedulerCommonBase::OsciExecSchedulerCommonBase (Osci_DefAlloc *)`
[protected]

7.135.3 Member Function Documentation

- 7.135.3.1 `void OsciExecSchedulerCommonBase::AddToExecTimerQ (PVAActiveBase * active, uint32)` [protected]
- 7.135.3.2 `void OsciExecSchedulerCommonBase::BeginScheduling (bool blocking, bool native)`
[protected]
- 7.135.3.3 `void OsciExecSchedulerCommonBase::BeginStats ()` [protected]
- 7.135.3.4 `void OsciExecSchedulerCommonBase::BlockingLoopL ()` [protected]
- 7.135.3.5 `void OsciExecSchedulerCommonBase::CallRunExec (PVAActiveBase *)` [protected]
- 7.135.3.6 `void OsciExecSchedulerCommonBase::CleanupExecQ ()` [protected]
- 7.135.3.7 `void OsciExecSchedulerCommonBase::CleanupStatQ ()` [protected]
- 7.135.3.8 `virtual void OsciExecSchedulerCommonBase::ConstructL (const char * name, int)`
[protected, virtual]
- 7.135.3.9 `void OsciExecSchedulerCommonBase::ConstructStatQ ()` [protected]
- 7.135.3.10 `void OsciExecSchedulerCommonBase::EndScheduling ()` [protected]
- 7.135.3.11 `void OsciExecSchedulerCommonBase::EndStats ()` [protected]
- 7.135.3.12 `void OsciExecSchedulerCommonBase::Error (int32 anError) const` [protected]
- 7.135.3.13 `PVAActiveBase* OsciExecSchedulerCommonBase::FindPVBase (PVAActiveBase * active, OsciDoubleList< PVAActiveBase > &)` [protected]
- 7.135.3.14 `OSCL_IMPORT_REF uint32 OsciExecSchedulerCommonBase::GetId ()` [static]

Get numeric ID of current thread.

- 7.135.3.15 `OSCL_IMPORT_REF OsciNameString<PVSCHEDNAMELEN>*`
`OsciExecSchedulerCommonBase::GetName ()` [static]

Get name of scheduler for current thread.

- 7.135.3.16 **OsciExecSchedulerCommonBase* OsciExecSchedulerCommonBase::GetScheduler ()**
[static, protected]
- 7.135.3.17 **void OsciExecSchedulerCommonBase::IncLogPerf (uint32)** [protected]
- 7.135.3.18 **void OsciExecSchedulerCommonBase::InitExecQ (int)** [protected]
- 7.135.3.19 **void OsciExecSchedulerCommonBase::InstallScheduler ()** [protected]
- 7.135.3.20 **bool OsciExecSchedulerCommonBase::IsInstalled ()** [inline, protected]
- 7.135.3.21 **bool OsciExecSchedulerCommonBase::IsStarted ()** [protected]
- 7.135.3.22 **void OsciExecSchedulerCommonBase::PendComplete (PVAActiveBase *, int32 aReason, TPVThreadContext aContext)** [protected]
- 7.135.3.23 **void OsciExecSchedulerCommonBase::RequestCanceled (PVAActiveBase *)**
[protected]
- 7.135.3.24 **void OsciExecSchedulerCommonBase::ResetLogPerf ()** [protected]
- 7.135.3.25 **OSCL_IMPORT_REF void OsciExecSchedulerCommonBase::ResumeScheduler ()**

Resume scheduling immediately. This API only applies to a blocking loop scheduler.

- 7.135.3.26 **OsciExecSchedulerCommonBase* OsciExecSchedulerCommonBase::SetScheduler (OsciExecSchedulerCommonBase *)** [static, protected]
- 7.135.3.27 **void OsciExecSchedulerCommonBase::ShowStats (PVAActiveStats * active)** [static, protected]
- 7.135.3.28 **void OsciExecSchedulerCommonBase::ShowSummaryStats (PVAActiveStats * active, PVLogger *, int64, int64 &, float &)** [static, protected]
- 7.135.3.29 **OSCL_IMPORT_REF void OsciExecSchedulerCommonBase::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.

- 7.135.3.30 **OSCL_IMPORT_REF void OsciExecSchedulerCommonBase::StartScheduler (OsciSemaphore * 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.

7.135.3.31 OSCL_IMPORT_REF void OsciExecSchedulerCommonBase::StopScheduler ()

Stop scheduling. This API may be called from the scheduling thread or some other thread.

7.135.3.32 OSCL_IMPORT_REF void OsciExecSchedulerCommonBase::SuspendScheduler ()

Suspend scheduling when the current Run is complete. This API only applies to a blocking loop scheduler.

7.135.3.33 void OsciExecSchedulerCommonBase::UninstallScheduler () [protected]

7.135.3.34 PVAActiveBase* OsciExecSchedulerCommonBase::UpdateTimers (uint32 & aDelay) [protected]

7.135.3.35 PVAActiveBase* OsciExecSchedulerCommonBase::UpdateTimersMsec (uint32 & aDelay) [protected]

7.135.3.36 PVAActiveBase* OsciExecSchedulerCommonBase::WaitForReadyAO () [protected]

7.135.4 Friends And Related Function Documentation

7.135.4.1 friend class OsciActiveObject [friend]

7.135.4.2 friend class OsciCoeActiveScheduler [friend]

7.135.4.3 friend class OsciError [friend]

7.135.4.4 friend class OsciExecScheduler [friend]

7.135.4.5 friend class OsciReadyQ [friend]

7.135.4.6 friend class OsciScheduler [friend]

Reimplemented in [OsciExecScheduler](#).

7.135.4.7 friend class OsciTimerCompare [friend]

7.135.4.8 friend class OsciTimerObject [friend]

7.135.4.9 friend class PVActiveBase [friend]

7.135.4.10 friend class PVActiveStats [friend]

7.135.4.11 friend class PVSchedulerStopper [friend]

7.135.4.12 friend class PVThreadContext [friend]

7.135.5 Field Documentation

7.135.5.1 **Osci_DefAlloc*** OsciExecSchedulerCommonBase::iAlloc [protected]

7.135.5.2 bool OsciExecSchedulerCommonBase::iBlockingMode [protected]

7.135.5.3 **PVLogger*** OsciExecSchedulerCommonBase::iDebugLogger [protected]

7.135.5.4 **OsciMemAllocator** OsciExecSchedulerCommonBase::iDefAlloc [protected]

7.135.5.5 int32 OsciExecSchedulerCommonBase::iDelta [protected]

7.135.5.6 bool OsciExecSchedulerCommonBase::iDoStop [protected]

7.135.5.7 bool OsciExecSchedulerCommonBase::iDoSuspend [protected]

7.135.5.8 **OsciErrorTrapImp*** OsciExecSchedulerCommonBase::iErrorTrapImp
[protected]

7.135.5.9 **OsciTimerQ** OsciExecSchedulerCommonBase::iExecTimerQ [protected]

7.135.5.10 **int64** OsciExecSchedulerCommonBase::iGrandTotalTicks [protected]

7.135.5.11 **PVLogger*** OsciExecSchedulerCommonBase::iLogger [protected]

7.135.5.12 char* OsciExecSchedulerCommonBase::iLogPerfIndentStr [protected]

7.135.5.13 int32 OsciExecSchedulerCommonBase::iLogPerfIndentStrLen [protected]

7.135.5.14 uint32 OsciExecSchedulerCommonBase::iLogPerfTotal [protected]

7.135.5.15 **OsciNameString**<PVSCHEDNAMELEN> OsciExecSchedulerCommonBase::iName
[protected]

7.135.5.16 bool OsciExecSchedulerCommonBase::iNativeMode [protected]

7.135.5.17 uint32 OsciExecSchedulerCommonBase::iNumAOAdded [protected]

7.135.5.18 **PVActiveStats*** OsciExecSchedulerCommonBase::iOtherExecStats[EOtherExecStats_-
Last] [protected]

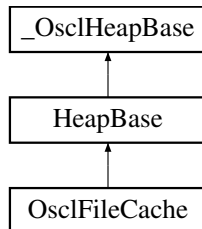
7.135.5.19 **OsciDoubleList**<**PVActiveStats**> OsciExecSchedulerCommonBase::iPVStatQ
[protected]

- [osci_scheduler.h](#)

7.136 OsciFileCache Class Reference

```
#include <osci_file_cache.h>
```

Inheritance diagram for OsciFileCache::



Public Methods

- [OsciFileCache](#) ([Osci_File](#) &aContainer)
- [~OsciFileCache](#) ()
- [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)
- [TOsciFileOffset FileSize](#) ()
- [int32 Seek](#) ([TOsciFileOffset](#) offset, [Osci_File::seek_type](#) origin)
- [TOsciFileOffset Tell](#) ()
- [int32 Flush](#) ()
- [int32 EndOfFile](#) ()
- [OSCL_IMPORT_REF OsciFileCacheBuffer * AddFixedCache](#) (const [Osci_File::OsciFixedCacheParam](#) &)

Data Fields

- [OsciFileCacheBuffer _movableCache](#)
- [Osci_Vector< OsciFileCacheBuffer, OsciMemAllocator > _fixedCaches](#)

Friends

- class [OsciFileCacheBuffer](#)

7.136.1 Constructor & Destructor Documentation

7.136.1.1 `OslFileCache::OslFileCache (Osl_File & aContainer)`

7.136.1.2 `OslFileCache::~~OslFileCache ()`

7.136.2 Member Function Documentation

7.136.2.1 `OSCL_IMPORT_REF OslFileCacheBuffer* OslFileCache::AddFixedCache (const Osl_File::OslFixedCacheParam &)`

7.136.2.2 `void OslFileCache::Close ()`

7.136.2.3 `int32 OslFileCache::EndOfFile () [inline]`

7.136.2.4 `TOslFileOffset OslFileCache::FileSize () [inline]`

7.136.2.5 `int32 OslFileCache::Flush ()`

7.136.2.6 `int32 OslFileCache::Open (uint32 mode, uint32 cache_size)`

7.136.2.7 `uint32 OslFileCache::Read (void * outputBuffer, uint32 size, uint32 numelements)`

7.136.2.8 `int32 OslFileCache::Seek (TOslFileOffset offset, Osl_File::seek_type origin)`

7.136.2.9 `TOslFileOffset OslFileCache::Tell () [inline]`

7.136.2.10 `uint32 OslFileCache::Write (const void * inputBuffer, uint32 size, uint32 numelements)`

7.136.3 Friends And Related Function Documentation

7.136.3.1 `friend class OslFileCacheBuffer [friend]`

7.136.4 Field Documentation

7.136.4.1 `Osl_Vector<OslFileCacheBuffer, OslMemAllocator> OslFileCache::_fixedCaches`

7.136.4.2 `OslFileCacheBuffer OslFileCache::_movableCache`

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

- [oscl_file_cache.h](#)

7.137 OsciFileCacheBuffer Class Reference

```
#include <osci_file_cache.h>
```

Public Methods

- [OsciFileCacheBuffer \(\)](#)
- [int32 SetPosition \(TOsciFileOffset pos\)](#)
- [int32 PrepRead \(\)](#)
- [int32 PrepWrite \(\)](#)
- [int32 WriteUpdatesToFile \(\)](#)
- [int32 FillFromFile \(uint32, uint32\)](#)
- [bool IsUpdated \(\)](#)
- [bool Contains \(TOsciFileOffset pos\)](#)
- [bool Precedes \(TOsciFileOffset pos\)](#)

Data Fields

- [OsciFileCache * iContainer](#)
- [bool isFixed](#)
- [uint32 capacity](#)
- [uint32 usableSize](#)
- [uint8 * pBuffer](#)
- [TOsciFileOffset filePosition](#)
- [uint32 currentPos](#)
- [uint32 endPos](#)
- [uint32 updateStart](#)
- [uint32 updateEnd](#)

7.137.1 Constructor & Destructor Documentation

7.137.1.1 `OsciFileCacheBuffer::OsciFileCacheBuffer ()` [inline]

7.137.2 Member Function Documentation

7.137.2.1 `bool OsciFileCacheBuffer::Contains (TOsciFileOffset pos)` [inline]

7.137.2.2 `int32 OsciFileCacheBuffer::FillFromFile (uint32, uint32)`

7.137.2.3 `bool OsciFileCacheBuffer::IsUpdated ()` [inline]

7.137.2.4 `bool OsciFileCacheBuffer::Preceeds (TOsciFileOffset pos)` [inline]

7.137.2.5 `int32 OsciFileCacheBuffer::PrepRead ()`

7.137.2.6 `int32 OsciFileCacheBuffer::PrepWrite ()`

7.137.2.7 `int32 OsciFileCacheBuffer::SetPosition (TOsciFileOffset pos)`

7.137.2.8 `int32 OsciFileCacheBuffer::WriteUpdatesToFile ()`

7.137.3 Field Documentation

7.137.3.1 `uint32 OsciFileCacheBuffer::capacity`

7.137.3.2 `uint32 OsciFileCacheBuffer::currentPos`

7.137.3.3 `uint32 OsciFileCacheBuffer::endPos`

7.137.3.4 `TOsciFileOffset OsciFileCacheBuffer::filePosition`

7.137.3.5 `OsciFileCache* OsciFileCacheBuffer::iContainer`

7.137.3.6 `bool OsciFileCacheBuffer::isFixed`

7.137.3.7 `uint8* OsciFileCacheBuffer::pBuffer`

7.137.3.8 `uint32 OsciFileCacheBuffer::updateEnd`

7.137.3.9 `uint32 OsciFileCacheBuffer::updateStart`

7.137.3.10 `uint32 OsciFileCacheBuffer::usableSize`

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

- [osci_file_cache.h](#)

7.138 OsciFileHandle Class Reference

```
#include <osci_file_handle.h>
```

Public Methods

- [OsciFileHandle](#) ([TOsciFileHandle](#) aHandle)
- [OsciFileHandle](#) (const [OsciFileHandle](#) &aHandle)
- [TOsciFileHandle Handle](#) () const

Friends

- class [Osci_File](#)

7.138.1 Detailed Description

OsciFileHandle is a container for a handle to a previously-opened file.

7.138.2 Constructor & Destructor Documentation

7.138.2.1 [OsciFileHandle::OsciFileHandle](#) ([TOsciFileHandle](#) aHandle) [inline]

7.138.2.2 [OsciFileHandle::OsciFileHandle](#) (const [OsciFileHandle](#) &aHandle) [inline]

7.138.3 Member Function Documentation

7.138.3.1 [TOsciFileHandle](#) [OsciFileHandle::Handle](#) () const [inline]

7.138.4 Friends And Related Function Documentation

7.138.4.1 friend class [Osci_File](#) [friend]

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

- [osci_file_handle.h](#)

7.139 OslFileManager Class Reference

```
#include <oscl_file_manager.h>
```

Public Types

- enum `OSCL_FILE_ATTRIBUTE_TYPE` { `OSCL_FILE_ATTRIBUTE_READONLY` = 0x00000001, `OSCL_FILE_ATTRIBUTE_HIDDEN` = 0x00000002, `OSCL_FILE_ATTRIBUTE_SYSTEM` = 0x00000004, `OSCL_FILE_ATTRIBUTE_DIRECTORY` = 0x00000010, `OSCL_FILE_ATTRIBUTE_ARCHIVE` = 0x00000020, `OSCL_FILE_ATTRIBUTE_NORMAL` = 0x00000080 }

Static Public Methods

- `OSCL_IMPORT_REF` bool `OslGetFileSize` (const `oscl_wchar` *aFileName, `uint64` &aFileSize)
- `OSCL_IMPORT_REF` bool `OslGetFileSize` (const char *aFileName, `uint64` &aFileSize)
- `OSCL_IMPORT_REF` bool `OslGetFileCreationTime` (const `oscl_wchar` *aFileName, `uint64` &aFileCreationTime)
- `OSCL_IMPORT_REF` bool `OslGetFileCreationTime` (const char *aFileName, `uint64` &aFileCreationTime)
- `OSCL_IMPORT_REF` bool `OslGetFileLastAccessTime` (const `oscl_wchar` *aFileName, `uint64` &aFileLastAccessTime)
- `OSCL_IMPORT_REF` bool `OslGetFileLastAccessTime` (const char *aFileName, `uint64` &aFileLastAccessTime)
- `OSCL_IMPORT_REF` bool `OslGetFileLastWriteTime` (const `oscl_wchar` *aFileName, `uint64` &aFileLastWriteTime)
- `OSCL_IMPORT_REF` bool `OslGetFileLastWriteTime` (const char *aFileName, `uint64` &aFileLastWriteTime)
- `OSCL_IMPORT_REF` bool `OslGetFileAttributes` (const `oscl_wchar` *aFileName, `uint32` &aFileAttributes)
- `OSCL_IMPORT_REF` bool `OslGetFileAttributes` (const char *aFileName, `uint32` &aFileAttributes)
- `OSCL_IMPORT_REF` void `OslExtractFilenameFromFullpath` (const char *aPath, char *&aFileName)
- `OSCL_IMPORT_REF` void `OslExtractFilenameFromFullpath` (const `oscl_wchar` *aPath, `oscl_wchar` *&aFileName)

7.139.1 Member Enumeration Documentation

7.139.1.1 enum OslFileManager::OSCL_FILE_ATTRIBUTE_TYPE

Enumeration values:

```
OSCL_FILE_ATTRIBUTE_READONLY
OSCL_FILE_ATTRIBUTE_HIDDEN
OSCL_FILE_ATTRIBUTE_SYSTEM
OSCL_FILE_ATTRIBUTE_DIRECTORY
OSCL_FILE_ATTRIBUTE_ARCHIVE
OSCL_FILE_ATTRIBUTE_NORMAL
```

7.139.2 Member Function Documentation

7.139.2.1 `OSCL_IMPORT_REF void OsciFileManager::OsciExtractFilenameFromFullpath (const oscl_wchar * aPath, oscl_wchar *& aFileName) [static]`

7.139.2.2 `OSCL_IMPORT_REF void OsciFileManager::OsciExtractFilenameFromFullpath (const char * aPath, char *& aFileName) [static]`

OsciExtractFilenameFromFullpath utility function provide the FileName From Path of a file.

Parameters:

in] character path; the full path of the file or directory

out] character FileName :file Name .It is assigned a pointer to file name in path itself.

Returns:

void for all condition

7.139.2.3 `OSCL_IMPORT_REF bool OsciFileManager::OsciGetFileAttributes (const char * aFileName, uint32 & aFileAttributes) [static]`

OsciGetFileAttributes utility function provides the various attributes of file (or directory) like if it is hidden, read only etc. The uint32 value is to be interpreted as per the enum OSCL_FILE_ATTRIBUTE_TYPE defined in [oscl_file_manager.h](#)

Parameters:

in] character path; the full path of the file or directory

out] file attributes.

Returns:

true if successful, otherwise false.

7.139.2.4 `OSCL_IMPORT_REF bool OsciFileManager::OsciGetFileAttributes (const oscl_wchar * aFileName, uint32 & aFileAttributes) [static]`

OsciGetFileAttributes utility function provides the various attributes of file (or directory) like if it is hidden, read only etc. The uint32 value is to be interpreted as per the enum OSCL_FILE_ATTRIBUTE_TYPE defined in [oscl_file_manager.h](#)

Parameters:

in] wide character path; the full path of the file or directory

out] file attributes.

Returns:

true if successful, otherwise false.

7.139.2.5 OSCL_IMPORT_REF bool OslFileManager::OslGetFileCreationTime (const char * *aFileName*, uint64 & *aFileCreationTime*) [static]

OslGetFileCreationTime utility function provides the file (or directory) creation time

Note:

On symbian platform, this api returns last modified time.

Parameters:

in] character path; the full path of the file or directory

out] creation time in microseconds.

Returns:

true if successful, otherwise false.

7.139.2.6 OSCL_IMPORT_REF bool OslFileManager::OslGetFileCreationTime (const oscl_wchar * *aFileName*, uint64 & *aFileCreationTime*) [static]

OslGetFileCreationTime utility function provides the file (or directory) creation time

Note:

On symbian platform, this api returns last modified time.

Parameters:

in] wide character path; the full path of the file or directory

out] creation time in microseconds

Returns:

true if successful, otherwise false.

7.139.2.7 OSCL_IMPORT_REF bool OslFileManager::OslGetFileLastAccessTime (const char * *aFileName*, uint64 & *aFileLastAccessTime*) [static]

OslGetFileLastAccessTime utility function provides the file (or directory) last access time, which might be different from last modified time.

Note:

On symbian platform, this api returns last modified time.

Parameters:

in] character path; the full path of the file or directory

out] Last access time in microseconds.

Returns:

true if successful, otherwise false.

7.139.2.8 OSCL_IMPORT_REF bool OsciFileManager::OsciGetFileLastAccessTime (const osci_wchar * aFileName, uint64 & aFileLastAccessTime) [static]

OsciGetFileLastAccessTime utility function provides the file (or directory) last access time, which might be different from last modified time.

Note:

On symbian platform, this api returns last modified time.

Parameters:

in] wide character path; the full path of the file or directory

out] Last access time in microseconds

Returns:

true if successful, otherwise false.

7.139.2.9 OSCL_IMPORT_REF bool OsciFileManager::OsciGetFileLastWriteTime (const char * aFileName, uint64 & aFileLastWriteTime) [static]

OsciGetFileLastWriteTime utility function provides the file (or directory) last modified time.

Parameters:

in] character path; the full path of the file or directory

out] last modified time in microseconds

Returns:

true if successful, otherwise false.

7.139.2.10 OSCL_IMPORT_REF bool OsciFileManager::OsciGetFileLastWriteTime (const osci_wchar * aFileName, uint64 & aFileLastWriteTime) [static]

OsciGetFileLastWriteTime utility function provides the file (or directory) last modified time.

Parameters:

in] wide character path; the full path of the file or directory

out] last modified time in microseconds

Returns:

true if successful, otherwise false.

7.139.2.11 OSCL_IMPORT_REF bool OsciFileManager::OsciGetFileSize (const char * aFileName, uint64 & aFileSize) [static]

OsciGetFileSize utility function provides the file size. For directory, this value is undefined.

Parameters:

in] character path; the full path of the file or directory

out] file size in bytes.

Returns:

true if successful, otherwise false.

7.139.2.12 OSCL_IMPORT_REF `bool OslFileManager::OslGetFileSize (const oscl_wchar * aFileName, uint64 & aFileSize)` [static]

OslGetFileSize utility function provides the file size. For directory, this value is undefined. creation time

Parameters:

- in*] wide character path; the full path of the file or directory
- out*] file size in bytes

Returns:

true if successful, otherwise false.

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

- [oscl_file_manager.h](#)

7.140 OsciFileStats Class Reference

```
#include <osci_file_stats.h>
```

Public Methods

- [OsciFileStats](#) ([Osci_File](#) *c)
- void [Start](#) (uint32 &aTicks)
- void [End](#) ([TOsciFileOp](#) aOp, uint32 aStart, uint32 aParam=0, [TOsciFileOffset](#) aParam2=0)
- void [Log](#) ([TOsciFileOp](#), [PVLogger](#) *, uint32)
- void [LogAll](#) ([PVLogger](#) *, uint32)

7.140.1 Constructor & Destructor Documentation

7.140.1.1 [OsciFileStats::OsciFileStats](#) ([Osci_File](#) * c)

7.140.2 Member Function Documentation

7.140.2.1 void [OsciFileStats::End](#) ([TOsciFileOp](#) aOp, uint32 aStart, uint32 aParam = 0, [TOsciFileOffset](#) aParam2 = 0)

7.140.2.2 void [OsciFileStats::Log](#) ([TOsciFileOp](#), [PVLogger](#) *, uint32)

7.140.2.3 void [OsciFileStats::LogAll](#) ([PVLogger](#) *, uint32)

7.140.2.4 void [OsciFileStats::Start](#) (uint32 & aTicks)

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

- [osci_file_stats.h](#)

7.141 OsciFileStatsItem Class Reference

```
#include <osci_file_stats.h>
```

Data Fields

- [uint32 iOpCount](#)
- [uint32 iParam](#)
- [TOsciFileOffset iParam2](#)
- [uint32 iStartTick](#)
- [uint32 iTotalTicks](#)

7.141.1 Field Documentation

7.141.1.1 [uint32 OsciFileStatsItem::iOpCount](#)

7.141.1.2 [uint32 OsciFileStatsItem::iParam](#)

7.141.1.3 [TOsciFileOffset](#) [OsciFileStatsItem::iParam2](#)

7.141.1.4 [uint32 OsciFileStatsItem::iStartTick](#)

7.141.1.5 [uint32 OsciFileStatsItem::iTotalTicks](#)

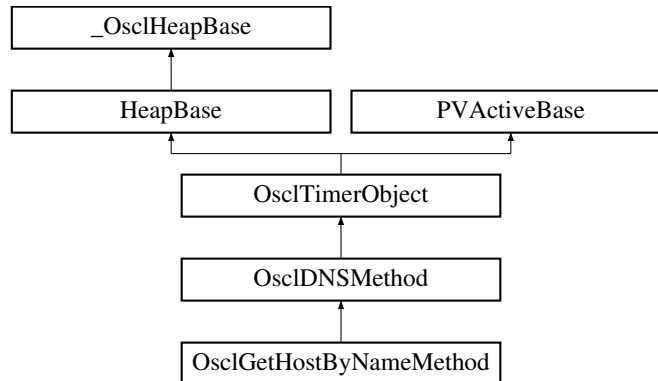
The documentation for this class was generated from the following file:

- [osci_file_stats.h](#)

7.142 OsciGetHostByNameMethod Class Reference

```
#include <osci_dns_gethostbyname.h>
```

Inheritance diagram for OsciGetHostByNameMethod::



Public Methods

- [~OsciGetHostByNameMethod \(\)](#)
- [TPVDNSEvent GetHostByName \(char *name, OsciNetworkAddress *addr, int32 aTimeout, Osci_Vector< OsciNetworkAddress, OsciMemAllocator > *aAddressList\)](#)

Static Public Methods

- [OsciGetHostByNameMethod * NewL \(Osci_DefAlloc &a, OsciDNSI *aDNS, OsciDNSObserver *aObserver, uint32 aId\)](#)

7.142.1 Constructor & Destructor Documentation

7.142.1.1 [OsciGetHostByNameMethod::~~OsciGetHostByNameMethod \(\)](#)

7.142.2 Member Function Documentation

7.142.2.1 [TPVDNSEvent OsciGetHostByNameMethod::GetHostByName \(char * name, OsciNetworkAddress * addr, int32 aTimeout, Osci_Vector< OsciNetworkAddress, OsciMemAllocator > * aAddressList\)](#)

7.142.2.2 [OsciGetHostByNameMethod* OsciGetHostByNameMethod::NewL \(Osci_DefAlloc &a, OsciDNSI * aDNS, OsciDNSObserver * aObserver, uint32 aId\) \[static\]](#)

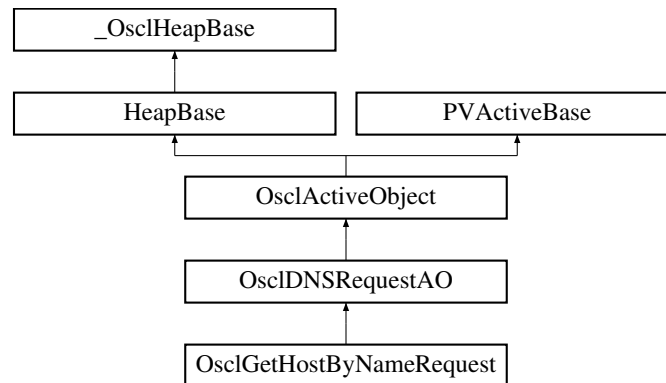
The documentation for this class was generated from the following file:

- [osci_dns_gethostbyname.h](#)

7.143 OsciGetHostByNameRequest Class Reference

```
#include <osci_dns_gethostbyname.h>
```

Inheritance diagram for OsciGetHostByNameRequest::



Friends

- class [OsciGetHostByNameMethod](#)

7.143.1 Friends And Related Function Documentation

7.143.1.1 friend class OsciGetHostByNameMethod [friend]

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

- [osci_dns_gethostbyname.h](#)

7.144 OslcInit Class Reference

```
#include <oscl_init.h>
```

Static Public Methods

- OSLC_IMPORT_REF void [Init](#) (int32 &aError, const [OslcSelect](#) *aSelect=NULL)
- OSLC_IMPORT_REF void [Cleanup](#) (int32 &aError, const [OslcSelect](#) *aSelect=NULL)

7.144.1 Detailed Description

Per-thread oscl initialization and cleanup.

7.144.2 Member Function Documentation

7.144.2.1 OSLC_IMPORT_REF void OslcInit::Cleanup (int32 & aError, const [OslcSelect](#) * aSelect = NULL) [static]

This routine cleans up the Oslc 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.

7.144.2.2 OSLC_IMPORT_REF void OslcInit::Init (int32 & aError, const [OslcSelect](#) * aSelect = NULL) [static]

This routine initializes the Oslc 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](#)

7.145 OsciInteger64Transport Struct Reference

```
#include <osci_int64_utils.h>
```

Data Fields

- uint32 [iHigh](#)
- uint32 [iLow](#)

7.145.1 Detailed Description

OsciInteger64Transport 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.

7.145.2 Field Documentation

7.145.2.1 uint32 OsciInteger64Transport::iHigh

7.145.2.2 uint32 OsciInteger64Transport::iLow

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

- [osci_int64_utils.h](#)

7.146 OsclIpMReq Class Reference

```
#include <oscl_socket_types.h>
```

Public Methods

- [OsclIpMReq](#) (const char *intrfcAddr, const char *multcstAddr)

Data Fields

- [OsclNameString](#)< PVNETWORKADDRESS_LEN > [interfaceAddr](#)
- [OsclNameString](#)< PVNETWORKADDRESS_LEN > [multicastAddr](#)

7.146.1 Constructor & Destructor Documentation

7.146.1.1 [OsclIpMReq::OsclIpMReq](#) (const char * *intrfcAddr*, const char * *multcstAddr*)
[inline]

7.146.2 Field Documentation

7.146.2.1 [OsclNameString](#)<PVNETWORKADDRESS_LEN> [OsclIpMReq::interfaceAddr](#)

7.146.2.2 [OsclNameString](#)<PVNETWORKADDRESS_LEN> [OsclIpMReq::multicastAddr](#)

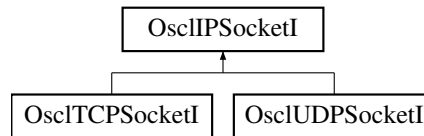
The documentation for this class was generated from the following file:

- [oscl_socket_types.h](#)

7.147 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)
- int32 [SetOptionToReuseAddress](#) ()
- int32 [SetTOS](#) (const [OsclSocketTOS](#) &aTOS)
- int32 [GetPeerName](#) ([OsclNetworkAddress](#) &peerName)
- virtual int32 [Close](#) ()=0
- virtual uint8 * [GetRecvData](#) (int32 *aLength)=0
- virtual uint8 * [GetSendData](#) (int32 *aLength)=0
- virtual [~OsclIPSocketI](#) ()
- void [ThreadLogoff](#) ()
- void [ThreadLogon](#) ([OsclSocketObserver](#) *aObs, [OsclSocketServI](#) *aServ)
- [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](#)

7.147.1 Constructor & Destructor Documentation

7.147.1.1 `virtual OsciPsocketI::~~OsciPsocketI ()` [inline, virtual]

7.147.1.2 `OsciPsocketI::OsciPsocketI (OsciDefAlloc & a)` [inline, protected]

7.147.2 Member Function Documentation

7.147.2.1 `OsciDefAlloc& OsciPsocketI::Alloc ()` [inline]

7.147.2.2 `int32 OsciPsocketI::Bind (OsciNetworkAddress & aAddress)`

7.147.2.3 `virtual int32 OsciPsocketI::Close ()` [pure virtual]

Implemented in [OsciTCPSocketI](#), and [OsciUDPSocketI](#).

7.147.2.4 `void OsciPsocketI::ConstructL (OsciSocketObserver * aObs, OsciSocketI * aSock, OsciSocketServI * aServ, uint32 aId)` [protected]

7.147.2.5 `int32 OsciPsocketI::GetPeerName (OsciNetworkAddress & peerName)`

7.147.2.6 `virtual uint8* OsciPsocketI::GetRecvData (int32 * aLength)` [pure virtual]

Implemented in [OsciTCPSocketI](#), and [OsciUDPSocketI](#).

7.147.2.7 `virtual uint8* OsciPsocketI::GetSendData (int32 * aLength)` [pure virtual]

Implemented in [OsciTCPSocketI](#), and [OsciUDPSocketI](#).

7.147.2.8 `int32 OsciPsocketI::Join (OsciNetworkAddress & aAddress)`

7.147.2.9 `int32 OsciPsocketI::SetOptionToReuseAddress ()`

7.147.2.10 `int32 OsciPsocketI::SetRecvBufferSize (uint32 size)`

7.147.2.11 `int32 OsciPsocketI::SetTOS (const OsciSocketTOS & aTOS)`

7.147.2.12 `OsciSocketServI* OsciPsocketI::SocketServ ()` [inline]

7.147.2.13 `void OsciPsocketI::ThreadLogoff ()`

Reimplemented in [OsciTCPSocketI](#), and [OsciUDPSocketI](#).

7.147.2.14 void OsciPsocketI::ThreadLogon ([OsciSocketObserver](#) * *aObs*, [OsciSocketServI](#) * *aServ*)

7.147.3 Friends And Related Function Documentation

7.147.3.1 friend class OsciSocketMethod [friend]

7.147.3.2 friend class OsciSocketRequestAO [friend]

7.147.4 Field Documentation

7.147.4.1 [OsciNetworkAddress](#) OsciPsocketI::iAddress [protected]

7.147.4.2 [Osci_DefAlloc&](#) OsciPsocketI::iAlloc [protected]

7.147.4.3 uint32 OsciPsocketI::iId [protected]

7.147.4.4 [PVLogger*](#) OsciPsocketI::iLogger [protected]

7.147.4.5 [OsciSocketObserver*](#) OsciPsocketI::iObserver [protected]

7.147.4.6 [OsciSocketI*](#) OsciPsocketI::iSocket [protected]

7.147.4.7 [OsciSocketServI*](#) OsciPsocketI::iSocketServ [protected]

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

- [osci_ip_socket.h](#)

7.148 OslJump Class Reference

```
#include <oscl_error_imp_jumps.h>
```

Public Methods

- void [Jump](#) (int a)
- jmp_buf * [Top](#) ()
- [~OslJump](#) ()

Static Public Methods

- OSCL_IMPORT_REF void [StaticJump](#) (int a)

Friends

- class [OslErrorTrapImp](#)

7.148.1 Constructor & Destructor Documentation

7.148.1.1 [OslJump::~~OslJump](#) () [inline]

7.148.2 Member Function Documentation

7.148.2.1 void [OslJump::Jump](#) (int *a*) [inline]

7.148.2.2 OSCL_IMPORT_REF void [OslJump::StaticJump](#) (int *a*) [static]

7.148.2.3 jmp_buf* [OslJump::Top](#) () [inline]

7.148.3 Friends And Related Function Documentation

7.148.3.1 friend class [OslErrorTrapImp](#) [friend]

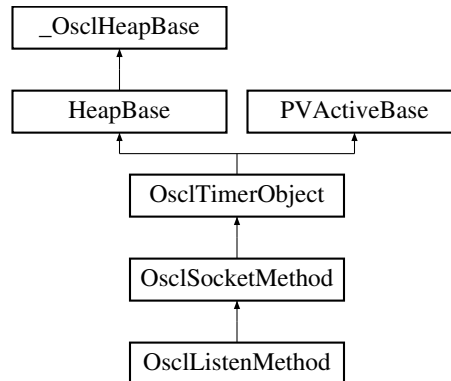
The documentation for this class was generated from the following file:

- [oscl_error_imp_jumps.h](#)

7.149 OsciListenMethod Class Reference

```
#include <osci_socket_listen.h>
```

Inheritance diagram for OsciListenMethod::



Public Methods

- [~OsciListenMethod \(\)](#)
- [TPVSocketEvent Listen \(uint32 qsize, int32 aTimeout\)](#)
- [OsciListenRequest * ListenRequest \(\)](#)

Static Public Methods

- [OsciListenMethod * NewL \(OsciIPSocketI &c\)](#)

7.149.1 Constructor & Destructor Documentation

7.149.1.1 [OsciListenMethod::~~OsciListenMethod \(\)](#)

7.149.2 Member Function Documentation

7.149.2.1 [TPVSocketEvent OsciListenMethod::Listen \(uint32 qsize, int32 aTimeout\)](#)

7.149.2.2 [OsciListenRequest* OsciListenMethod::ListenRequest \(\) \[inline\]](#)

7.149.2.3 [OsciListenMethod* OsciListenMethod::NewL \(OsciIPSocketI & c\) \[static\]](#)

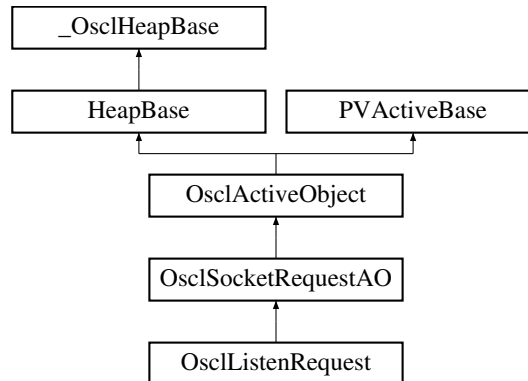
The documentation for this class was generated from the following file:

- [osci_socket_listen.h](#)

7.150 OscListenRequest Class Reference

```
#include <oscl_socket_listen.h>
```

Inheritance diagram for OscListenRequest::



Public Methods

- [OscListenRequest \(OscSocketMethod &c\)](#)
- void [Listen](#) (uint32 qsize)

7.150.1 Detailed Description

This is the AO that interacts with the socket server

7.150.2 Constructor & Destructor Documentation

7.150.2.1 [OscListenRequest::OscListenRequest \(OscSocketMethod & c\)](#) [inline]

7.150.3 Member Function Documentation

7.150.3.1 void [OscListenRequest::Listen](#) (uint32 qsize)

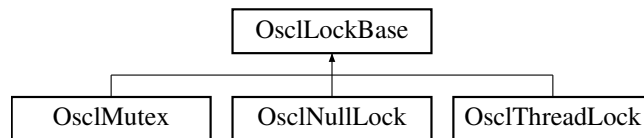
The documentation for this class was generated from the following file:

- [oscl_socket_listen.h](#)

7.151 OsciLockBase Class Reference

```
#include <osci_lock_base.h>
```

Inheritance diagram for OsciLockBase::



Public Methods

- virtual void [Lock](#) ()=0
- virtual void [Unlock](#) ()=0
- virtual [~OsciLockBase](#) ()

7.151.1 Constructor & Destructor Documentation

7.151.1.1 virtual [OsciLockBase::~OsciLockBase](#) () [inline, virtual]

7.151.2 Member Function Documentation

7.151.2.1 virtual void [OsciLockBase::Lock](#) () [pure virtual]

Implemented in [OsciNullLock](#), [OsciMutex](#), and [OsciThreadLock](#).

7.151.2.2 virtual void [OsciLockBase::Unlock](#) () [pure virtual]

Implemented in [OsciNullLock](#), [OsciMutex](#), and [OsciThreadLock](#).

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

- [osci_lock_base.h](#)

7.152 OslMem Class Reference

```
#include <oscl_mem.h>
```

Static Public Methods

- OSCL_IMPORT_REF void [Init](#) ()
- OSCL_IMPORT_REF void [Cleanup](#) ()

7.152.1 Member Function Documentation

7.152.1.1 OSCL_IMPORT_REF void OslMem::Cleanup () [static]

Per-thread cleanup of Osl Memory @exception: Leaves on error;

7.152.1.2 OSCL_IMPORT_REF void OslMem::Init () [static]

Per-thread initialization of Osl 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 [OslMem::Init](#) call in each thread. If no lock is provided, the memory manager will not be thread-safe. @exception: Leaves on error

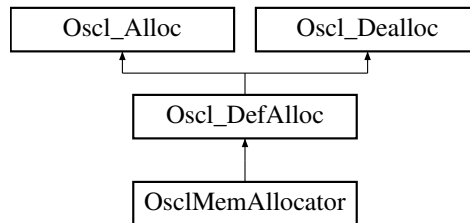
The documentation for this class was generated from the following file:

- [oscl_mem.h](#)

7.153 OsciMemAllocator Class Reference

```
#include <osci_mem.h>
```

Inheritance diagram for OsciMemAllocator::



Public Methods

- `OsciAny * allocate` (const uint32 n)
- `OsciAny * allocate_fl` (const uint32 n, const char *file_name, const int line_num)
- void `deallocate` (`OsciAny *p`)

7.153.1 Detailed Description

A simple allocator class. Configurable as to whether this goes through the memory manager or not.

7.153.2 Member Function Documentation

7.153.2.1 `OsciAny* OsciMemAllocator::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 OsciErrNoMemory)

Implements `Osci_DefAlloc`.

7.153.2.2 `OsciAny* OsciMemAllocator::allocate_fl` (const uint32 n, const char *file_name, const int line_num) [inline, virtual]

Reimplemented from `Osci_DefAlloc`.

7.153.2.3 void `OsciMemAllocator::deallocate` (`OsciAny *p`) [inline, virtual]

Implements `Osci_DefAlloc`.

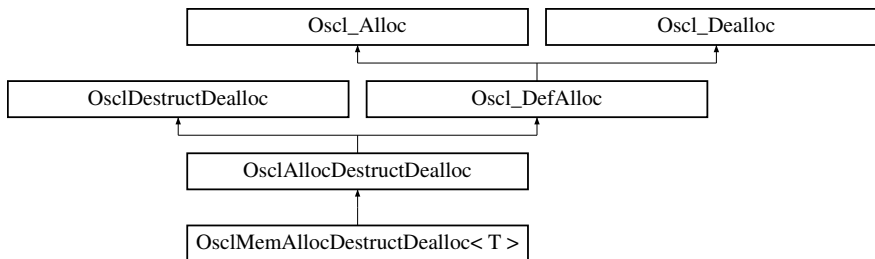
The documentation for this class was generated from the following file:

- [osci_mem.h](#)

7.154 OsciMemAllocDestructDealloc< T > Class Template Reference

```
#include <osci_mem.h>
```

Inheritance diagram for OsciMemAllocDestructDealloc< T >::



Public Methods

- [OsciAny](#) * [allocate_fl](#) (const uint32 size, const char *file_name, const int line_num)
- [OsciAny](#) * [allocate](#) (const uint32 size)
- void [deallocate](#) ([OsciAny](#) *p)
- void [destruct_and_dealloc](#) ([OsciAny](#) *p)

7.154.1 Detailed Description

`template<class T> class OsciMemAllocDestructDealloc< T >`

An [OsciAllocDestructDealloc](#) class that uses [OsciMemAllocator](#).

7.154.2 Member Function Documentation

7.154.2.1 `template<class T> OsciAny* OsciMemAllocDestructDealloc< T >::allocate (const uint32 size) [inline, virtual]`

Implements [Osci_DefAlloc](#).

7.154.2.2 `template<class T> OsciAny* OsciMemAllocDestructDealloc< T >::allocate_fl (const uint32 size, const char *file_name, const int line_num) [inline, virtual]`

Reimplemented from [Osci_DefAlloc](#).

7.154.2.3 `template<class T> void OsciMemAllocDestructDealloc< T >::deallocate (OsciAny *p) [inline, virtual]`

Implements [Osci_DefAlloc](#).

7.154.2.4 `template<class T> void OsciMemAllocDestructDealloc< T >::destruct_and_dealloc`
`(OsciAny *p)` [inline, virtual]

Implements [OsciDestructDealloc](#).

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

- [osci_mem.h](#)

7.155 OsciMemAudit Class Reference

```
#include <osci_mem_audit.h>
```

Public Methods

- [OsciMemAudit \(\)](#)
- [~OsciMemAudit \(\)](#)
- void * [MM_allocate](#) (const [OsciMemStatsNode](#) *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_GetTreeNodees](#) (const char *tagIn)
- bool [MM_AddTag](#) (const char *tagIn)
- const [OsciMemStatsNode](#) * [MM_GetTagNode](#) (const char *tagIn)
- const [OsciMemStatsNode](#) * [MM_GetExistingTag](#) (const char *tagIn)
- const [OsciMemStatsNode](#) * [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](#) ()
- [OsciLockBase](#) * [GetLock](#) ()

Friends

- class [OsciMemGlobalAuditObject](#)

7.155.1 Constructor & Destructor Documentation

7.155.1.1 OsciMemAudit::OsciMemAudit () [inline]

Constructor, create the root node in statistics table

7.155.1.2 OslMemAudit::~~OslMemAudit () [inline]

A destructor, remove all the nodes in allocation and statistics table

7.155.2 Member Function Documentation

7.155.2.1 OslLockBase* OslMemAudit::GetLock () [inline]

API to obtain mem lock ptr

7.155.2.2 bool OslMemAudit::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;

7.155.2.3 void* OslMemAudit::MM_allocate (const OslMemStatsNode * statsNode, uint32 sizeIn, const char * pFileName, uint32 lineNumber, bool allocNodeTracking = false) [inline]

The following are APIs t __nothrow_/ const __nothrow_

Returns:

the memory pointer if operation succeeds.

7.155.2.4 MM_AllocQueryInfo* OslMemAudit::MM_CreateAllocNodeInfo (uint32 max_array_size) [inline]

7.155.2.5 bool OslMemAudit::MM_deallocate (void * pMemBlockIn) [inline]

Returns:

true if operation succeeds;

7.155.2.6 uint32 OslMemAudit::MM_GetAllocNo (void) [inline]

API to get the current allocation number

Returns:

the current allocation number

7.155.2.7 `uint32 OsciMemAudit::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

7.155.2.8 `const OsciMemStatsNode* OsciMemAudit::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;

7.155.2.9 `uint32 OsciMemAudit::MM_GetMode (void) [inline]`

API to get the operating mode of the mm_audit class.

7.155.2.10 `uint32 OsciMemAudit::MM_GetNumAllocNodes () [inline]`

API to get the number of allocation nodes (records) for allocations that are being tracked individually.

7.155.2.11 `void OsciMemAudit::MM_GetOverheadStats (MM_AuditOverheadStats & stats) [inline]`

API to get the overhead statistics for the memory used by the mm_audit class.

7.155.2.12 `uint32 OsciMemAudit::MM_GetPostfillPattern (void) [inline]`

API to get the postfill pattern. The pattern is used to fill the memory before freeing it.

7.155.2.13 `uint8 OsciMemAudit::MM_GetPrefillPattern (void) [inline]`

API to get the prefill pattern. The pattern is used to fill the memory before returning it to the caller.

7.155.2.14 `int32 OslMemAudit::MM_GetRefCount ()` [inline]

7.155.2.15 `const OslMemStatsNode* OslMemAudit::MM_GetRootNode ()` [inline]

7.155.2.16 `MM_Stats_t* OslMemAudit::MM_GetStats (const char *const tagIn)` [inline]

API to get memory statistics through context string(tag)

Returns:

statistics pointer if operation succeeds

7.155.2.17 `uint32 OslMemAudit::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

7.155.2.18 `const OslMemStatsNode* OslMemAudit::MM_GetTagNode (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 `OslMemStatsNode` which should be passed to `MM_allocate`

7.155.2.19 `uint32 OslMemAudit::MM_GetTreeNodees (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

7.155.2.20 `void OslMemAudit::MM_ReleaseAllocNodeInfo (MM_AllocQueryInfo * info)` [inline]

7.155.2.21 `bool OslMemAudit::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;

7.155.2.22 void OsciMemAudit::MM_SetMode (uint32 inMode) [inline]

API to set the operating mode of the mm_audit class.

7.155.2.23 void OsciMemAudit::MM_SetPostfillPattern (uint8 pattern) [inline]

API to set the postfill pattern.

7.155.2.24 void OsciMemAudit::MM_SetPrefillPattern (uint8 pattern) [inline]

API to set the prefill pattern.

7.155.2.25 void OsciMemAudit::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

7.155.2.26 void OsciMemAudit::MM_UnsetFailurePoint (const char * tagIn) [inline]

API to cancel the allocation failure point associated with tag

Parameters:

tagIn input tag

7.155.2.27 bool OsciMemAudit::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;

7.155.3 Friends And Related Function Documentation

7.155.3.1 friend class OsciMemGlobalAuditObject [friend]

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

- [osci_mem_audit.h](#)

7.156 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](#)

7.156.1 Detailed Description

template<class T, class _Allocator = Osci_TAlloc<T, OsciMemAllocator>> class 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 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

7.156.2 Constructor & Destructor Documentation

7.156.2.1 **template<class T, class _Allocator = Osci_TAlloc<T, OsciMemAllocator>> OSCLMemAutoPtr< T, _Allocator >::OSCLMemAutoPtr (T * *inPtr* = 0) [inline, explicit]**

Default constructor Initializes the pointer and takes ownership.

7.156.2.2 **template<class T, class _Allocator = Osci_TAlloc<T, OsciMemAllocator>> 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.

7.156.2.3 **template<class T, class _Allocator = Osci_TAlloc<T, OsciMemAllocator>> OSCLMemAutoPtr< T, _Allocator >::~OSCLMemAutoPtr () [inline]**

Destructor.

The pointer is deleted in case this class still has ownership

7.156.3 Member Function Documentation

7.156.3.1 `template<class T, class _Allocator = Osci_TAlloc<T, OsciMemAllocator>> void OSCLMemAutoPtr< T, _Allocator >::allocate (oscl_memsize_t size) [inline]`

7.156.3.2 `template<class T, class _Allocator = Osci_TAlloc<T, OsciMemAllocator>> void OSCLMemAutoPtr< T, _Allocator >::deallocate (T *ptr) [inline, static]`

7.156.3.3 `template<class T, class _Allocator = Osci_TAlloc<T, OsciMemAllocator>> T* OSCLMemAutoPtr< T, _Allocator >::get () const [inline]`

`get()` method returns the pointer, currently owned by the class.

7.156.3.4 `template<class T, class _Allocator = Osci_TAlloc<T, OsciMemAllocator>> 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.

7.156.3.5 `template<class T, class _Allocator = Osci_TAlloc<T, OsciMemAllocator>> 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.

7.156.3.6 `template<class T, class _Allocator = Osci_TAlloc<T, OsciMemAllocator>> 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.

7.156.3.7 `template<class T, class _Allocator = Osci_TAlloc<T, OsciMemAllocator>> 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.

7.156.3.8 `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.

7.156.3.9 `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.

7.156.4 Field Documentation

7.156.4.1 `template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>> bool OSCLMemAutoPtr< T, _Allocator >::_Ownership`

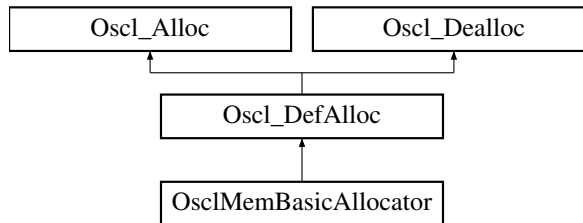
The documentation for this class was generated from the following file:

- [oscl_mem_auto_ptr.h](#)

7.157 OsciMemBasicAllocator Class Reference

```
#include <osci_mem.h>
```

Inheritance diagram for OsciMemBasicAllocator::



Public Methods

- [OsciAny * allocate](#) (const uint32 n)
- void [deallocate](#) (OsciAny *p)

7.157.1 Detailed Description

A simple allocator class that does not use the memory management.

Note: this allocator is for internal use by Osci only. Higher level code should use [OsciMemAllocator](#).

7.157.2 Member Function Documentation

7.157.2.1 [OsciAny* OsciMemBasicAllocator::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 OsciErrNoMemory)

Implements [Osci_DefAlloc](#).

7.157.2.2 void [OsciMemBasicAllocator::deallocate](#) (OsciAny *p) [inline, virtual]

Implements [Osci_DefAlloc](#).

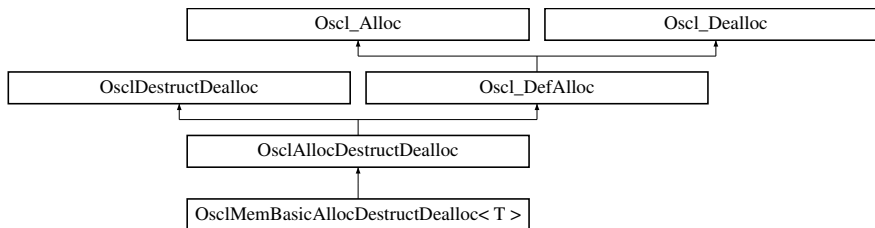
The documentation for this class was generated from the following file:

- [osci_mem.h](#)

7.158 OslMemBasicAllocDestructDealloc< T > Class Template Reference

```
#include <oscl_mem.h>
```

Inheritance diagram for OslMemBasicAllocDestructDealloc< T >::



Public Methods

- [OslAny * allocate](#) (const uint32 size)
- void [deallocate](#) ([OslAny *p](#))
- void [destruct_and_dealloc](#) ([OslAny *p](#))

7.158.1 Detailed Description

```
template<class T> class OslMemBasicAllocDestructDealloc< T >
```

An [OslAllocDestructDealloc](#) class that uses [OslMemBasicAllocator](#).

7.158.2 Member Function Documentation

7.158.2.1 `template<class T> OslAny* OslMemBasicAllocDestructDealloc< T >::allocate (const uint32 size) [inline, virtual]`

Implements [Osl_DefAlloc](#).

7.158.2.2 `template<class T> void OslMemBasicAllocDestructDealloc< T >::deallocate (OslAny *p) [inline, virtual]`

Implements [Osl_DefAlloc](#).

7.158.2.3 `template<class T> void OslMemBasicAllocDestructDealloc< T >::destruct_and_dealloc (OslAny *p) [inline, virtual]`

Implements [OslDestructDealloc](#).

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

- [oscl_mem.h](#)

7.159 OslMemGlobalAuditObject Class Reference

```
#include <oscl_mem.h>
```

Public Types

- typedef [OslMemAudit](#) `audit_type`

Static Public Methods

- `OSCL_IMPORT_REF` [audit_type](#) * `getGlobalMemAuditObject ()`

Friends

- class [OslMem](#)

7.159.1 Member Typedef Documentation

7.159.1.1 typedef [OslMemAudit](#) `OslMemGlobalAuditObject::audit_type`

7.159.2 Member Function Documentation

7.159.2.1 `OSCL_IMPORT_REF` [audit_type](#)* `OslMemGlobalAuditObject::getGlobalMemAuditObject ()` [`static`]

returns the global audit object. For use in macros only– not a public API.

7.159.3 Friends And Related Function Documentation

7.159.3.1 `friend class` [OslMem](#) [`friend`]

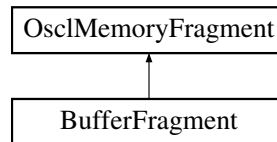
The documentation for this class was generated from the following file:

- [oscl_mem.h](#)

7.160 OslMemoryFragment Struct Reference

```
#include <oscl_types.h>
```

Inheritance diagram for OslMemoryFragment::



Data Fields

- uint32 [len](#)
- void * [ptr](#)

7.160.1 Field Documentation

7.160.1.1 [uint32 OslMemoryFragment::len](#)

7.160.1.2 [void* OslMemoryFragment::ptr](#)

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

- [oscl_types.h](#)

7.161 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)

7.161.1 Constructor & Destructor Documentation

7.161.1.1 [OsclMemPoolAllocator::OsclMemPoolAllocator](#) ([Oscl_DefAlloc](#) * *gen_alloc* = NULL)

7.161.1.2 virtual [OsclMemPoolAllocator::~OsclMemPoolAllocator](#) () [virtual]

7.161.2 Member Function Documentation

7.161.2.1 [OsclAny](#)* [OsclMemPoolAllocator::CreateMemPool](#) (const uint32 *aNumChunk* = 2, const uint32 *aChunkSize* = 4)

7.161.2.2 void [OsclMemPoolAllocator::DestroyMemPool](#) ()

7.161.2.3 [uint](#) [OsclMemPoolAllocator::oscl_mem_aligned_size](#) ([uint](#) *size*)

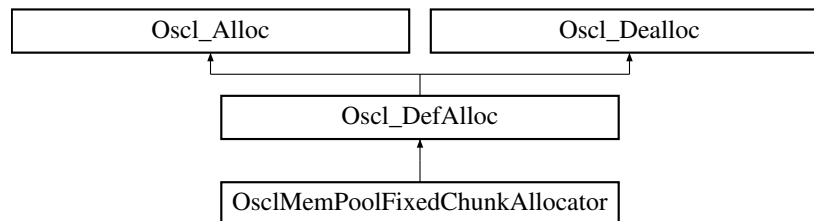
The documentation for this class was generated from the following file:

- [oscl_mempool_allocator.h](#)

7.162 OslcMemPoolFixedChunkAllocator Class Reference

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

Inheritance diagram for OslcMemPoolFixedChunkAllocator::



Public Methods

- OSCL_IMPORT_REF [OslcMemPoolFixedChunkAllocator](#) (const uint32 numchunk=1, const uint32 chunksize=0, [Oslc_DefAlloc](#) *gen_alloc=NULL, const uint32 chunkalignment=0)
- virtual OSCL_IMPORT_REF void [enablenullpointerreturn](#) ()
- virtual OSCL_IMPORT_REF [~OslcMemPoolFixedChunkAllocator](#) ()
- virtual OSCL_IMPORT_REF [OslcAny](#) * [allocate](#) (const uint32 n)
- virtual OSCL_IMPORT_REF void [deallocate](#) ([OslcAny](#) *p)
- virtual OSCL_IMPORT_REF void [notifyfreechunkavailable](#) ([OslcMemPoolFixedChunkAllocatorObserver](#) &obs, [OslcAny](#) *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](#)
- uint32 [iChunkAlignment](#)
- [Oslc_DefAlloc](#) * [iMemPoolAllocator](#)
- [OslcAny](#) * [iMemPool](#)
- [OslcAny](#) * [iMemPoolAligned](#)
- [Oslc_Vector](#)< [OslcAny](#) *, [OslcMemAllocator](#) > [iFreeMemChunkList](#)
- bool [iCheckNextAvailableFreeChunk](#)
- [OslcMemPoolFixedChunkAllocatorObserver](#) * [iObserver](#)
- [OslcAny](#) * [iNextAvailableContextData](#)
- int32 [iRefCount](#)
- bool [iEnableNullPtrReturn](#)

7.162.1 Constructor & Destructor Documentation

7.162.1.1 `OSCL_IMPORT_REF OslMemPoolFixedChunkAllocator::OslMemPoolFixedChunkAllocator (const uint32 numchunk = 1, const uint32 chunksize = 0, Osl_DefAlloc * gen_alloc = NULL, const uint32 chunkalignment = 0)`

This API throws an exception when the memory allocation for pool fails. If *numchunk* and *chunksize* 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*. If *chunkalignment* is set to 0, memory pool will use default allocator alignment (8-byte). If *chunkalignment* is > 0, memory pool will align all buffers in the mempool to the specified alignment. Alignment should be a power of 2.

Returns:

void

7.162.1.2 `virtual OSCL_IMPORT_REF OslMemPoolFixedChunkAllocator::~~OslMemPoolFixedChunkAllocator () [virtual]`

The destructor for the memory pool.

7.162.2 Member Function Documentation

7.162.2.1 `OSCL_IMPORT_REF void OslMemPoolFixedChunkAllocator::addRef ()`

Increments the reference count for this memory pool allocator.

Returns:

void

7.162.2.2 `virtual OSCL_IMPORT_REF OslAny* OslMemPoolFixedChunkAllocator::allocate (const uint32 n) [virtual]`

This API throws an exception when *n* is greater than the fixed chunk size or there are no free chunks 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 [Osl_DefAlloc](#).

7.162.2.3 `virtual OSCL_IMPORT_REF void OslMemPoolFixedChunkAllocator::CancelFreeChunkAvailableCallback () [virtual]`

This API will cancel any past callback requests..

Returns:

void

7.162.2.4 virtual OSCL_IMPORT_REF void OsciMemPoolFixedChunkAllocator::createmempool () [protected, virtual]

7.162.2.5 virtual OSCL_IMPORT_REF void OsciMemPoolFixedChunkAllocator::deallocate (OsciAny *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 [Osci_DefAlloc](#).

7.162.2.6 virtual OSCL_IMPORT_REF void OsciMemPoolFixedChunk-Allocator::destroymempool () [protected, virtual]

7.162.2.7 virtual OSCL_IMPORT_REF void OsciMemPoolFixedChunk-Allocator::enablenullpointerreturn () [virtual]

This API will disable exceptions in case the memory pool runs out of memory Instead of doing "[OSCL_LEAVE\(OsciErrNoResources\)](#)" allocate API will return NULL.

Returns:

void

7.162.2.8 virtual OSCL_IMPORT_REF void OsciMemPoolFixedChunk-Allocator::notifyfreechunkavailable (OsciMemPoolFixedChunkAllocatorObserver &obs, OsciAny * 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

7.162.2.9 OSCL_IMPORT_REF void OsciMemPoolFixedChunkAllocator::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

7.162.3 Field Documentation

- 7.162.3.1 **bool** `OsclMemPoolFixedChunkAllocator::iCheckNextAvailableFreeChunk` [protected]
- 7.162.3.2 **uint32** `OsclMemPoolFixedChunkAllocator::iChunkAlignment` [protected]
- 7.162.3.3 **uint32** `OsclMemPoolFixedChunkAllocator::iChunkSize` [protected]
- 7.162.3.4 **uint32** `OsclMemPoolFixedChunkAllocator::iChunkSizeMemAligned` [protected]
- 7.162.3.5 **bool** `OsclMemPoolFixedChunkAllocator::iEnableNullPtrReturn` [protected]
- 7.162.3.6 **Oscl_Vector**<**OsclAny***, **OsclMemAllocator**> `OsclMemPoolFixedChunkAllocator::iFreeMemChunkList` [protected]
- 7.162.3.7 **OsclAny*** `OsclMemPoolFixedChunkAllocator::iMemPool` [protected]
- 7.162.3.8 **OsclAny*** `OsclMemPoolFixedChunkAllocator::iMemPoolAligned` [protected]
- 7.162.3.9 **Oscl_DefAlloc*** `OsclMemPoolFixedChunkAllocator::iMemPoolAllocator` [protected]
- 7.162.3.10 **OsclAny*** `OsclMemPoolFixedChunkAllocator::iNextAvailableContextData` [protected]
- 7.162.3.11 **uint32** `OsclMemPoolFixedChunkAllocator::iNumChunk` [protected]
- 7.162.3.12 **OsclMemPoolFixedChunkAllocatorObserver*** `OsclMemPoolFixedChunkAllocator::iObserver` [protected]
- 7.162.3.13 **int32** `OsclMemPoolFixedChunkAllocator::iRefCount` [protected]

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

- [oscl_mem_mempool.h](#)

7.163 OsciMemPoolFixedChunkAllocatorObserver Class Reference

```
#include <osci_mem_mempool.h>
```

Public Methods

- virtual void [freechunkavailable](#) (OsciAny *aContextData)=0
- virtual [~OsciMemPoolFixedChunkAllocatorObserver](#) ()

7.163.1 Constructor & Destructor Documentation

7.163.1.1 virtual OsciMemPoolFixedChunkAllocatorObserver::~~OsciMemPoolFixedChunkAllocatorObserver () [inline, virtual]

7.163.2 Member Function Documentation

7.163.2.1 virtual void OsciMemPoolFixedChunkAllocatorObserver::freechunkavailable (OsciAny * *aContextData*) [pure virtual]

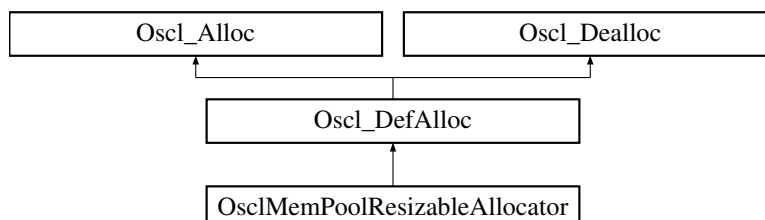
The documentation for this class was generated from the following file:

- [osci_mem_mempool.h](#)

7.164 OsciMemPoolResizableAllocator Class Reference

```
#include <osci_mem_mempool.h>
```

Inheritance diagram for OsciMemPoolResizableAllocator::



Public Methods

- OSCL_IMPORT_REF [OsciMemPoolResizableAllocator](#) (uint32 aMemPoolBufferSize, uint32 aMemPoolBufferNumLimit=0, uint32 aExpectedNumBlocksPerBuffer=0, [Osci_DefAlloc](#) *gen_alloc=NULL)
- virtual OSCL_IMPORT_REF void [enablenullpointerreturn](#) ()
- virtual OSCL_IMPORT_REF [OsciAny](#) * [allocate](#) (const uint32 aNumBytes)
- virtual OSCL_IMPORT_REF void [deallocate](#) ([OsciAny](#) *aPtr)
- virtual OSCL_IMPORT_REF bool [trim](#) ([OsciAny](#) *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](#) ([OsciMemPoolResizableAllocator-Observer](#) &aObserver, uint32 aRequestedSize=0, [OsciAny](#) *aContextData=NULL)
- virtual OSCL_IMPORT_REF void [CancelFreeChunkAvailableCallback](#) ()
- virtual OSCL_IMPORT_REF void [notifyfreememoryavailable](#) ([OsciMemPoolResizableAllocator-MemoryObserver](#) &aObserver, uint32 aRequestedSize=0, [OsciAny](#) *aContextData=NULL)
- OSCL_IMPORT_REF void [CancelFreeMemoryAvailableCallback](#) ()
- OSCL_IMPORT_REF void [addRef](#) ()
- OSCL_IMPORT_REF void [removeRef](#) ()

Protected Methods

- virtual OSCL_IMPORT_REF [~OsciMemPoolResizableAllocator](#) ()
- [MemPoolBufferInfo](#) * [addnewmempoolbuffer](#) (uint32 aBufferSize)
- void [destroyallmempoolbuffers](#) ()
- [MemPoolBlockInfo](#) * [findfreeblock](#) (uint32 aBlockSize)
- [OsciAny](#) * [allocateblock](#) ([MemPoolBlockInfo](#) &aBlockPtr, uint32 aNumBytes)
- void [deallocateblock](#) ([MemPoolBlockInfo](#) &aBlockPtr)
- bool [validateblock](#) ([OsciAny](#) *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`

7.164.1 Constructor & Destructor Documentation

7.164.1.1 `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

7.164.1.2 `virtual OSCL_IMPORT_REF OsclMemPoolResizableAllocator::~~OsclMemPoolResizableAllocator ()` [protected, virtual]

The destructor for the memory pool. Should not be called directly. Use `removeRef()` instead.

7.164.2 Member Function Documentation

7.164.2.1 `MemPoolBufferInfo* OsclMemPoolResizableAllocator::addnewmempoolbuffer (uint32 aBufferSize)` [protected]

7.164.2.2 `OSCL_IMPORT_REF void OsclMemPoolResizableAllocator::addRef ()`

Increments the reference count for this memory pool allocator

Returns:

void

7.164.2.3 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](#).

7.164.2.4 [OsclAny*](#) [OsclMemPoolResizableAllocator::allocatblock](#) ([MemPoolBlockInfo](#) & *aBlockPtr*, uint32 *aNumBytes*) [protected]

7.164.2.5 virtual OSCL_IMPORT_REF void [OsclMemPoolResizableAllocator::CancelFreeChunkAvailableCallback](#) () [virtual]

This API will cancel any past callback requests..

Returns:

void

7.164.2.6 OSCL_IMPORT_REF void [OsclMemPoolResizableAllocator::CancelFreeMemoryAvailableCallback](#) ()

7.164.2.7 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](#).

7.164.2.8 void [OsclMemPoolResizableAllocator::deallocateblock](#) ([MemPoolBlockInfo](#) & *aBlockPtr*) [protected]

7.164.2.9 void [OsclMemPoolResizableAllocator::destroyallmempoolbuffers](#) () [protected]

7.164.2.10 virtual OSCL_IMPORT_REF void [OsclMemPoolResizableAllocator::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

7.164.2.11 [MemPoolBlockInfo*](#) OsciMemPoolResizableAllocator::findfreeblock (uint32 *aBlockSize*) [protected]

7.164.2.12 virtual OSCL_IMPORT_REF uint32 OsciMemPoolResizableAllocator::getAllocated-Size () [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>

7.164.2.13 virtual OSCL_IMPORT_REF uint32 OsciMemPoolResizableAllocator::getAvailable-Size () [virtual]

Returns the number of bytes available with the buffer

7.164.2.14 OSCL_IMPORT_REF uint32 OsciMemPoolResizableAllocator::getBufferSize ()

Returns the size of the buffer <including the overhead bytes that may be allocated by the allocator>

7.164.2.15 virtual OSCL_IMPORT_REF uint32 OsciMemPoolResizableAllocator::getLargest-ContiguousFreeBlockSize () [virtual]

Returns the size of the largest available chunk in the memory.

7.164.2.16 uint32 OsciMemPoolResizableAllocator::getMemPoolBufferAllocatedSize ([MemPoolBufferInfo](#) * *aBufferInfo*) const [protected]

7.164.2.17 uint32 OsciMemPoolResizableAllocator::getMemPoolBufferSize ([MemPoolBufferInfo](#) * *aBufferInfo*) const [protected]

7.164.2.18 uint32 OsciMemPoolResizableAllocator::memoryPoolBufferMgmtOverhead () [protected]

7.164.2.19 virtual OSCL_IMPORT_REF void OsciMemPoolResizable-Allocator::notifyfreeblockavailable ([OsciMemPoolResizableAllocatorObserver](#) & *aObserver*, uint32 *aRequestedSize* = 0, [OsciAny](#) * *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

7.164.2.20 virtual OSCL_IMPORT_REF void OsclMemPoolResizableAllocator::notifyfreememoryavailable (OsclMemPoolResizableAllocatorMemoryObserver & aObserver, uint32 aRequestedSize = 0, OsclAny * aContextData = NULL) [virtual]

7.164.2.21 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

7.164.2.22 virtual OSCL_IMPORT_REF bool OsclMemPoolResizableAllocator::setMaxSzForNewMemPoolBuffer (uint32 aMaxNewMemPoolBufferSz) [virtual]

7.164.2.23 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 the 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

7.164.2.24 `bool OsciMemPoolResizableAllocator::validateblock (OsciAny * aBlockBufPtr)`
 [protected]

7.164.3 Field Documentation

7.164.3.1 `uint32 OsciMemPoolResizableAllocator::iBlockInfoAlignedSize` [protected]

7.164.3.2 `uint32 OsciMemPoolResizableAllocator::iBufferInfoAlignedSize` [protected]

7.164.3.3 `bool OsciMemPoolResizableAllocator::iCheckFreeMemoryAvailable` [protected]

7.164.3.4 `bool OsciMemPoolResizableAllocator::iCheckNextAvailable` [protected]

7.164.3.5 `bool OsciMemPoolResizableAllocator::iEnableNullPtrReturn` [protected]

7.164.3.6 `uint32 OsciMemPoolResizableAllocator::iExpectedNumBlocksPerBuffer`
 [protected]

7.164.3.7 `OsciAny* OsciMemPoolResizableAllocator::iFreeMemContextData` [protected]

7.164.3.8 `OsciMemPoolResizableAllocatorMemoryObserver* OsciMemPoolResizable-
 Allocator::iFreeMemPoolObserver` [protected]

7.164.3.9 `uint32 OsciMemPoolResizableAllocator::iMaxNewMemPoolBufferSz` [protected]

7.164.3.10 `Osci_DefAlloc* OsciMemPoolResizableAllocator::iMemPoolBufferAllocator`
 [protected]

7.164.3.11 `Osci_Vector<MemPoolBufferInfo*, OsciMemAllocator>
 OsciMemPoolResizableAllocator::iMemPoolBufferList` [protected]

7.164.3.12 `uint32 OsciMemPoolResizableAllocator::iMemPoolBufferNumLimit` [protected]

7.164.3.13 `uint32 OsciMemPoolResizableAllocator::iMemPoolBufferSize` [protected]

7.164.3.14 `OsciAny* OsciMemPoolResizableAllocator::iNextAvailableContextData`
 [protected]

7.164.3.15 `OsciMemPoolResizableAllocatorObserver* OsciMemPoolResizableAllocator::i-
 Observer` [protected]

7.164.3.16 `int32 OsciMemPoolResizableAllocator::iRefCount` [protected]

7.164.3.17 `uint32 OsciMemPoolResizableAllocator::iRequestedAvailableFreeMemSize`
 [protected]

7.164.3.18 `uint32 OsciMemPoolResizableAllocator::iRequestedNextAvailableSize`
 [protected]

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

- [osci_mem_mempool.h](#)

7.165 OsciMemPoolResizableAllocator::MemPoolBlockInfo Struct Reference

```
#include <osci_mem_mempool.h>
```

Data Fields

- uint32 [iBlockPreFence](#)
- MemPoolBlockInfo * [iNextFreeBlock](#)
- MemPoolBlockInfo * [iPrevFreeBlock](#)
- uint32 [iBlockSize](#)
- uint8 * [iBlockBuffer](#)
- [MemPoolBufferInfo](#) * [iParentBuffer](#)
- uint32 [iBlockPostFence](#)

7.165.1 Field Documentation

7.165.1.1 uint8* [OsciMemPoolResizableAllocator::MemPoolBlockInfo::iBlockBuffer](#)

7.165.1.2 uint32 [OsciMemPoolResizableAllocator::MemPoolBlockInfo::iBlockPostFence](#)

7.165.1.3 uint32 [OsciMemPoolResizableAllocator::MemPoolBlockInfo::iBlockPreFence](#)

7.165.1.4 uint32 [OsciMemPoolResizableAllocator::MemPoolBlockInfo::iBlockSize](#)

7.165.1.5 [MemPoolBlockInfo](#)* [OsciMemPoolResizableAllocator::MemPoolBlockInfo::iNextFree-Block](#)

7.165.1.6 [MemPoolBufferInfo](#)* [OsciMemPoolResizableAllocator::MemPoolBlockInfo::iParent-Buffer](#)

7.165.1.7 [MemPoolBlockInfo](#)* [OsciMemPoolResizableAllocator::MemPoolBlockInfo::iPrevFree-Block](#)

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

- [osci_mem_mempool.h](#)

7.166 OslMemPoolResizableAllocator::MemPoolBufferInfo Struct Reference

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

Data Fields

- [uint32 iBufferPreFence](#)
- [OslAny * iStartAddr](#)
- [OslAny * iEndAddr](#)
- [uint32 iBufferSize](#)
- [uint32 iNumOutstanding](#)
- [MemPoolBlockInfo * iNextFreeBlock](#)
- [uint32 iAllocatedSz](#)
- [uint32 iBufferPostFence](#)

7.166.1 Field Documentation

7.166.1.1 [uint32 OslMemPoolResizableAllocator::MemPoolBufferInfo::iAllocatedSz](#)

7.166.1.2 [uint32 OslMemPoolResizableAllocator::MemPoolBufferInfo::iBufferPostFence](#)

7.166.1.3 [uint32 OslMemPoolResizableAllocator::MemPoolBufferInfo::iBufferPreFence](#)

7.166.1.4 [uint32 OslMemPoolResizableAllocator::MemPoolBufferInfo::iBufferSize](#)

7.166.1.5 [OslAny*](#) [OslMemPoolResizableAllocator::MemPoolBufferInfo::iEndAddr](#)

7.166.1.6 [MemPoolBlockInfo*](#) [OslMemPoolResizableAllocator::MemPoolBufferInfo::iNextFreeBlock](#)

7.166.1.7 [uint32 OslMemPoolResizableAllocator::MemPoolBufferInfo::iNumOutstanding](#)

7.166.1.8 [OslAny*](#) [OslMemPoolResizableAllocator::MemPoolBufferInfo::iStartAddr](#)

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

- [oscl_mem_mempool.h](#)

7.167 OsclMemPoolResizableAllocatorMemoryObserver Class Reference

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

Public Methods

- virtual void [freememoryavailable](#) ([OsclAny](#) *aContextData)=0
- virtual [~OsclMemPoolResizableAllocatorMemoryObserver](#) ()

7.167.1 Constructor & Destructor Documentation

7.167.1.1 virtual [OsclMemPoolResizableAllocatorMemoryObserver::~OsclMemPoolResizableAllocatorMemoryObserver](#) () [inline, virtual]

7.167.2 Member Function Documentation

7.167.2.1 virtual void [OsclMemPoolResizableAllocatorMemoryObserver::freememoryavailable](#) ([OsclAny](#) * *aContextData*) [pure virtual]

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

- [oscl_mem_mempool.h](#)

7.168 OsciMemPoolResizableAllocatorObserver Class Reference

```
#include <osci_mem_mempool.h>
```

Public Methods

- virtual void [freeblockavailable](#) ([OsciAny](#) *aContextData)=0
- virtual [~OsciMemPoolResizableAllocatorObserver](#) ()

7.168.1 Constructor & Destructor Documentation

7.168.1.1 virtual [OsciMemPoolResizableAllocatorObserver::~OsciMemPoolResizableAllocatorObserver](#) () [[inline](#), [virtual](#)]

7.168.2 Member Function Documentation

7.168.2.1 virtual void [OsciMemPoolResizableAllocatorObserver::freeblockavailable](#) ([OsciAny](#) **aContextData*) [[pure virtual](#)]

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

- [osci_mem_mempool.h](#)

7.169 OslcMemStatsNode Class Reference

```
#include <oscl_mem_audit.h>
```

Public Methods

- [OslcMemStatsNode \(\)](#)
- void [reset \(\)](#)
- [~OslcMemStatsNode \(\)](#)
- void * [operator new \(oscl_memsize_t size\)](#)
- void * [operator new \(oscl_memsize_t size, OslcMemStatsNode *ptr\)](#)
- void [operator delete \(void *ptr\) throw \(\)](#)

Data Fields

- [MM_Stats_t * pMMStats](#)
- [MM_FailInsertParam * pMMFIPParam](#)
- char * [tag](#)

7.169.1 Constructor & Destructor Documentation

7.169.1.1 [OslcMemStatsNode::OslcMemStatsNode \(\)](#) [inline]

7.169.1.2 [OslcMemStatsNode::~~OslcMemStatsNode \(\)](#) [inline]

7.169.2 Member Function Documentation

7.169.2.1 void [OslcMemStatsNode::operator delete \(void * ptr\) throw \(\)](#) [inline]

7.169.2.2 void* [OslcMemStatsNode::operator new \(oscl_memsize_t size, OslcMemStatsNode * ptr\)](#) [inline]

7.169.2.3 void* [OslcMemStatsNode::operator new \(oscl_memsize_t size\)](#) [inline]

7.169.2.4 void [OslcMemStatsNode::reset \(\)](#) [inline]

7.169.3 Field Documentation

7.169.3.1 [MM_FailInsertParam*](#) [OslcMemStatsNode::pMMFIPParam](#)

7.169.3.2 [MM_Stats_t*](#) [OslcMemStatsNode::pMMStats](#)

7.169.3.3 char* [OslcMemStatsNode::tag](#)

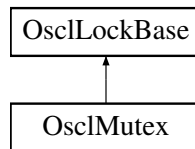
The documentation for this class was generated from the following file:

- [oscl_mem_audit.h](#)

7.170 OsciMutex Class Reference

```
#include <osci_mutex.h>
```

Inheritance diagram for OsciMutex::



Public Methods

- OSCL_IMPORT_REF [OsciMutex](#) ()
- virtual OSCL_IMPORT_REF [~OsciMutex](#) ()
- OSCL_IMPORT_REF [OsciProcStatus::eOsciProcError](#) [Create](#) (void)
- OSCL_IMPORT_REF void [Lock](#) ()
- OSCL_IMPORT_REF [OsciProcStatus::eOsciProcError](#) [TryLock](#) ()
- OSCL_IMPORT_REF void [Unlock](#) ()
- OSCL_IMPORT_REF [OsciProcStatus::eOsciProcError](#) [Close](#) (void)

7.170.1 Detailed Description

Class OsciMutex

7.170.2 Constructor & Destructor Documentation

7.170.2.1 OSCL_IMPORT_REF OsciMutex::OsciMutex ()

Class constructor

7.170.2.2 virtual OSCL_IMPORT_REF OsciMutex::~~OsciMutex () [virtual]

Class destructor

7.170.3 Member Function Documentation

7.170.3.1 OSCL_IMPORT_REF [OsciProcStatus::eOsciProcError](#) OsciMutex::Close (void)

Closes the Mutex

Parameters:

It wont take any prameters

Returns:

Returns the Error whether it is success or failure. Incase of failure it will return what is the specific error

7.170.3.2 OSCL_IMPORT_REF OsciProcStatus::eOsciProcError OsciMutex::Create (void)

Creates the Mutex

Parameters:

No input arguments

Returns:

Returns the Error whether it is success or failure. In case of failure it will return what is the specific error

7.170.3.3 OSCL_IMPORT_REF void OsciMutex::Lock () [virtual]

Locks the Mutex

Parameters:

It wont take any parameters

Returns:

Returns nothing

Implements [OsciLockBase](#).

7.170.3.4 OSCL_IMPORT_REF OsciProcStatus::eOsciProcError OsciMutex::TryLock ()

Try to lock the mutex, if the Mutex is already locked calling thread immediately returns without 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.

7.170.3.5 OSCL_IMPORT_REF void OsciMutex::Unlock () [virtual]

Releases the Mutex

Parameters:

It wont take any parameters

Returns:

Returns nothing

Implements [OsciLockBase](#).

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

- [oscl_mutex.h](#)

7.171 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

7.171.1 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.

7.171.2 Constructor & Destructor Documentation

7.171.2.1 `template<int __len> OsclNameString< __len >::OsclNameString () [inline]`

7.171.2.2 `template<int __len> OsclNameString< __len >::OsclNameString (const char a[]) [inline]`

7.171.2.3 `template<int __len> OsclNameString< __len >::OsclNameString (uint8 * a) [inline]`

7.171.3 Member Function Documentation

7.171.3.1 `template<int __len> int32 OsclNameString< __len >::MaxLen () const [inline]`

7.171.3.2 `template<int __len> void OsclNameString< __len >::Set (const char a[]) [inline]`

7.171.3.3 `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.

7.171.3.4 `template<int __len> uint8* OsclNameString< __len >::Str () const [inline]`

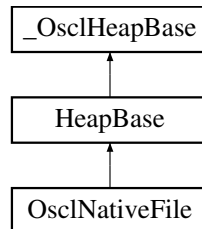
The documentation for this class was generated from the following file:

- [oscl_namestring.h](#)

7.172 OsciNativeFile Class Reference

```
#include <osci_file_native.h>
```

Inheritance diagram for OsciNativeFile::



Public Methods

- [OsciNativeFile \(\)](#)
- [~OsciNativeFile \(\)](#)
- [int32 Open \(const \[OsciFileHandle\]\(#\) &, uint32 mode, const \[OsciNativeFileParams\]\(#\) ¶ms, \[Osci_FileServer\]\(#\) &fileserv\)](#)
- [int32 Open \(const \[osci_wchar\]\(#\) *filename, uint32 mode, const \[OsciNativeFileParams\]\(#\) ¶ms, \[Osci_FileServer\]\(#\) &fileserv\)](#)
- [int32 Open \(const char *filename, uint32 mode, const \[OsciNativeFileParams\]\(#\) ¶ms, \[Osci_FileServer\]\(#\) &fileserv\)](#)
- [uint32 Read \(\[OsciAny\]\(#\) *buffer, uint32 size, uint32 numelements\)](#)
- [uint32 Write \(const \[OsciAny\]\(#\) *buffer, uint32 size, uint32 numelements\)](#)
- [int32 Seek \(\[TOsciFileOffset\]\(#\) offset, \[Osci_File::seek_type\]\(#\) origin\)](#)
- [TOsciFileOffset Tell \(\)](#)
- [int32 Flush \(\)](#)
- [int32 EndOfFile \(\)](#)
- [TOsciFileOffset Size \(\)](#)
- [int32 Close \(\)](#)
- [int32 SetSize \(uint32 size\)](#)
- [uint32 Mode \(\)](#)
- [int32 GetError \(\)](#)
- [int32 ReadAsync \(\[OsciAny\]\(#\) *buffer, uint32 size, uint32 numelements, \[OsciAOSStatus\]\(#\) &status\)](#)
- [uint32 GetReadAsyncNumElements \(\)](#)
- [bool HasAsyncRead \(\)](#)
- [void ReadAsyncCancel \(\)](#)

7.172.1 Constructor & Destructor Documentation

7.172.1.1 `OsciNativeFile::OsciNativeFile ()`

7.172.1.2 `OsciNativeFile::~~OsciNativeFile ()`

7.172.2 Member Function Documentation

7.172.2.1 `int32 OsciNativeFile::Close ()`

7.172.2.2 `int32 OsciNativeFile::EndOfFile ()`

7.172.2.3 `int32 OsciNativeFile::Flush ()`

7.172.2.4 `int32 OsciNativeFile::GetError ()`

7.172.2.5 `uint32 OsciNativeFile::GetReadAsyncNumElements ()`

Get the number of elements read in the last call to ReadAsync. @returns: number of elements read.

7.172.2.6 `bool OsciNativeFile::HasAsyncRead ()`

@returns: true if async read is supported natively.

7.172.2.7 `uint32 OsciNativeFile::Mode () [inline]`

7.172.2.8 `int32 OsciNativeFile::Open (const char * filename, uint32 mode, const OsciNativeFileParams & params, OsciFileServer & fileserv)`

7.172.2.9 `int32 OsciNativeFile::Open (const osci_wchar * filename, uint32 mode, const OsciNativeFileParams & params, OsciFileServer & fileserv)`

7.172.2.10 `int32 OsciNativeFile::Open (const OsciFileHandle &, uint32 mode, const OsciNativeFileParams & params, OsciFileServer & fileserv)`

7.172.2.11 `uint32 OsciNativeFile::Read (OsciAny * buffer, uint32 size, uint32 numelements)`

7.172.2.12 `int32 OsciNativeFile::ReadAsync (OsciAny * buffer, uint32 size, uint32 numelements, OsciAOSStatus & 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.

7.172.2.13 void OsciNativeFile::ReadAsyncCancel ()

Cancel any pending async read.

7.172.2.14 int32 OsciNativeFile::Seek (TOsciFileOffset *offset*, Osci_File::seek_type *origin*)**7.172.2.15 int32 OsciNativeFile::SetSize (uint32 *size*)****7.172.2.16 TOsciFileOffset OsciNativeFile::Size ()****7.172.2.17 TOsciFileOffset OsciNativeFile::Tell ()****7.172.2.18 uint32 OsciNativeFile::Write (const OsciAny * *buffer*, uint32 *size*, uint32 *numelements*)**

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

- [osci_file_native.h](#)

7.173 OsciNativeFileParams Class Reference

```
#include <osci_file_types.h>
```

Public Methods

- [OsciNativeFileParams](#) (uint32 mode=0, uint32 bufsize=0, uint32 asyncsize=0)

Data Fields

- uint32 [iNativeAccessMode](#)
- uint32 [iNativeBufferSize](#)
- uint32 [iAsyncReadBufferSize](#)

7.173.1 Constructor & Destructor Documentation

7.173.1.1 `OsciNativeFileParams::OsciNativeFileParams (uint32 mode = 0, uint32 bufsize = 0, uint32 asyncsize = 0) [inline]`

7.173.2 Field Documentation

7.173.2.1 `uint32 OsciNativeFileParams::iAsyncReadBufferSize`

7.173.2.2 `uint32 OsciNativeFileParams::iNativeAccessMode`

7.173.2.3 `uint32 OsciNativeFileParams::iNativeBufferSize`

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

- [osci_file_types.h](#)

7.174 OsciNetworkAddress Class Reference

```
#include <osci_socket_types.h>
```

Public Methods

- [OsciNetworkAddress](#) ()
- [OsciNetworkAddress](#) (const char *addr, int p)
- bool [operator==](#) (const OsciNetworkAddress &rhs) const

Data Fields

- [OsciNameString](#)< PVNETWORKADDRESS_LEN > [ipAddr](#)
- int [port](#)

7.174.1 Constructor & Destructor Documentation

7.174.1.1 [OsciNetworkAddress::OsciNetworkAddress](#) () [inline]

7.174.1.2 [OsciNetworkAddress::OsciNetworkAddress](#) (const char * *addr*, int *p*) [inline]

7.174.2 Member Function Documentation

7.174.2.1 bool [OsciNetworkAddress::operator==](#) (const OsciNetworkAddress & *rhs*) const [inline]

7.174.3 Field Documentation

7.174.3.1 [OsciNameString](#)<PVNETWORKADDRESS_LEN> [OsciNetworkAddress::ipAddr](#)

7.174.3.2 int [OsciNetworkAddress::port](#)

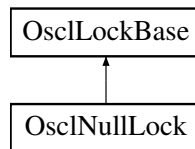
The documentation for this class was generated from the following file:

- [osci_socket_types.h](#)

7.175 OsclNullLock Class Reference

```
#include <oscl_lock_base.h>
```

Inheritance diagram for OsclNullLock::



Public Methods

- virtual void [Lock](#) ()
- virtual void [Unlock](#) ()
- virtual [~OsclNullLock](#) ()

7.175.1 Constructor & Destructor Documentation

7.175.1.1 virtual [OsclNullLock::~~OsclNullLock](#) () [inline, virtual]

7.175.2 Member Function Documentation

7.175.2.1 virtual void [OsclNullLock::Lock](#) () [inline, virtual]

Implements [OsclLockBase](#).

7.175.2.2 virtual void [OsclNullLock::Unlock](#) () [inline, virtual]

Implements [OsclLockBase](#).

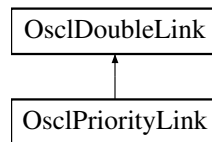
The documentation for this class was generated from the following file:

- [oscl_lock_base.h](#)

7.176 OslPriorityLink Class Reference

```
#include <osl_double_list.h>
```

Inheritance diagram for OslPriorityLink::



Data Fields

- `int32 iPriority`

7.176.1 Field Documentation

7.176.1.1 `int32 OslPriorityLink::iPriority`

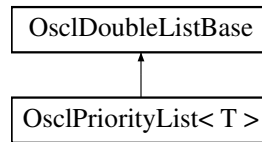
The documentation for this class was generated from the following file:

- `osl_double_list.h`

7.177 OslPriorityList< T > Class Template Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OslPriorityList< T >::



Public Methods

- OSCL_INLINE [OslPriorityList](#) ()
- OSCL_INLINE [OslPriorityList](#) (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 OslPriorityList< T >
```

7.177.1 Constructor & Destructor Documentation

7.177.1.1 `template<class T> OSCL_INLINE OslPriorityList< T >::OslPriorityList ()`

7.177.1.2 `template<class T> OSCL_INLINE OslPriorityList< T >::OslPriorityList (int32 anOffset)`

7.177.2 Member Function Documentation

7.177.2.1 `template<class T> OSCL_INLINE T* OslPriorityList< T >::Head ()`

7.177.2.2 `template<class T> OSCL_INLINE void OslPriorityList< T >::Insert (T &aRef)`

7.177.2.3 `template<class T> OSCL_INLINE bool OslPriorityList< T >::IsHead (const T * aPtr) const`

7.177.2.4 `template<class T> OSCL_INLINE bool OslPriorityList< T >::IsTail (const T * aPtr) const`

7.177.2.5 `template<class T> OSCL_INLINE T* OslPriorityList< T >::Tail ()`

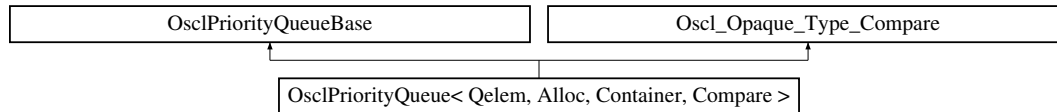
The documentation for this class was generated from the following file:

- [oscl_double_list.h](#)

7.178 OsciPriorityQueue< Qelem, Alloc, Container, Compare > Class Template Reference

```
#include <osci_priqueue.h>
```

Inheritance diagram for OsciPriorityQueue< 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)
- [OsciPriorityQueue](#) ()
- virtual [~OsciPriorityQueue](#) ()

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](#) ([OsciAny](#) *dest, const [OsciAny](#) *src)
- int [compare_LT](#) ([OsciAny](#) *a, [OsciAny](#) *b) const
- int [compare_EQ](#) (const [OsciAny](#) *a, const [OsciAny](#) *b) const

Protected Attributes

- Container [c](#)
- Compare [comp](#)

Friends

- class [oscl_priqueue_test](#)

```
template<class Qelem, class Alloc, class Container = Osci_Vector<Qelem, Alloc>, class Compare =
OsciCompareLess<Qelem>> class OsciPriorityQueue< Qelem, Alloc, Container, Compare >
```

7.178.1 Member Typedef Documentation

7.178.1.1 `template<class Qelem, class Alloc, class Container = Osci_Vector<Qelem, Alloc>, class Compare = OsciCompareLess<Qelem>> typedef Container::const_reference OsciPriorityQueue< Qelem, Alloc, Container, Compare >::const_reference`

7.178.1.2 `template<class Qelem, class Alloc, class Container = Osci_Vector<Qelem, Alloc>, class Compare = OsciCompareLess<Qelem>> typedef Container OsciPriorityQueue< Qelem, Alloc, Container, Compare >::container_type`

7.178.1.3 `template<class Qelem, class Alloc, class Container = Osci_Vector<Qelem, Alloc>, class Compare = OsciCompareLess<Qelem>> typedef Container::iterator OsciPriorityQueue< Qelem, Alloc, Container, Compare >::iterator`

7.178.1.4 `template<class Qelem, class Alloc, class Container = Osci_Vector<Qelem, Alloc>, class Compare = OsciCompareLess<Qelem>> typedef Container::value_type OsciPriorityQueue< Qelem, Alloc, Container, Compare >::value_type`

7.178.2 Constructor & Destructor Documentation

7.178.2.1 `template<class Qelem, class Alloc, class Container = Osci_Vector<Qelem, Alloc>, class Compare = OsciCompareLess<Qelem>> OsciPriorityQueue< Qelem, Alloc, Container, Compare >::OsciPriorityQueue () [inline]`

7.178.2.2 `template<class Qelem, class Alloc, class Container = Osci_Vector<Qelem, Alloc>, class Compare = OsciCompareLess<Qelem>> virtual OsciPriorityQueue< Qelem, Alloc, Container, Compare >::~~OsciPriorityQueue () [inline, virtual]`

7.178.3 Member Function Documentation

7.178.3.1 `template<class Qelem, class Alloc, class Container = Osci_Vector<Qelem, Alloc>, class Compare = OsciCompareLess<Qelem>> int OsciPriorityQueue< Qelem, Alloc, Container, Compare >::compare_EQ (const OsciAny * a, const OsciAny * b) const [inline, protected, virtual]`

Return a==b.

Implements [Osci_Opaque_Type_Compare](#).

7.178.3.2 `template<class Qelem, class Alloc, class Container = Osci_Vector<Qelem, Alloc>, class Compare = OsciCompareLess<Qelem>> int OsciPriorityQueue< Qelem, Alloc, Container, Compare >::compare_LT (OsciAny * a, OsciAny * b) const [inline, protected, virtual]`

Return a<b.

Implements [Oscl_Opaque_Type_Compare](#).

7.178.3.3 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> bool OsclPriorityQueue< Qelem, Alloc, Container, Compare >::empty () const [inline]`

7.178.3.4 `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]`

7.178.3.5 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> void OsclPriorityQueue< Qelem, Alloc, Container, Compare >::pop () [inline]`

7.178.3.6 `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]`

7.178.3.7 `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]`

7.178.3.8 `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]`

7.178.3.9 `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]`

7.178.3.10 `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]`

7.178.3.11 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> uint32 OsclPriorityQueue< Qelem, Alloc, Container, Compare >::size () const [inline]`

7.178.3.12 `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](#).

- 7.178.3.13 `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]
- 7.178.3.14 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> int OsclPriorityQueue< Qelem, Alloc, Container, Compare >::validate ()` [inline, protected]
- 7.178.3.15 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> const Container& OsclPriorityQueue< Qelem, Alloc, Container, Compare >::vec ()` [inline]

7.178.4 Friends And Related Function Documentation

- 7.178.4.1 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> friend class oscl_priqueue_test` [friend]

7.178.5 Field Documentation

- 7.178.5.1 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> Container OsclPriorityQueue< Qelem, Alloc, Container, Compare >::c` [protected]
- 7.178.5.2 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> Compare OsclPriorityQueue< Qelem, Alloc, Container, Compare >::comp` [protected]

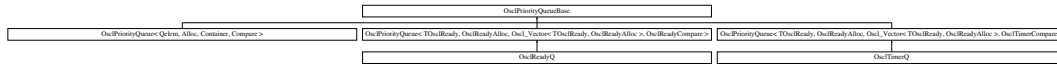
The documentation for this class was generated from the following file:

- [oscl_priqueue.h](#)

7.179 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`)

7.179.1 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`.

7.179.2 Constructor & Destructor Documentation

7.179.2.1 virtual `OsclPriorityQueueBase::~~OsclPriorityQueueBase()` [`inline`, `protected`, `virtual`]

7.179.3 Member Function Documentation

7.179.3.1 void `OsclPriorityQueueBase::construct` (`Oscl_Opaque_Type_Compare *ot`, `Oscl_Vector_Base *vec`) [`inline`, `protected`]

7.179.3.2 OSCL_IMPORT_REF `OsclAny*` `OsclPriorityQueueBase::find_heap` (`const OsclAny *input`, `OsclAny *first`, `OsclAny *last`) [`protected`]

7.179.3.3 OSCL_IMPORT_REF void `OsclPriorityQueueBase::pop_heap` (`OsclAny *first`, `OsclAny *last`) [`protected`]

7.179.3.4 OSCL_IMPORT_REF void `OsclPriorityQueueBase::push_heap` (`OsclAny *first`, `OsclAny *last`) [`protected`]

7.179.3.5 OSCL_IMPORT_REF int `OsclPriorityQueueBase::remove` (`const OsclAny *input`) [`protected`]

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

- [oscl_priqueue.h](#)

7.180 OsciProcStatus Class Reference

```
#include <osci_procstatus.h>
```

Public Types

- enum eOsciProcError { 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, OUTFOMEMORY_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_SIGNED_ERROR, PSHARED_ATTRIBUTE_SETTING_ERROR, NOT_IMPLEMENTED }

7.180.1 Detailed Description

Class OsciProcStatus

7.180.2 Member Enumeration Documentation

7.180.2.1 enum OsciProcStatus::eOsciProcError

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
OUTFOMEMORY_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

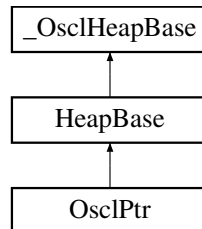
The documentation for this class was generated from the following file:

- [osci_procstatus.h](#)

7.181 OsciPtr Class Reference

```
#include <osci_file_async_read.h>
```

Inheritance diagram for OsciPtr::



Public Methods

- [OsciPtr](#) (uint8 *ptr, int32 &len, int32 max)
- [OsciPtr](#) (const OsciPtr &d)
- uint8 * [Ptr](#) ()
- void [SetLength](#) (int32 l)
- int32 [Length](#) ()
- void [Zero](#) ()
- void [Set](#) (OsciPtr &v)
- void [Set](#) (uint8 *ptr, int32 len, int32 max)
- void [Append](#) (OsciPtrC &v)

7.181.1 Constructor & Destructor Documentation

7.181.1.1 [OsciPtr::OsciPtr](#) (uint8 * ptr, int32 & len, int32 max) [inline]

7.181.1.2 [OsciPtr::OsciPtr](#) (const OsciPtr & d) [inline]

7.181.2 Member Function Documentation

7.181.2.1 void [OsciPtr::Append](#) ([OsciPtrC](#) & v) [inline]

7.181.2.2 int32 [OsciPtr::Length](#) () [inline]

7.181.2.3 uint8* [OsciPtr::Ptr](#) () [inline]

7.181.2.4 void [OsciPtr::Set](#) (uint8 * ptr, int32 len, int32 max) [inline]

7.181.2.5 void [OsciPtr::Set](#) (OsciPtr & v) [inline]

7.181.2.6 void [OsciPtr::SetLength](#) (int32 l) [inline]

7.181.2.7 void [OsciPtr::Zero](#) () [inline]

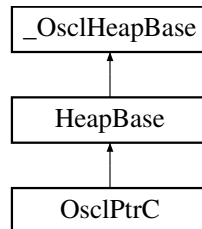
The documentation for this class was generated from the following file:

- [oscl_file_async_read.h](#)

7.182 OsciPtrC Class Reference

```
#include <osci_file_async_read.h>
```

Inheritance diagram for OsciPtrC::



Public Methods

- [OsciPtrC](#) (const uint8 *ptr, int32 len, int32 max)
- [OsciPtrC](#) (const OsciPtrC &d)
- const uint8 * [Ptr](#) ()
- void [SetLength](#) (int32 l)
- int32 [Length](#) ()
- void [Zero](#) ()
- void [Set](#) (OsciPtrC *v)
- void [Set](#) (uint8 *ptr, int32 len, int32 max)
- OsciPtrC [Right](#) (int32 size)
- OsciPtrC [Left](#) (int32 size)

7.182.1 Constructor & Destructor Documentation

7.182.1.1 `OslPtrC::OslPtrC (const uint8 * ptr, int32 len, int32 max)` [inline]

7.182.1.2 `OslPtrC::OslPtrC (const OslPtrC & d)` [inline]

7.182.2 Member Function Documentation

7.182.2.1 `OslPtrC OslPtrC::Left (int32 size)` [inline]

7.182.2.2 `int32 OslPtrC::Length ()` [inline]

7.182.2.3 `const uint8* OslPtrC::Ptr ()` [inline]

7.182.2.4 `OslPtrC OslPtrC::Right (int32 size)` [inline]

7.182.2.5 `void OslPtrC::Set (uint8 * ptr, int32 len, int32 max)` [inline]

7.182.2.6 `void OslPtrC::Set (OslPtrC * v)` [inline]

7.182.2.7 `void OslPtrC::SetLength (int32 l)` [inline]

7.182.2.8 `void OslPtrC::Zero ()` [inline]

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

- [oscl_file_async_read.h](#)

7.183 OslRand Class Reference

```
#include <oscl_rand.h>
```

Public Methods

- OSCL_COND_IMPORT_REF void [Seed](#) (int32 *seed*)
- OSCL_COND_IMPORT_REF int32 [Rand](#) ()

7.183.1 Member Function Documentation

7.183.1.1 OSCL_COND_IMPORT_REF int32 OslRand::Rand ()

7.183.1.2 OSCL_COND_IMPORT_REF void OslRand::Seed (int32 *seed*)

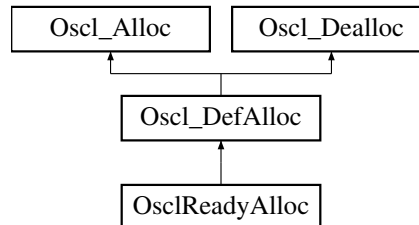
The documentation for this class was generated from the following file:

- [oscl_rand.h](#)

7.184 OsciReadyAlloc Class Reference

```
#include <osci_scheduler_readyq.h>
```

Inheritance diagram for OsciReadyAlloc::



Public Methods

- `OsciAny * allocate` (const uint32 size)
- `OsciAny * allocate_fl` (const uint32 size, const char *file_name, const int line_num)
- void `deallocate` (`OsciAny *p`)

7.184.1 Member Function Documentation

7.184.1.1 `OsciAny* OsciReadyAlloc::allocate` (const uint32 *size*) [virtual]

Implements `OsciDefAlloc`.

7.184.1.2 `OsciAny* OsciReadyAlloc::allocate_fl` (const uint32 *size*, const char **file_name*, const int *line_num*) [virtual]

Reimplemented from `OsciDefAlloc`.

7.184.1.3 void `OsciReadyAlloc::deallocate` (`OsciAny *p`) [virtual]

Implements `OsciDefAlloc`.

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

- [osci_scheduler_readyq.h](#)

7.185 OsciReadyCompare Class Reference

```
#include <osci_scheduler_readyq.h>
```

Static Public Methods

- `int compare (TOsciReady &a, TOsciReady &b)`

7.185.1 Member Function Documentation

7.185.1.1 `int OsciReadyCompare::compare (TOsciReady & a, TOsciReady & b)` [static]

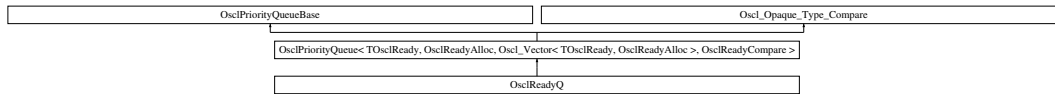
The documentation for this class was generated from the following file:

- [osci_scheduler_readyq.h](#)

7.186 OsciReadyQ Class Reference

```
#include <osci_scheduler_readyq.h>
```

Inheritance diagram for OsciReadyQ::



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](#) (PActiveBase *pvbase, int32 aReason)
- int32 [WaitForRequestComplete](#) (PActiveBase *)
- void [RegisterForCallback](#) (OsciSchedulerObserver *aCallback, OsciAny *aCallbackContext)
- void [TimerCallback](#) (uint32 aDelayMicrosec)
- OsciSchedulerObserver * [Callback](#) ()

7.186.1 Member Function Documentation

- 7.186.1.1 [OsclSchedulerObserver](#)* [OsclReadyQ::Callback](#) () [inline]
- 7.186.1.2 void [OsclReadyQ::Construct](#) (int)
- 7.186.1.3 uint32 [OsclReadyQ::Depth](#) () [inline]
- 7.186.1.4 bool [OsclReadyQ::IsIn](#) ([TOsclReady](#))
- 7.186.1.5 int32 [OsclReadyQ::PendComplete](#) ([PVAActiveBase](#) * *pvbase*, int32 *aReason*)
- 7.186.1.6 [TOsclReady](#) [OsclReadyQ::PopTop](#) ()
- 7.186.1.7 void [OsclReadyQ::RegisterForCallback](#) ([OsclSchedulerObserver](#) * *aCallback*, [OsclAny](#) * *aCallbackContext*)
- 7.186.1.8 void [OsclReadyQ::Remove](#) ([TOsclReady](#))
- 7.186.1.9 void [OsclReadyQ::ThreadLogoff](#) ()
- 7.186.1.10 void [OsclReadyQ::ThreadLogon](#) ()
- 7.186.1.11 void [OsclReadyQ::TimerCallback](#) (uint32 *aDelayMicrosec*)
- 7.186.1.12 [TOsclReady](#) [OsclReadyQ::Top](#) ()
- 7.186.1.13 [TOsclReady](#) [OsclReadyQ::WaitAndPopTop](#) (uint32)
- 7.186.1.14 [TOsclReady](#) [OsclReadyQ::WaitAndPopTop](#) ()
- 7.186.1.15 int32 [OsclReadyQ::WaitForRequestComplete](#) ([PVAActiveBase](#) *)

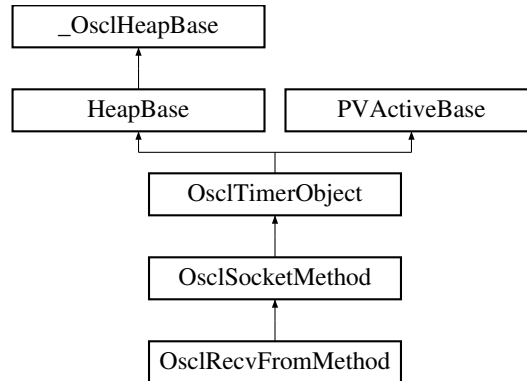
The documentation for this class was generated from the following file:

- [oscl_scheduler_readyq.h](#)

7.187 OsciRecvFromMethod Class Reference

```
#include <osci_socket_recv_from.h>
```

Inheritance diagram for OsciRecvFromMethod::



Public Methods

- `~OsciRecvFromMethod ()`
- `TPVSocketEvent RecvFrom (uint8 *&aPtr, uint32 aMaxLen, OsciNetworkAddress &aAddress, int32 aTimeout, uint32 aMultiMax, Osci_Vector< uint32, OsciMemAllocator > *aPacketLen, Osci_Vector< OsciNetworkAddress, OsciMemAllocator > *aPacketSource)`
- `uint8 * GetRecvData (int32 *aLength)`
- `OsciRecvFromRequest * RecvFromRequest ()`

Static Public Methods

- `OsciRecvFromMethod * NewL (OsciIPSocketI &c)`

7.187.1 Constructor & Destructor Documentation

7.187.1.1 `OsciRecvFromMethod::~~OsciRecvFromMethod ()`

7.187.2 Member Function Documentation

7.187.2.1 `uint8* OsciRecvFromMethod::GetRecvData (int32 *aLength)`

7.187.2.2 `OsciRecvFromMethod* OsciRecvFromMethod::NewL (OsciIPSocketI &c)`
[static]

7.187.2.3 `TPVSocketEvent OsciRecvFromMethod::RecvFrom (uint8 *&aPtr, uint32 aMaxLen, OsciNetworkAddress &aAddress, int32 aTimeout, uint32 aMultiMax, Osci_Vector< uint32, OsciMemAllocator > *aPacketLen, Osci_Vector< OsciNetworkAddress, OsciMemAllocator > *aPacketSource)`

7.187.2.4 `OsciRecvFromRequest* OsciRecvFromMethod::RecvFromRequest ()` [inline]

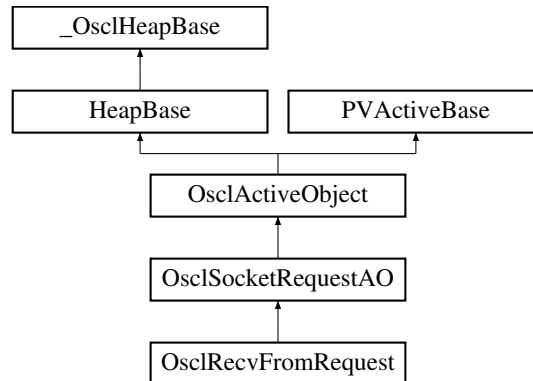
The documentation for this class was generated from the following file:

- [osci_socket_recv_from.h](#)

7.188 OsciRecvFromRequest Class Reference

```
#include <osci_socket_recv_from.h>
```

Inheritance diagram for OsciRecvFromRequest::



Public Methods

- uint8 * [GetRecvData](#) (int32 *aLength)
- [OsciRecvFromRequest](#) ([OsciSocketMethod](#) &c)
- void [RecvFrom](#) (uint8 *&aPtr, uint32 aMaxLen, [OsciNetworkAddress](#) &aAddress, uint32 aMultiMax, [Osci_Vector](#)< uint32, [OsciMemAllocator](#) > *aPacketLen, [Osci_Vector](#)< [OsciNetworkAddress](#), [OsciMemAllocator](#) > *aPacketSource)
- void [Success](#) ()

7.188.1 Detailed Description

This is the AO that interacts with the socket server

7.188.2 Constructor & Destructor Documentation

7.188.2.1 [OsciRecvFromRequest::OsciRecvFromRequest](#) ([OsciSocketMethod](#) &c) [inline]

7.188.3 Member Function Documentation

7.188.3.1 uint8* [OsciRecvFromRequest::GetRecvData](#) (int32 *aLength)

7.188.3.2 void [OsciRecvFromRequest::RecvFrom](#) (uint8 *&aPtr, uint32 aMaxLen, [OsciNetworkAddress](#) &aAddress, uint32 aMultiMax, [Osci_Vector](#)< uint32, [OsciMemAllocator](#) > *aPacketLen, [Osci_Vector](#)< [OsciNetworkAddress](#), [OsciMemAllocator](#) > *aPacketSource)

7.188.3.3 void [OsciRecvFromRequest::Success](#) () [virtual]

Reimplemented from [OsciSocketRequestAO](#).

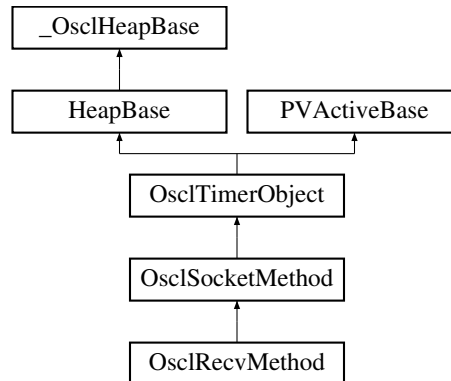
The documentation for this class was generated from the following file:

- [oscl_socket_rcv_from.h](#)

7.189 OsciRecvMethod Class Reference

```
#include <osci_socket_recv.h>
```

Inheritance diagram for OsciRecvMethod::



Public Methods

- [~OsciRecvMethod \(\)](#)
- [TPVSocketEvent Recv](#) (uint8 *&aPtr, uint32 aMaxLen, int32 aTimeout)
- [uint8 * GetRecvData](#) (int32 *aLength)
- [OsciRecvRequest * RecvRequest](#) ()

Static Public Methods

- [OsciRecvMethod * NewL](#) ([OsciIPSocketI](#) &c)

7.189.1 Constructor & Destructor Documentation

7.189.1.1 [OsciRecvMethod::~~OsciRecvMethod \(\)](#)

7.189.2 Member Function Documentation

7.189.2.1 [uint8* OsciRecvMethod::GetRecvData](#) (int32 * *aLength*)

7.189.2.2 [OsciRecvMethod* OsciRecvMethod::NewL](#) ([OsciIPSocketI](#) & *c*) [static]

7.189.2.3 [TPVSocketEvent OsciRecvMethod::Recv](#) (uint8 *& *aPtr*, uint32 *aMaxLen*, int32 *aTimeout*)

7.189.2.4 [OsciRecvRequest* OsciRecvMethod::RecvRequest](#) () [inline]

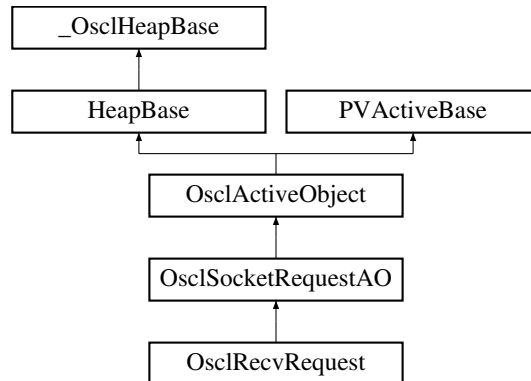
The documentation for this class was generated from the following file:

- [osci_socket_recv.h](#)

7.190 OsciRecvRequest Class Reference

```
#include <osci_socket_recv.h>
```

Inheritance diagram for OsciRecvRequest::



Public Methods

- `uint8 * GetRecvData (int32 *aLength)`
- `OsciRecvRequest (OsciSocketMethod &c)`
- `void Recv (uint8 *&aPtr, uint32 aMaxLen)`
- `void Success ()`

7.190.1 Detailed Description

This is the AO that interacts with the socket server

7.190.2 Constructor & Destructor Documentation

7.190.2.1 `OsciRecvRequest::OsciRecvRequest (OsciSocketMethod &c)` [inline]

7.190.3 Member Function Documentation

7.190.3.1 `uint8* OsciRecvRequest::GetRecvData (int32 * aLength)`

7.190.3.2 `void OsciRecvRequest::Recv (uint8 *& aPtr, uint32 aMaxLen)`

7.190.3.3 `void OsciRecvRequest::Success ()` [virtual]

Reimplemented from [OsciSocketRequestAO](#).

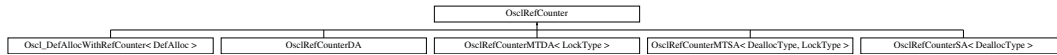
The documentation for this class was generated from the following file:

- [osci_socket_recv.h](#)

7.191 OsciRefCounter Class Reference

```
#include <osci_refcounter.h>
```

Inheritance diagram for OsciRefCounter::



Public Methods

- virtual void [addRef](#) ()=0
- virtual void [removeRef](#) ()=0
- virtual uint32 [getCount](#) ()=0
- virtual [~OsciRefCounter](#) ()

7.191.1 Detailed Description

Interface class for OsciRefCounter implementations

7.191.2 Constructor & Destructor Documentation

7.191.2.1 virtual OsciRefCounter::~~OsciRefCounter () [inline, virtual]

7.191.3 Member Function Documentation

7.191.3.1 virtual void OsciRefCounter::addRef () [pure virtual]

Add to the reference count

Implemented in [OsciRefCounterDA](#), [OsciRefCounterSA< DeallocType >](#), [OsciRefCounterMTDA< LockType >](#), [OsciRefCounterMTSA< DeallocType, LockType >](#), and [Osci_DefAllocWithRefCounter< DefAlloc >](#).

7.191.3.2 virtual uint32 OsciRefCounter::getCount () [pure virtual]

Gets the current number of references

Implemented in [OsciRefCounterDA](#), [OsciRefCounterSA< DeallocType >](#), [OsciRefCounterMTDA< LockType >](#), [OsciRefCounterMTSA< DeallocType, LockType >](#), and [Osci_DefAllocWithRefCounter< DefAlloc >](#).

7.191.3.3 virtual void OsciRefCounter::removeRef () [pure virtual]

Delete from reference count

Implemented in [OsciRefCounterDA](#), [OsciRefCounterSA< DeallocType >](#), [OsciRefCounterMTDA< LockType >](#), [OsciRefCounterMTSA< DeallocType, LockType >](#), and [Osci_DefAllocWithRefCounter< DefAlloc >](#).

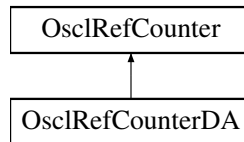
The documentation for this class was generated from the following file:

- [oscl_refcounter.h](#)

7.192 OsciRefCounterDA Class Reference

```
#include <osci_refcounter.h>
```

Inheritance diagram for OsciRefCounterDA::



Public Methods

- [OsciRefCounterDA](#) ([OsciAny](#) *p, [OsciDestructDealloc](#) *dealloc)
- virtual [~OsciRefCounterDA](#) ()
- void [addRef](#) ()
- void [removeRef](#) ()
- uint32 [getCount](#) ()

7.192.1 Detailed Description

Implementation of an [OsciRefCounter](#) that uses a dynamically created deallocator.

7.192.2 Constructor & Destructor Documentation

7.192.2.1 [OsciRefCounterDA::OsciRefCounterDA](#) ([OsciAny](#) * p, [OsciDestructDealloc](#) * *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 [OsciRefCounter](#) object (this) will self-destruct when the reference count is zero. In some cases the [OsciRefCounter](#) 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

7.192.2.2 virtual [OsciRefCounterDA::~~OsciRefCounterDA](#) () [inline, virtual]

Destructor empty

7.192.3 Member Function Documentation

7.192.3.1 void OsciRefCounterDA::addRef () [inline, virtual]

Add to the reference count

Implements [OsciRefCounter](#).

7.192.3.2 uint32 OsciRefCounterDA::getCount () [inline, virtual]

Gets the current number of references

Implements [OsciRefCounter](#).

7.192.3.3 void OsciRefCounterDA::removeRef () [inline, virtual]

Remove from the reference count

Implements [OsciRefCounter](#).

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

- [oscl_refcounter.h](#)

7.193 OsciRefCounterMemFrag Class Reference

```
#include <osci_refcounter_memfrag.h>
```

Public Methods

- [OsciRefCounterMemFrag](#) ([OsciMemoryFragment](#) &m, [OsciRefCounter](#) *r, uint32 in_capacity)
- [OsciRefCounterMemFrag](#) (const [OsciRefCounterMemFrag](#) &x)
- [OsciRefCounterMemFrag](#) ()
- [OsciRefCounterMemFrag](#) & [operator=](#) (const [OsciRefCounterMemFrag](#) &x)
- [~OsciRefCounterMemFrag](#) ()
- [OsciRefCounter](#) * [getRefCounter](#) ()
- [OsciMemoryFragment](#) & [getMemFrag](#) ()
- [OsciAny](#) * [getMemFragPtr](#) ()
- uint32 [getMemFragSize](#) ()
- uint32 [getCapacity](#) ()
- uint32 [getCount](#) ()

7.193.1 Detailed Description

Class to contain a memory fragment with it's associated reference counter.

7.193.2 Constructor & Destructor Documentation

7.193.2.1 [OsciRefCounterMemFrag::OsciRefCounterMemFrag](#) ([OsciMemoryFragment](#) & m, [OsciRefCounter](#) * 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.

7.193.2.2 [OsciRefCounterMemFrag::OsciRefCounterMemFrag](#) (const [OsciRefCounterMemFrag](#) & x) [inline]

Copy constructor.

7.193.2.3 [OsciRefCounterMemFrag::OsciRefCounterMemFrag](#) () [inline]

Default constructor.

7.193.2.4 [OsciRefCounterMemFrag::~~OsciRefCounterMemFrag](#) () [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.

7.193.3 Member Function Documentation

7.193.3.1 `uint32 OsciRefCounterMemFrag::getCapacity () [inline]`

Returns the capacity of the memory fragment

Returns:

7.193.3.2 `uint32 OsciRefCounterMemFrag::getCount () [inline]`

Returns the reference counter's current count.

7.193.3.3 `OsciMemoryFragment& OsciRefCounterMemFrag::getMemFrag () [inline]`

Returns a reference to the contained memory fragment structure.

7.193.3.4 `OsciAny* OsciRefCounterMemFrag::getMemFragPtr () [inline]`

Returns a pointer to the memory fragment data.

7.193.3.5 `uint32 OsciRefCounterMemFrag::getMemFragSize () [inline]`

Returns the size of the memory fragment data which equals its filled size.

Returns:

7.193.3.6 `OsciRefCounter* OsciRefCounterMemFrag::getRefCounter () [inline]`

Returns a pointer to the contained reference counter object

7.193.3.7 `OsciRefCounterMemFrag& OsciRefCounterMemFrag::operator= (const OsciRefCounterMemFrag & x) [inline]`

Assignment Operator

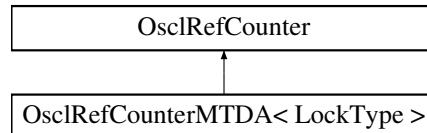
The documentation for this class was generated from the following file:

- [osci_refcounter_memfrag.h](#)

7.194 OsciRefCounterMTDA< LockType > Class Template Reference

```
#include <osci_refcounter.h>
```

Inheritance diagram for OsciRefCounterMTDA< LockType >::



Public Methods

- [OsciRefCounterMTDA](#) ([OsciAny](#) *p, [OsciDestructDealloc](#) *dealloc)
- virtual [~OsciRefCounterMTDA](#) ()
- void [addRef](#) ()
- void [removeRef](#) ()
- uint32 [getCount](#) ()

7.194.1 Detailed Description

template<class LockType> class OsciRefCounterMTDA< LockType >

Implementation of [OsciRefCounterDA](#) for multi-threaded use. A templated lock class must be specified.

7.194.2 Constructor & Destructor Documentation

7.194.2.1 template<class LockType> OsciRefCounterMTDA< LockType >::OsciRefCounterMTDA ([OsciAny](#) * p, [OsciDestructDealloc](#) * 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 [OsciRefCounter](#) object (this) will self-destruct when the reference count is zero. In some cases the [OsciRefCounter](#) 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

7.194.2.2 template<class LockType> virtual OsciRefCounterMTDA< LockType >::~~OsciRefCounterMTDA () [inline, virtual]

Destructor empty

7.194.3 Member Function Documentation

7.194.3.1 `template<class LockType> void OsciRefCounterMTDA< LockType >::addRef ()`
[inline, virtual]

Add to the reference count

Implements [OsciRefCounter](#).

7.194.3.2 `template<class LockType> uint32 OsciRefCounterMTDA< LockType >::getCount ()`
[inline, virtual]

Gets the current number of references

Implements [OsciRefCounter](#).

7.194.3.3 `template<class LockType> void OsciRefCounterMTDA< LockType >::removeRef ()`
[inline, virtual]

Remove from the reference count

Implements [OsciRefCounter](#).

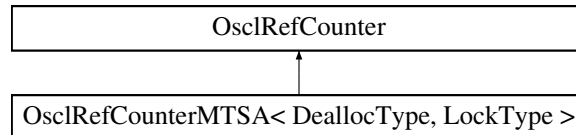
The documentation for this class was generated from the following file:

- [osci_refcounter.h](#)

7.195 OsciRefCounterMTSA< DeallocType, LockType > Class Template Reference

```
#include <osci_refcounter.h>
```

Inheritance diagram for OsciRefCounterMTSA< DeallocType, LockType >::



Public Methods

- [OsciRefCounterMTSA \(OsciAny *p\)](#)
- virtual [~OsciRefCounterMTSA \(\)](#)
- void [addRef \(\)](#)
- void [removeRef \(\)](#)
- uint32 [getCount \(\)](#)

7.195.1 Detailed Description

```
template<class DeallocType, class LockType> class OsciRefCounterMTSA< DeallocType, LockType >
```

Implementation of [OsciRefCounterSA](#) for multi-threaded use. A templated lock class must be specified.

7.195.2 Constructor & Destructor Documentation

```
7.195.2.1 template<class DeallocType, class LockType> OsciRefCounterMTSA< DeallocType, LockType >::OsciRefCounterMTSA (OsciAny *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 [OsciRefCounter](#) object (this) will self-destruct when the reference count is zero. In some cases the [OsciRefCounter](#) 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

```
7.195.2.2 template<class DeallocType, class LockType> virtual OsciRefCounterMTSA< DeallocType, LockType >::~~OsciRefCounterMTSA () [inline, virtual]
```

Destructor empty

7.195.3 Member Function Documentation

7.195.3.1 `template<class DeallocType, class LockType> void OsciRefCounterMTSA< DeallocType, LockType >::addRef () [inline, virtual]`

Add to the reference count

Implements [OsciRefCounter](#).

7.195.3.2 `template<class DeallocType, class LockType> uint32 OsciRefCounterMTSA< DeallocType, LockType >::getCount () [inline, virtual]`

Gets the current number of references

Implements [OsciRefCounter](#).

7.195.3.3 `template<class DeallocType, class LockType> void OsciRefCounterMTSA< DeallocType, LockType >::removeRef () [inline, virtual]`

Remove from the reference count

Implements [OsciRefCounter](#).

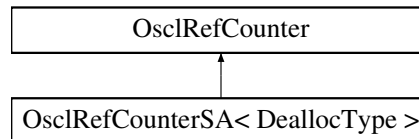
The documentation for this class was generated from the following file:

- [oscl_refcounter.h](#)

7.196 OsciRefCounterSA< DeallocType > Class Template Reference

```
#include <osci_refcounter.h>
```

Inheritance diagram for OsciRefCounterSA< DeallocType >::



Public Methods

- [OsciRefCounterSA](#) ([OsciAny](#) *p)
- virtual [~OsciRefCounterSA](#) ()
- void [addRef](#) ()
- void [removeRef](#) ()
- uint32 [getCount](#) ()

7.196.1 Detailed Description

template<class DeallocType> class OsciRefCounterSA< DeallocType >

Implementation of an [OsciRefCounter](#) that uses a statically created deallocator.

7.196.2 Constructor & Destructor Documentation

7.196.2.1 **template<class DeallocType> OsciRefCounterSA< DeallocType >::OsciRefCounterSA**
([OsciAny](#) *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 [OsciRefCounter](#) object (this) will self-destruct when the reference count is zero. In some cases the [OsciRefCounter](#) 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

7.196.2.2 **template<class DeallocType> virtual OsciRefCounterSA< DeallocType**
>::~~OsciRefCounterSA () [inline, virtual]

Destructor empty

7.196.3 Member Function Documentation

7.196.3.1 `template<class DeallocType> void OsciRefCounterSA< DeallocType >::addRef ()`
[inline, virtual]

Add to the reference count

Implements [OsciRefCounter](#).

7.196.3.2 `template<class DeallocType> uint32 OsciRefCounterSA< DeallocType >::getCount ()`
[inline, virtual]

Gets the current number of references

Implements [OsciRefCounter](#).

7.196.3.3 `template<class DeallocType> void OsciRefCounterSA< DeallocType >::removeRef ()`
[inline, virtual]

Remove from the reference count

Implements [OsciRefCounter](#).

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

- [osci_refcounter.h](#)

7.197 OslRegistryAccessClient Class Reference

```
#include <osl_registry_access_client.h>
```

Public Methods

- OSCL_IMPORT_REF [OslRegistryAccessClient](#) ()
- OSCL_IMPORT_REF [~OslRegistryAccessClient](#) ()
- OSCL_IMPORT_REF int32 [Connect](#) ()
- OSCL_IMPORT_REF [OslComponentFactory](#) [GetFactory](#) ([OSCL_String](#) &aComponent)
- OSCL_IMPORT_REF void [GetFactories](#) ([OSCL_String](#) &aRegistry, [Osl_Vector](#)< [OslRegistryAccessElement](#), [OslMemAllocator](#) > &aVec)
- OSCL_IMPORT_REF void [Close](#) ()

7.197.1 Constructor & Destructor Documentation

7.197.1.1 OSCL_IMPORT_REF [OslRegistryAccessClient::OslRegistryAccessClient](#) ()

7.197.1.2 OSCL_IMPORT_REF [OslRegistryAccessClient::~~OslRegistryAccessClient](#) ()

7.197.2 Member Function Documentation

7.197.2.1 OSCL_IMPORT_REF void [OslRegistryAccessClient::Close](#) ()

Close and cleanup session.

7.197.2.2 OSCL_IMPORT_REF int32 [OslRegistryAccessClient::Connect](#) ()

Create a session.

Returns:

OslErrNone on success.

7.197.2.3 OSCL_IMPORT_REF void [OslRegistryAccessClient::GetFactories](#) ([OSCL_String](#) &*aRegistry*, [Osl_Vector](#)< [OslRegistryAccessElement](#), [OslMemAllocator](#) > & *aVec*)

Get all factories for a given registry type.

Parameters:

aRegistry: registry MIME type

aVec: output component factory + mimestring vector.

7.197.2.4 OSCL_IMPORT_REF [OslComponentFactory](#) [OslRegistryAccessClient::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.

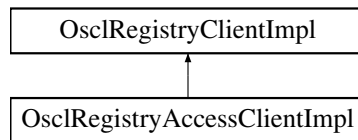
The documentation for this class was generated from the following file:

- [oscl_registry_access_client.h](#)

7.198 OsciRegistryAccessClientImpl Class Reference

```
#include <osci_registry_client_impl.h>
```

Inheritance diagram for OsciRegistryAccessClientImpl::



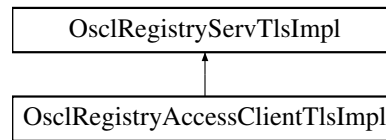
The documentation for this class was generated from the following file:

- [osci_registry_client_impl.h](#)

7.199 OsciRegistryAccessClientTlsImpl Class Reference

```
#include <osci_registry_client_impl.h>
```

Inheritance diagram for OsciRegistryAccessClientTlsImpl::



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

- [osci_registry_client_impl.h](#)

7.200 OslRegistryAccessElement Class Reference

```
#include <oscl_registry_types.h>
```

Data Fields

- [OslComponentFactory](#) iFactory
- [OSCL_HeapString< OslMemAllocator >](#) iMimeString

7.200.1 Detailed Description

A class used to access the registry data

7.200.2 Field Documentation

7.200.2.1 [OslComponentFactory](#) OslRegistryAccessElement::iFactory

7.200.2.2 [OSCL_HeapString<OslMemAllocator>](#) OslRegistryAccessElement::iMimeString

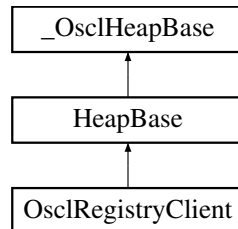
The documentation for this class was generated from the following file:

- [oscl_registry_types.h](#)

7.201 OslRegistryClient Class Reference

```
#include <oscl_registry_client.h>
```

Inheritance diagram for OslRegistryClient::



Public Methods

- OSCL_IMPORT_REF [OslRegistryClient \(\)](#)
- OSCL_IMPORT_REF [~OslRegistryClient \(\)](#)
- OSCL_IMPORT_REF int32 [Connect](#) (bool aPerThread=false)
- OSCL_IMPORT_REF int32 [Register](#) (OSCL_String &aComponentID, [OslComponentFactory](#) aFactory)
- OSCL_IMPORT_REF int32 [UnRegister](#) (OSCL_String &aComponentID)
- OSCL_IMPORT_REF void [Close](#) ()

7.201.1 Constructor & Destructor Documentation

7.201.1.1 OSCL_IMPORT_REF [OslRegistryClient::OslRegistryClient \(\)](#)

7.201.1.2 OSCL_IMPORT_REF [OslRegistryClient::~~OslRegistryClient \(\)](#)

7.201.2 Member Function Documentation

7.201.2.1 OSCL_IMPORT_REF void [OslRegistryClient::Close \(\)](#)

Close and cleanup. All components registered in this session are automatically unregistered.

7.201.2.2 OSCL_IMPORT_REF int32 [OslRegistryClient::Connect \(bool aPerThread = false\)](#)

Create a session.

Parameters:

aPerThread: Select per-thread registry instead of global registry.

Returns:

OslErrNone on success.

7.201.2.3 OSCL_IMPORT_REF int32 OsciRegistryClient::Register (OSCL_String & *aComponentID*, OsciComponentFactory *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:

OsciErrNone on success.

7.201.2.4 OSCL_IMPORT_REF int32 OsciRegistryClient::UnRegister (OSCL_String & *aComponentID*)

Unregister a previously registered component.

Returns:

OsciErrNone on success.

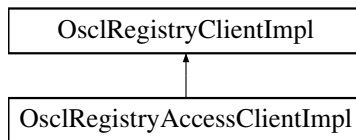
The documentation for this class was generated from the following file:

- [oscl_registry_client.h](#)

7.202 OsciRegistryClientImpl Class Reference

```
#include <osci_registry_client_impl.h>
```

Inheritance diagram for OsciRegistryClientImpl::



Protected Methods

- int32 [Connect](#) ()
- void [Close](#) ()
- int32 [Register](#) (OSCL_String &, OsciComponentFactory)
- int32 [UnRegister](#) (OSCL_String &)
- OsciComponentFactory [GetFactory](#) (OSCL_String &)
- void [GetFactories](#) (OSCL_String &, Osci_Vector< OsciRegistryAccessElement, OsciMemAllocator > &)

Friends

- class [OsciRegistryClient](#)
- class [OsciRegistryAccessClient](#)

7.202.1 Member Function Documentation

7.202.1.1 `void OslRegistryClientImpl::Close (void)` [inline, protected]

7.202.1.2 `int32 OslRegistryClientImpl::Connect ()` [inline, protected]

7.202.1.3 `void OslRegistryClientImpl::GetFactories (OSCL_String &, Osl_Vector< OslRegistryAccessElement, OslMemAllocator > &)` [inline, protected]

7.202.1.4 `OslComponentFactory OslRegistryClientImpl::GetFactory (OSCL_String &)`
[inline, protected]

7.202.1.5 `int32 OslRegistryClientImpl::Register (OSCL_String &, OslComponentFactory)`
[inline, protected]

7.202.1.6 `int32 OslRegistryClientImpl::UnRegister (OSCL_String &)` [inline, protected]

7.202.2 Friends And Related Function Documentation

7.202.2.1 `friend class OslRegistryAccessClient` [friend]

7.202.2.2 `friend class OslRegistryClient` [friend]

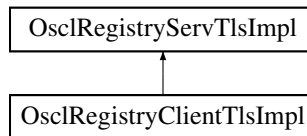
The documentation for this class was generated from the following file:

- [oscl_registry_client_impl.h](#)

7.203 OsciRegistryClientTlsImpl Class Reference

```
#include <osci_registry_client_impl.h>
```

Inheritance diagram for OsciRegistryClientTlsImpl::



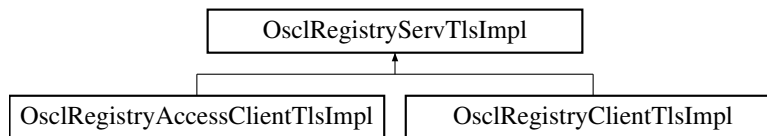
The documentation for this class was generated from the following file:

- [osci_registry_client_impl.h](#)

7.204 OslRegistryServTlsImpl Class Reference

```
#include <osl_registry_serv_impl_tls.h>
```

Inheritance diagram for OslRegistryServTlsImpl::



Protected Methods

- [OslRegistryServTlsImpl \(\)](#)
- [virtual ~OslRegistryServTlsImpl \(\)](#)
- [int32 Connect \(\)](#)
- [void Close \(\)](#)
- [int32 Register \(OSCL_String &aComponentID, OslComponentFactory aFactory\)](#)
- [int32 UnRegister \(OSCL_String &aComponentID\)](#)
- [OslComponentFactory GetFactory \(OSCL_String &aComponent\)](#)
- [void GetFactories \(OSCL_String &aRegistry, Osl_Vector< OslRegistryAccessElement, Osl-MemAllocator > &aVec\)](#)

Friends

- class [OslRegistryClient](#)
- class [OslRegistryAccessClient](#)

7.204.1 Constructor & Destructor Documentation

- 7.204.1.1 `OslRegistryServTlsImpl::OslRegistryServTlsImpl ()` [protected]
- 7.204.1.2 `virtual OslRegistryServTlsImpl::~~OslRegistryServTlsImpl ()` [protected, virtual]

7.204.2 Member Function Documentation

- 7.204.2.1 `void OslRegistryServTlsImpl::Close ()` [protected]
- 7.204.2.2 `int32 OslRegistryServTlsImpl::Connect ()` [protected]
- 7.204.2.3 `void OslRegistryServTlsImpl::GetFactories (OSCL_String & aRegistry, Osl_Vector< OslRegistryAccessElement, OslMemAllocator > & aVec)` [protected]
- 7.204.2.4 `OslComponentFactory OslRegistryServTlsImpl::GetFactory (OSCL_String & aComponent)` [protected]
- 7.204.2.5 `int32 OslRegistryServTlsImpl::Register (OSCL_String & aComponentID, OslComponentFactory aFactory)` [protected]
- 7.204.2.6 `int32 OslRegistryServTlsImpl::UnRegister (OSCL_String & aComponentID)` [protected]

7.204.3 Friends And Related Function Documentation

- 7.204.3.1 `friend class OslRegistryAccessClient` [friend]
- 7.204.3.2 `friend class OslRegistryClient` [friend]

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

- [oscl_registry_serv_impl_tls.h](#)

7.205 OsciScheduler Class Reference

```
#include <osci_scheduler.h>
```

Static Public Methods

- OSCL_IMPORT_REF void [Init](#) (const char *name, [Osci_DefAlloc](#) *alloc=NULL, int nreserve=20)
- OSCL_IMPORT_REF void [Cleanup](#) ()

7.205.1 Detailed Description

Per-thread scheduler initialization and cleanup.

7.205.2 Member Function Documentation

7.205.2.1 OSCL_IMPORT_REF void OsciScheduler::Cleanup () [static]

This routine uninstalls and destroys Osci scheduler for the calling thread.

7.205.2.2 OSCL_IMPORT_REF void OsciScheduler::Init (const char * name, [Osci_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:

- [osci_scheduler.h](#)

7.206 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](#) ()

7.206.1 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.

7.206.2 Constructor & Destructor Documentation

7.206.2.1 virtual [OsclSchedulerObserver::~OsclSchedulerObserver](#) () [inline, virtual]

7.206.3 Member Function Documentation

7.206.3.1 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.

7.206.3.2 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](#)

7.207 OsciScopedLock< LockClass > Class Template Reference

The OsciScopedLock 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 OsciScopedLock goes out of scope.

```
#include <osci_lock_base.h>
```

Public Methods

- [OsciScopedLock](#) (LockClass &inLock)
Default constructor Initializes the pointer and takes ownership.
- [~OsciScopedLock](#) ()
Destructor.

7.207.1 Detailed Description

```
template<class LockClass> class OsciScopedLock< LockClass >
```

The OsciScopedLock 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 OsciScopedLock 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.

7.207.2 Constructor & Destructor Documentation

7.207.2.1 `template<class LockClass> OsciScopedLock< LockClass >::OsciScopedLock (LockClass & inLock) [inline, explicit]`

Default constructor Initializes the pointer and takes ownership.

7.207.2.2 `template<class LockClass> OsciScopedLock< LockClass >::~~OsciScopedLock () [inline]`

Destructor.

The pointer is deleted in case this class still has ownership

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

- [osci_lock_base.h](#)

7.208 OslSelect Class Reference

```
#include <oscl_init.h>
```

Public Methods

- [OslSelect \(\)](#)
- [OslSelect \(Osl_DefAlloc *erralloc, Osl_DefAlloc *schedalloc, const char *name, int32 reserve=10, bool heapcheck=false, FILE *output=NULL\)](#)

Data Fields

- bool [iOslBase](#)
- bool [iOslMemory](#)
- bool [iOslErrorTrap](#)
- bool [iOslLogger](#)
- bool [iOslScheduler](#)
- [Osl_DefAlloc](#) * [iErrAlloc](#)
- [Osl_DefAlloc](#) * [iSchedulerAlloc](#)
- const char * [iSchedulerName](#)
- int32 [iSchedulerReserve](#)
- bool [iHeapCheck](#)
- FILE * [iOutputFile](#)

7.208.1 Detailed Description

Osl Module selection and Init/Cleanup options.

7.208.2 Constructor & Destructor Documentation

7.208.2.1 `OsciSelect::OsciSelect ()` [inline]

7.208.2.2 `OsciSelect::OsciSelect (Osci_DefAlloc * erralloc, Osci_DefAlloc * schedalloc, const char * name, int32 reserve = 10, bool heapcheck = false, FILE * output = NULL)` [inline]

7.208.3 Field Documentation

7.208.3.1 `Osci_DefAlloc*` `OsciSelect::iErrAlloc`

7.208.3.2 `bool` `OsciSelect::iHeapCheck`

7.208.3.3 `bool` `OsciSelect::iOsciBase`

7.208.3.4 `bool` `OsciSelect::iOsciErrorTrap`

7.208.3.5 `bool` `OsciSelect::iOsciLogger`

7.208.3.6 `bool` `OsciSelect::iOsciMemory`

7.208.3.7 `bool` `OsciSelect::iOsciScheduler`

7.208.3.8 `FILE*` `OsciSelect::iOutputFile`

7.208.3.9 `Osci_DefAlloc*` `OsciSelect::iSchedulerAlloc`

7.208.3.10 `const char*` `OsciSelect::iSchedulerName`

7.208.3.11 `int32` `OsciSelect::iSchedulerReserve`

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

- [osci_init.h](#)

7.209 OsciSemaphore Class Reference

```
#include <osci_semaphore.h>
```

Public Methods

- OSCL_IMPORT_REF [OsciSemaphore \(\)](#)
- OSCL_IMPORT_REF [~OsciSemaphore \(\)](#)
- OSCL_IMPORT_REF [OsciProcStatus::eOsciProcError Create \(uint32 initVal=0\)](#)
- OSCL_IMPORT_REF [OsciProcStatus::eOsciProcError Close \(\)](#)
- OSCL_IMPORT_REF [OsciProcStatus::eOsciProcError Wait \(\)](#)
- OSCL_IMPORT_REF [OsciProcStatus::eOsciProcError Wait \(uint32 timeout_msec\)](#)
- OSCL_IMPORT_REF [OsciProcStatus::eOsciProcError TryWait \(\)](#)
- OSCL_IMPORT_REF [OsciProcStatus::eOsciProcError Signal \(\)](#)

7.209.1 Detailed Description

Class Semaphore

7.209.2 Constructor & Destructor Documentation

7.209.2.1 OSCL_IMPORT_REF OsciSemaphore::OsciSemaphore ()

Class constructor

7.209.2.2 OSCL_IMPORT_REF OsciSemaphore::~~OsciSemaphore ()

Class destructor

7.209.3 Member Function Documentation

7.209.3.1 OSCL_IMPORT_REF [OsciProcStatus::eOsciProcError](#) OsciSemaphore::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

7.209.3.2 OSCL_IMPORT_REF [OsciProcStatus::eOsciProcError](#) OsciSemaphore::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

7.209.3.3 OSCL_IMPORT_REF OsciProcStatus::eOsciProcError OsciSemaphore::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

7.209.3.4 OSCL_IMPORT_REF OsciProcStatus::eOsciProcError OsciSemaphore::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.

7.209.3.5 OSCL_IMPORT_REF OsciProcStatus::eOsciProcError OsciSemaphore::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.

7.209.3.6 OSCL_IMPORT_REF OsciProcStatus::eOsciProcError OsciSemaphore::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

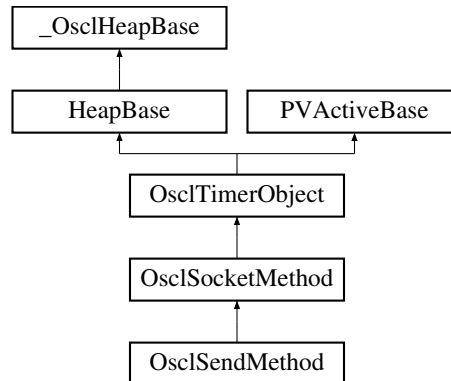
The documentation for this class was generated from the following file:

- [oscl_semaphore.h](#)

7.210 OsciSendMethod Class Reference

```
#include <osci_socket_send.h>
```

Inheritance diagram for OsciSendMethod::



Public Methods

- [~OsciSendMethod \(\)](#)
- [TPVSocketEvent Send](#) (const uint8 *&aPtr, uint32 aLen, int32 aTimeout)
- [uint8 * GetSendData](#) (int32 *aLength)
- [OsciSendRequest * SendRequest](#) ()

Static Public Methods

- [OsciSendMethod * NewL](#) ([OsciIPSocketI](#) &c)

7.210.1 Constructor & Destructor Documentation

7.210.1.1 [OsciSendMethod::~~OsciSendMethod \(\)](#)

7.210.2 Member Function Documentation

7.210.2.1 [uint8* OsciSendMethod::GetSendData](#) (int32 * *aLength*)

7.210.2.2 [OsciSendMethod* OsciSendMethod::NewL](#) ([OsciIPSocketI](#) & *c*) [static]

7.210.2.3 [TPVSocketEvent OsciSendMethod::Send](#) (const uint8 *& *aPtr*, uint32 *aLen*, int32 *aTimeout*)

7.210.2.4 [OsciSendRequest* OsciSendMethod::SendRequest](#) () [inline]

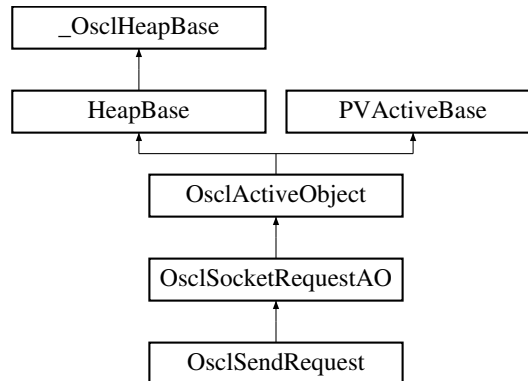
The documentation for this class was generated from the following file:

- [osci_socket_send.h](#)

7.211 OslSendRequest Class Reference

```
#include <oscl_socket_send.h>
```

Inheritance diagram for OslSendRequest::



Public Methods

- [OslSendRequest](#) ([OslSocketMethod](#) &c)
- void [Send](#) (const uint8 *&aPtr, uint32 aLen)
- void [Success](#) ()
- uint8 * [GetSendData](#) (int32 *aLength)

7.211.1 Constructor & Destructor Documentation

7.211.1.1 [OslSendRequest::OslSendRequest](#) ([OslSocketMethod](#) & c) [inline]

7.211.2 Member Function Documentation

7.211.2.1 uint8* [OslSendRequest::GetSendData](#) (int32 * aLength)

7.211.2.2 void [OslSendRequest::Send](#) (const uint8 *& aPtr, uint32 aLen)

7.211.2.3 void [OslSendRequest::Success](#) () [virtual]

Reimplemented from [OslSocketRequestAO](#).

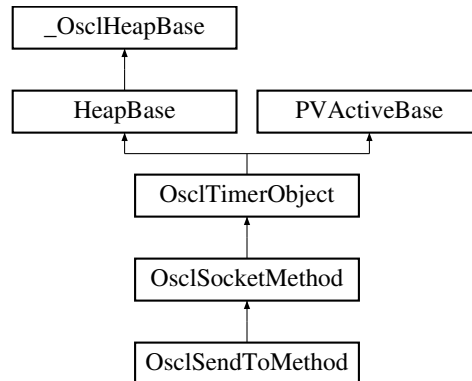
The documentation for this class was generated from the following file:

- [oscl_socket_send.h](#)

7.212 OsciSendToMethod Class Reference

```
#include <osci_socket_send_to.h>
```

Inheritance diagram for OsciSendToMethod::



Public Methods

- [~OsciSendToMethod \(\)](#)
- [TPVSocketEvent SendTo](#) (const uint8 *&aPtr, uint32 aLen, [OsciNetworkAddress](#) &aAddress, int32 aTimeout)
- uint8 * [GetSendData](#) (int32 *aLength)
- [OsciSendToRequest](#) * [SendToRequest](#) ()

Static Public Methods

- [OsciSendToMethod](#) * [NewL](#) ([OsciIPSocketI](#) &c)

7.212.1 Constructor & Destructor Documentation

7.212.1.1 [OsciSendToMethod::~OsciSendToMethod \(\)](#)

7.212.2 Member Function Documentation

7.212.2.1 uint8* [OsciSendToMethod::GetSendData](#) (int32 * *aLength*)

7.212.2.2 [OsciSendToMethod](#)* [OsciSendToMethod::NewL](#) ([OsciIPSocketI](#) & *c*) [static]

7.212.2.3 [TPVSocketEvent](#) [OsciSendToMethod::SendTo](#) (const uint8 *& *aPtr*, uint32 *aLen*, [OsciNetworkAddress](#) & *aAddress*, int32 *aTimeout*)

7.212.2.4 [OsciSendToRequest](#)* [OsciSendToMethod::SendToRequest](#) () [inline]

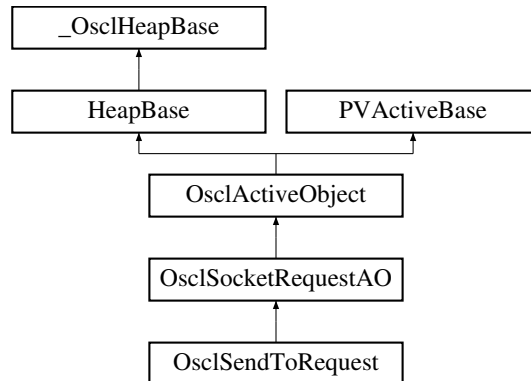
The documentation for this class was generated from the following file:

- [osci_socket_send_to.h](#)

7.213 OsciSendToRequest Class Reference

```
#include <osci_socket_send_to.h>
```

Inheritance diagram for OsciSendToRequest::



Public Methods

- [OsciSendToRequest](#) ([OsciSocketMethod](#) &c)
- void [SendTo](#) (const uint8 *&aPtr, uint32 aLen, [OsciNetworkAddress](#) &aAddress)
- void [Success](#) ()
- uint8 * [GetSendData](#) (int32 *aLength)

7.213.1 Detailed Description

This is the AO that interacts with the socket server

7.213.2 Constructor & Destructor Documentation

7.213.2.1 [OsciSendToRequest::OsciSendToRequest](#) ([OsciSocketMethod](#) &c) [inline]

7.213.3 Member Function Documentation

7.213.3.1 uint8* [OsciSendToRequest::GetSendData](#) (int32 *aLength)

7.213.3.2 void [OsciSendToRequest::SendTo](#) (const uint8 *&aPtr, uint32 aLen, [OsciNetworkAddress](#) &aAddress)

7.213.3.3 void [OsciSendToRequest::Success](#) () [virtual]

Reimplemented from [OsciSocketRequestAO](#).

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

- [osci_socket_send_to.h](#)

7.214 OsciSharedPtr< TheClass > Class Template Reference

A parameterized smart pointer class.

```
#include <osci_shared_ptr.h>
```

Public Methods

- [OsciSharedPtr](#) ()
Constructor.
- [OsciSharedPtr](#) (TheClass *inClassPtr, [OsciRefCount](#) *in_refcnt)
Constructor.
- [OsciSharedPtr](#) (const OsciSharedPtr &inSharedPtr)
Copy constructor.
- virtual [~OsciSharedPtr](#) ()
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.
- [OsciRefCount](#) * [GetRefCount](#) ()
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 OsciSharedPtr &inHandle)
Use this function to bind an existing OsciSharedPtr to a already-wrapped object.
- void [Bind](#) (TheClass *ptr, [OsciRefCount](#) *in_refcnt)
Use this function to bind an existing OsciSharedPtr to a new (unwrapped) object.
- void [Unbind](#) ()
Use this function of unbind an existing OsciSharedPtr.
- OsciSharedPtr & [operator=](#) (const OsciSharedPtr &inSharedPtr)
Assignment operator.
- bool [operator==](#) (const OsciSharedPtr &b) const
Test for equality to see if two PVHandles wrap the same object.

7.214.1 Detailed Description

template<class TheClass> class OsciSharedPtr< TheClass >

A parameterized smart pointer class.

7.214.2 Constructor & Destructor Documentation

7.214.2.1 template<class TheClass> OsciSharedPtr< TheClass >::OsciSharedPtr () [inline]

Constructor.

7.214.2.2 template<class TheClass> OsciSharedPtr< TheClass >::OsciSharedPtr (TheClass * *inClassPtr*, *OsciRefCounter* * *in_refcnt*) [inline]

Constructor.

Parameters:

inClassPtr A pointer to an instance of the parameterized type that the new OsciSharedPtr will wrap.

7.214.2.3 template<class TheClass> OsciSharedPtr< TheClass >::OsciSharedPtr (const OsciSharedPtr< TheClass > & *inSharedPtr*) [inline]

Copy constructor.

7.214.2.4 template<class TheClass> virtual OsciSharedPtr< TheClass >::~~OsciSharedPtr () [inline, virtual]

Destructor.

7.214.3 Member Function Documentation

7.214.3.1 template<class TheClass> int OsciSharedPtr< TheClass >::get_count () [inline]

Get a count of how many references to the object exist.

7.214.3.2 template<class TheClass> *OsciRefCounter OsciSharedPtr< TheClass >::GetRefCounter () [inline]**

Get the refcount pointer. This should primarily be used for conversion operations.

7.214.3.3 template<class TheClass> TheClass* OsciSharedPtr< TheClass >::GetRep () [inline]

Use this function to get a pointer to the wrapped object.

7.214.3.4 `template<class TheClass> TheClass& OsclSharedPtr< TheClass >::operator * ()`
[inline]

The indirection operator returns a reference to an object of the parameterized type.

7.214.3.5 `template<class TheClass> OsclSharedPtr< TheClass >::operator TheClass * ()`
[inline]

Casting operator.

7.214.3.6 `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.

7.214.3.7 `template<class TheClass> OsclSharedPtr& OsclSharedPtr< TheClass >::operator=`
`(const OsclSharedPtr< TheClass > & inSharedPtr) [inline]`

Assignment operator.

7.214.3.8 `template<class TheClass> void OsclSharedPtr< TheClass >::Unbind ()` [inline]

Use this function of unbind an existing OsclSharedPtr.

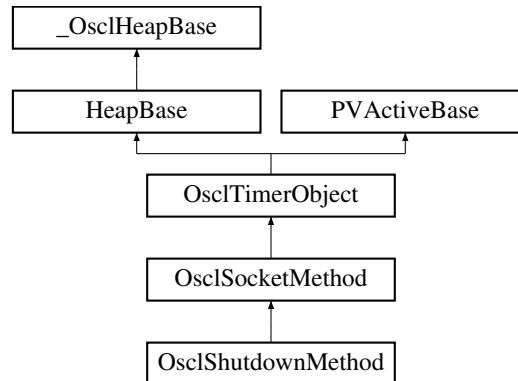
The documentation for this class was generated from the following file:

- [oscl_shared_ptr.h](#)

7.215 OscShutdownMethod Class Reference

```
#include <oscl_socket_shutdown.h>
```

Inheritance diagram for OscShutdownMethod::



Public Methods

- [~OscShutdownMethod \(\)](#)
- [TPVSocketEvent Shutdown \(TPVSocketShutdown aHow, int32 aTimeout\)](#)
- [OscShutdownRequest * ShutdownRequest \(\)](#)

Static Public Methods

- [OscShutdownMethod * NewL \(OscIPSocketI &c\)](#)

7.215.1 Constructor & Destructor Documentation

7.215.1.1 [OscShutdownMethod::~~OscShutdownMethod \(\)](#)

7.215.2 Member Function Documentation

7.215.2.1 [OscShutdownMethod* OscShutdownMethod::NewL \(OscIPSocketI &c\) \[static\]](#)

7.215.2.2 [TPVSocketEvent OscShutdownMethod::Shutdown \(TPVSocketShutdown aHow, int32 aTimeout\)](#)

7.215.2.3 [OscShutdownRequest* OscShutdownMethod::ShutdownRequest \(\) \[inline\]](#)

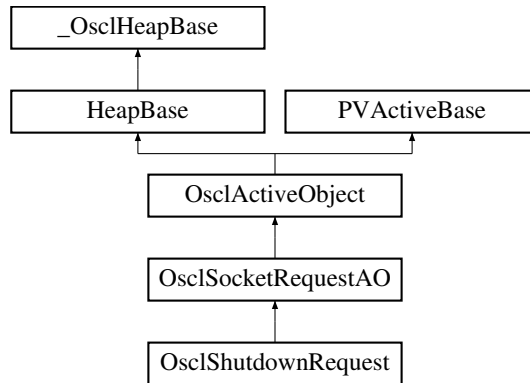
The documentation for this class was generated from the following file:

- [oscl_socket_shutdown.h](#)

7.216 OsciShutdownRequest Class Reference

```
#include <osci_socket_shutdown.h>
```

Inheritance diagram for OsciShutdownRequest::



Public Methods

- [OsciShutdownRequest](#) ([OsciSocketMethod](#) &c)
- void [Shutdown](#) ([TPVSocketShutdown](#) aHow)

7.216.1 Detailed Description

This is the AO that interacts with the socket server

7.216.2 Constructor & Destructor Documentation

7.216.2.1 [OsciShutdownRequest::OsciShutdownRequest](#) ([OsciSocketMethod](#) & c) [inline]

7.216.3 Member Function Documentation

7.216.3.1 void [OsciShutdownRequest::Shutdown](#) ([TPVSocketShutdown](#) aHow)

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

- [osci_socket_shutdown.h](#)

7.217 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 >
```

7.217.1 Constructor & Destructor Documentation

7.217.1.1 `template<class T, uint32 ID, class Registry = OsclSingletonRegistry> OsclSingleton< T, ID, Registry >::OsclSingleton () [inline]`

7.217.1.2 `template<class T, uint32 ID, class Registry = OsclSingletonRegistry> OsclSingleton< T, ID, Registry >::~~OsclSingleton () [inline]`

7.217.2 Member Function Documentation

7.217.2.1 `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.

7.217.2.2 `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.

7.217.2.3 `template<class T, uint32 ID, class Registry = OsciSingletonRegistry> bool
OsciSingleton< 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.

7.217.3 Field Documentation

7.217.3.1 `template<class T, uint32 ID, class Registry = OsciSingletonRegistry> T*
OsciSingleton< T, ID, Registry >::_Ptr [protected]`

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

- [osci_singleton.h](#)

7.218 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](#)

7.218.1 Member Function Documentation

7.218.1.1 OSCL_IMPORT_REF [OsclAny](#)* [OsclSingletonRegistry::getInstance](#) (uint32 *ID*, int32 & *error*) [static]

7.218.1.2 OSCL_IMPORT_REF [OsclAny](#)* [OsclSingletonRegistry::lockAndGetInstance](#) (uint32 *ID*, int32 & *error*) [static]

7.218.1.3 OSCL_IMPORT_REF void [OsclSingletonRegistry::registerInstance](#) ([OsclAny](#) * *ptr*, uint32 *ID*, int32 & *error*) [static]

7.218.1.4 OSCL_IMPORT_REF void [OsclSingletonRegistry::registerInstanceAndUnlock](#) ([OsclAny](#) * *ptr*, uint32 *ID*, int32 & *error*) [static]

7.218.2 Friends And Related Function Documentation

7.218.2.1 friend class [OsclBase](#) [friend]

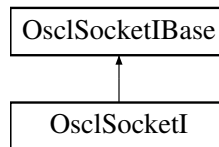
The documentation for this class was generated from the following file:

- [oscl_singleton.h](#)

7.219 OsciSocketI Class Reference

```
#include <osci_socket_imp_pv.h>
```

Inheritance diagram for OsciSocketI:



Public Methods

- [~OsciSocketI](#) ()
- [int32 Open](#) ([OsciSocketServI](#) &aServer, uint32 addrFamily, uint32 sockType, uint32 protocol)
- [int32 Open](#) ([OsciSocketServI](#) &aServer)
- [int32 Bind](#) ([OsciNetworkAddress](#) &anAddr)
- [int32 SetSockOpt](#) ([TPVSocketOptionLevel](#) aOptionLevel, [TPVSocketOptionName](#) aOptionName, [OsciAny](#) *aOptionValue, int32 aOptionLen)
- [int32 GetPeerName](#) ([OsciNetworkAddress](#) &peerName)
- [int32 Join](#) ([OsciNetworkAddress](#) &anAddr)
- [int32 Close](#) ()
- [int32 Listen](#) (uint32 qSize)
- [int32 SetRecvBufferSize](#) (uint32 size)
- [TPVSocketEvent ThreadLogoff](#) ()
- [TPVSocketEvent ThreadLogon](#) ([OsciSocketServI](#) *aServ)
- [void Connect](#) ([ConnectParam](#) &, [OsciSocketRequestAO](#) &)
- [void Accept](#) ([AcceptParam](#) &, [OsciSocketRequestAO](#) &)
- [void Shutdown](#) ([ShutdownParam](#) &, [OsciSocketRequestAO](#) &)
- [void Send](#) ([SendParam](#) &, [OsciSocketRequestAO](#) &)
- [void SendSuccess](#) ([SendParam](#) &)
- [void SendTo](#) ([SendToParam](#) &, [OsciSocketRequestAO](#) &)
- [void SendToSuccess](#) ([SendToParam](#) &)
- [void Recv](#) ([RecvParam](#) &, [OsciSocketRequestAO](#) &)
- [void RecvSuccess](#) ([RecvParam](#) &)
- [void RecvFrom](#) ([RecvFromParam](#) &, [OsciSocketRequestAO](#) &)
- [void RecvFromSuccess](#) ([RecvFromParam](#) &)
- [TOsciSocket Socket](#) ()
- [void ProcessConnect](#) ([OsciSocketServRequestQElem](#) *)
- [void ProcessShutdown](#) ([OsciSocketServRequestQElem](#) *)
- [void ProcessAccept](#) ([OsciSocketServRequestQElem](#) *)
- [void ProcessSendTo](#) ([OsciSocketServRequestQElem](#) *)
- [void ProcessRecvFrom](#) ([OsciSocketServRequestQElem](#) *)
- [void ProcessSend](#) ([OsciSocketServRequestQElem](#) *)
- [void ProcessRecv](#) ([OsciSocketServRequestQElem](#) *)
- [PVLogger * Logger](#) ()

Static Public Methods

- OsclSocketI * [NewL](#) ([Oscl_DefAlloc](#) &a)
- bool [MakeAddr](#) ([OsclNetworkAddress](#) &in, [TOsclSockAddr](#) &addr)
- void [MakeAddr](#) ([TOsclSockAddr](#) &in, [OsclNetworkAddress](#) &addr)
- bool [MakeMulticastGroupInformation](#) ([OsclIpMReq](#) &in, [TIpMReq](#) &addr)
- void [MakeMulticastGroupInformation](#) ([TIpMReq](#) &in, [OsclIpMReq](#) &addr)

Friends

- class [OsclAcceptRequest](#)
- class [OsclConnectRequest](#)
- class [OsclRecvRequest](#)
- class [OsclRecvFromRequest](#)
- class [OsclSendRequest](#)
- class [OsclSendToRequest](#)
- class [OsclShutdownRequest](#)
- class [OsclUDPSocket](#)
- class [OsclTCPSocket](#)

7.219.1 Detailed Description

Socket implementation class

7.219.2 Constructor & Destructor Documentation

7.219.2.1 [OsclSocketI::~OsclSocketI](#) ()

7.219.3 Member Function Documentation

7.219.3.1 [void OsclSocketI::Accept](#) ([AcceptParam](#) &, [OsclSocketRequestAO](#) &) [virtual]

Implements [OsclSocketIBase](#).

7.219.3.2 [int32 OsclSocketI::Bind](#) ([OsclNetworkAddress](#) & *anAddr*) [virtual]

Implements [OsclSocketIBase](#).

7.219.3.3 [int32 OsclSocketI::Close](#) () [virtual]

Implements [OsclSocketIBase](#).

7.219.3.4 [void OsclSocketI::Connect](#) ([ConnectParam](#) &, [OsclSocketRequestAO](#) &) [virtual]

Implements [OsclSocketIBase](#).

7.219.3.5 `int32 OslSocketI::GetPeerName (OslNetworkAddress & peerName)`

7.219.3.6 `int32 OslSocketI::Join (OslNetworkAddress & anAddr) [virtual]`

Implements [OslSocketIBase](#).

7.219.3.7 `int32 OslSocketI::Listen (uint32 qSize) [virtual]`

Implements [OslSocketIBase](#).

7.219.3.8 `PVLogger* OslSocketI::Logger () [inline]`

7.219.3.9 `void OslSocketI::MakeAddr (TOslSockAddr & in, OslNetworkAddress & addr) [static]`

7.219.3.10 `bool OslSocketI::MakeAddr (OslNetworkAddress & in, TOslSockAddr & addr) [static]`

7.219.3.11 `void OslSocketI::MakeMulticastGroupInformation (TIpMReq & in, OslIpMReq & addr) [static]`

7.219.3.12 `bool OslSocketI::MakeMulticastGroupInformation (OslIpMReq & in, TIpMReq & addr) [static]`

7.219.3.13 `OslSocketI* OslSocketI::NewL (Osl_DefAlloc & a) [static]`

7.219.3.14 `int32 OslSocketI::Open (OslSocketServI & aServer) [virtual]`

Implements [OslSocketIBase](#).

7.219.3.15 `int32 OslSocketI::Open (OslSocketServI & aServer, uint32 addrFamily, uint32 sockType, uint32 protocol) [virtual]`

Implements [OslSocketIBase](#).

7.219.3.16 void OsclSocketI::ProcessAccept ([OsclSocketServRequestQElem *](#))

7.219.3.17 void OsclSocketI::ProcessConnect ([OsclSocketServRequestQElem *](#))

7.219.3.18 void OsclSocketI::ProcessRecv ([OsclSocketServRequestQElem *](#))

7.219.3.19 void OsclSocketI::ProcessRecvFrom ([OsclSocketServRequestQElem *](#))

7.219.3.20 void OsclSocketI::ProcessSend ([OsclSocketServRequestQElem *](#))

7.219.3.21 void OsclSocketI::ProcessSendTo ([OsclSocketServRequestQElem *](#))

7.219.3.22 void OsclSocketI::ProcessShutdown ([OsclSocketServRequestQElem *](#))

7.219.3.23 void OsclSocketI::Recv ([RecvParam &](#), [OsclSocketRequestAO &](#)) [virtual]

Implements [OsclSocketIBase](#).

7.219.3.24 void OsclSocketI::RecvFrom ([RecvFromParam &](#), [OsclSocketRequestAO &](#))
[virtual]

Implements [OsclSocketIBase](#).

7.219.3.25 void OsclSocketI::RecvFromSuccess ([RecvFromParam &](#)) [virtual]

Implements [OsclSocketIBase](#).

7.219.3.26 void OsclSocketI::RecvSuccess ([RecvParam &](#)) [virtual]

Implements [OsclSocketIBase](#).

7.219.3.27 void OsclSocketI::Send ([SendParam &](#), [OsclSocketRequestAO &](#)) [virtual]

Implements [OsclSocketIBase](#).

7.219.3.28 void OsclSocketI::SendSuccess ([SendParam &](#)) [virtual]

Implements [OsclSocketIBase](#).

7.219.3.29 void OsclSocketI::SendTo ([SendToParam &](#), [OsclSocketRequestAO &](#)) [virtual]

Implements [OsclSocketIBase](#).

7.219.3.30 void OsclSocketI::SendToSuccess ([SendToParam &](#)) [virtual]

Implements [OsclSocketIBase](#).

7.219.3.31 `int32 OsclSocketI::SetRecvBufferSize (uint32 size)`

7.219.3.32 `int32 OsclSocketI::SetSockOpt (TPVSocketOptionLevel aOptionLevel, TPVSocketOptionName aOptionName, OsclAny * aOptionValue, int32 aOptionLen)`

7.219.3.33 `void OsclSocketI::Shutdown (ShutdownParam &, OsclSocketRequestAO &)`
[virtual]

Implements [OsclSocketIBase](#).

7.219.3.34 `TOsclSocket OsclSocketI::Socket ()` [inline]

7.219.3.35 `TPVSocketEvent OsclSocketI::ThreadLogoff ()`

7.219.3.36 `TPVSocketEvent OsclSocketI::ThreadLogon (OsclSocketServI * aServ)`

7.219.4 Friends And Related Function Documentation

7.219.4.1 friend class `OsclAcceptRequest` [friend]

7.219.4.2 friend class `OsclConnectRequest` [friend]

7.219.4.3 friend class `OsclRecvFromRequest` [friend]

7.219.4.4 friend class `OsclRecvRequest` [friend]

7.219.4.5 friend class `OsclSendRequest` [friend]

7.219.4.6 friend class `OsclSendToRequest` [friend]

7.219.4.7 friend class `OsclShutdownRequest` [friend]

7.219.4.8 friend class `OsclTCPSocket` [friend]

Reimplemented from [OsclSocketIBase](#).

7.219.4.9 friend class `OsclUDPSocket` [friend]

Reimplemented from [OsclSocketIBase](#).

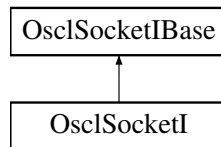
The documentation for this class was generated from the following file:

- [oscl_socket_imp_pv.h](#)

7.220 OsciSocketIBase Class Reference

```
#include <osci_socket_imp_base.h>
```

Inheritance diagram for OsciSocketIBase::



Public Methods

- virtual `~OsciSocketIBase ()`
- virtual `int32 Open (OsciSocketServI &aServer, uint32 addrFamily, uint32 sockType, uint32 protocol)=0`
- virtual `int32 Open (OsciSocketServI &aServer)=0`
- virtual `int32 Bind (OsciNetworkAddress &anAddr)=0`
- virtual `int32 Join (OsciNetworkAddress &anAddr)=0`
- virtual `int32 Close ()=0`
- virtual `int32 Listen (uint32 qSize)=0`
- virtual `void Connect (ConnectParam &, OsciSocketRequestAO &)=0`
- virtual `void Accept (AcceptParam &, OsciSocketRequestAO &)=0`
- virtual `void Shutdown (ShutdownParam &, OsciSocketRequestAO &)=0`
- virtual `void Send (SendParam &, OsciSocketRequestAO &)=0`
- virtual `void SendSuccess (SendParam &)=0`
- virtual `void SendTo (SendToParam &, OsciSocketRequestAO &)=0`
- virtual `void SendToSuccess (SendToParam &)=0`
- virtual `void Recv (RecvParam &, OsciSocketRequestAO &)=0`
- virtual `void RecvSuccess (RecvParam &)=0`
- virtual `void RecvFrom (RecvFromParam &, OsciSocketRequestAO &)=0`
- virtual `void RecvFromSuccess (RecvFromParam &)=0`
- virtual `void BindAsync (BindParam &, OsciSocketRequestAO &)`
- virtual `void ListenAsync (ListenParam &, OsciSocketRequestAO &)`
- `void CancelFxn (TPVSocketFxn)`

Static Public Methods

- `bool HasAsyncBind ()`
- `bool HasAsyncListen ()`

Protected Methods

- `OsciSocketIBase (Osci_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](#) aOsciVal)

Protected Attributes

- [Osci_DefAlloc](#) & [iAlloc](#)
- [OsciSocketServI](#) * [iSocketServ](#)

Friends

- class [OsciSocketRequest](#)
- class [OsciSocketMethod](#)
- class [OsciSocketRequestAO](#)
- class [OsciUDPSocket](#)
- class [OsciTCPSocket](#)

7.220.1 Detailed Description

Socket implementation base class

7.220.2 Constructor & Destructor Documentation

7.220.2.1 virtual [OsciSocketIBase::~OsciSocketIBase](#) () [virtual]

7.220.2.2 [OsciSocketIBase::OsciSocketIBase](#) ([Osci_DefAlloc](#) & *a*) [protected]

7.220.3 Member Function Documentation

7.220.3.1 virtual void [OsciSocketIBase::Accept](#) ([AcceptParam](#) &, [OsciSocketRequestAO](#) &) [pure virtual]

Implemented in [OsciSocketI](#).

7.220.3.2 virtual int32 [OsciSocketIBase::Bind](#) ([OsciNetworkAddress](#) & *anAddr*) [pure virtual]

Implemented in [OsciSocketI](#).

- 7.220.3.3 **virtual void OsclSocketIBase::BindAsync** ([BindParam](#) &, [OsclSocketRequestAO](#) &) [inline, virtual]
- 7.220.3.4 **virtual void OsclSocketIBase::CancelAccept** () [protected, pure virtual]
- 7.220.3.5 **virtual void OsclSocketIBase::CancelBind** () [inline, protected, virtual]
- 7.220.3.6 **virtual void OsclSocketIBase::CancelConnect** () [protected, pure virtual]
- 7.220.3.7 **void OsclSocketIBase::CancelFxn** ([TPVSocketFxn](#))
- 7.220.3.8 **virtual void OsclSocketIBase::CancelListen** () [inline, protected, virtual]
- 7.220.3.9 **virtual void OsclSocketIBase::CancelRecv** () [protected, pure virtual]
- 7.220.3.10 **virtual void OsclSocketIBase::CancelRecvFrom** () [protected, pure virtual]
- 7.220.3.11 **virtual void OsclSocketIBase::CancelSend** () [protected, pure virtual]
- 7.220.3.12 **virtual void OsclSocketIBase::CancelSendTo** () [protected, pure virtual]
- 7.220.3.13 **virtual void OsclSocketIBase::CancelShutdown** () [protected, pure virtual]
- 7.220.3.14 **virtual int32 OsclSocketIBase::Close** () [pure virtual]

Implemented in [OsclSocketI](#).

- 7.220.3.15 **virtual void OsclSocketIBase::Connect** ([ConnectParam](#) &, [OsclSocketRequestAO](#) &) [pure virtual]

Implemented in [OsclSocketI](#).

- 7.220.3.16 **int OsclSocketIBase::GetShutdown** ([TPVSocketShutdown](#) *aOscIVal*) [static, protected]
- 7.220.3.17 **bool OsclSocketIBase::HasAsyncBind** () [static]
- 7.220.3.18 **bool OsclSocketIBase::HasAsyncListen** () [static]
- 7.220.3.19 **virtual bool OsclSocketIBase::IsOpen** () [protected, pure virtual]
- 7.220.3.20 **virtual int32 OsclSocketIBase::Join** ([OsclNetworkAddress](#) & *anAddr*) [pure virtual]

Implemented in [OsclSocketI](#).

- 7.220.3.21 **virtual int32 OsclSocketIBase::Listen** (uint32 *qSize*) [pure virtual]

Implemented in [OsclSocketI](#).

7.220.3.22 `virtual void OsciSocketIBase::ListenAsync (ListenParam &, OsciSocketRequestAO &)` [inline, virtual]

7.220.3.23 `virtual int32 OsciSocketIBase::Open (OsciSocketServI & aServer)` [pure virtual]

Implemented in [OsciSocketI](#).

7.220.3.24 `virtual int32 OsciSocketIBase::Open (OsciSocketServI & aServer, uint32 addrFamily, uint32 sockType, uint32 protocol)` [pure virtual]

Implemented in [OsciSocketI](#).

7.220.3.25 `virtual void OsciSocketIBase::Recv (RecvParam &, OsciSocketRequestAO &)` [pure virtual]

Implemented in [OsciSocketI](#).

7.220.3.26 `virtual void OsciSocketIBase::RecvFrom (RecvFromParam &, OsciSocketRequestAO &)` [pure virtual]

Implemented in [OsciSocketI](#).

7.220.3.27 `virtual void OsciSocketIBase::RecvFromSuccess (RecvFromParam &)` [pure virtual]

Implemented in [OsciSocketI](#).

7.220.3.28 `virtual void OsciSocketIBase::RecvSuccess (RecvParam &)` [pure virtual]

Implemented in [OsciSocketI](#).

7.220.3.29 `virtual void OsciSocketIBase::Send (SendParam &, OsciSocketRequestAO &)` [pure virtual]

Implemented in [OsciSocketI](#).

7.220.3.30 `virtual void OsciSocketIBase::SendSuccess (SendParam &)` [pure virtual]

Implemented in [OsciSocketI](#).

7.220.3.31 `virtual void OsciSocketIBase::SendTo (SendToParam &, OsciSocketRequestAO &)` [pure virtual]

Implemented in [OsciSocketI](#).

7.220.3.32 virtual void OslSocketIBase::SendToSuccess ([SendToParam](#) &) [pure virtual]

Implemented in [OslSocketI](#).

7.220.3.33 virtual void OslSocketIBase::Shutdown ([ShutdownParam](#) &, [OslSocketRequestAO](#) &) [pure virtual]

Implemented in [OslSocketI](#).

7.220.4 Friends And Related Function Documentation

7.220.4.1 friend class OslSocketMethod [friend]

7.220.4.2 friend class OslSocketRequest [friend]

7.220.4.3 friend class OslSocketRequestAO [friend]

7.220.4.4 friend class OslTCPSocket [friend]

Reimplemented in [OslSocketI](#).

7.220.4.5 friend class OslUDPSocket [friend]

Reimplemented in [OslSocketI](#).

7.220.5 Field Documentation

7.220.5.1 [Osl_DefAlloc](#)& OslSocketIBase::iAlloc [protected]

7.220.5.2 [OslSocketServI](#)* OslSocketIBase::iSocketServ [protected]

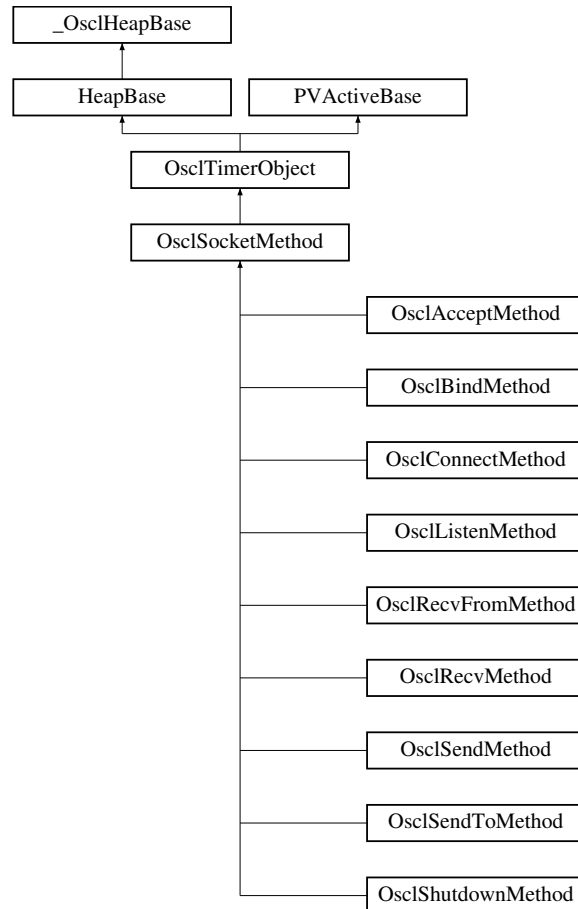
The documentation for this class was generated from the following file:

- [osl_socket_imp_base.h](#)

7.221 OsciSocketMethod Class Reference

```
#include <osci_socket_method.h>
```

Inheritance diagram for OsciSocketMethod::



Public Methods

- [OsciSocketMethod](#) ([OsciIPSocketI](#) &aContainer, const char *name, [TPVSocketFxn](#) fxn)
- virtual [~OsciSocketMethod](#) ()
- void [Abort](#) ()
- void [AbortAll](#) ()
- void [CancelMethod](#) ()
- [Osci_DefAlloc](#) & [Alloc](#) ()
- [TPVSocketEvent](#) [ThreadLogon](#) ()
- [TPVSocketEvent](#) [ThreadLogoff](#) ()

Data Fields

- [OsciIPSocketI](#) & [iContainer](#)
- [TPVSocketFxn](#) [iSocketFxn](#)

Protected Methods

- void [ConstructL](#) ([OsclSocketRequestAO](#) *aAO)
- bool [StartMethod](#) (int32 aTimeoutMsec)
- void [MethodDone](#) ()
- void [Run](#) ()

Protected Attributes

- [OsclSocketRequestAO](#) * [iSocketRequestAO](#)

7.221.1 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.

7.221.2 Constructor & Destructor Documentation

7.221.2.1 [OsclSocketMethod::OsclSocketMethod](#) ([OsclIPSocketI](#) & *aContainer*, const char * *name*, [TPVSocketFxn](#) *fxn*) [inline]

7.221.2.2 virtual [OsclSocketMethod::~~OsclSocketMethod](#) () [inline, virtual]

7.221.3 Member Function Documentation

7.221.3.1 void [OsclSocketMethod::Abort](#) () [inline]

7.221.3.2 void [OsclSocketMethod::AbortAll](#) () [inline]

7.221.3.3 [Oscl_DefAlloc](#)& [OsclSocketMethod::Alloc](#) () [inline]

7.221.3.4 void [OsclSocketMethod::CancelMethod](#) () [inline]

7.221.3.5 void [OsclSocketMethod::ConstructL](#) ([OsclSocketRequestAO](#) * *aAO*) [inline, protected]

7.221.3.6 void [OsclSocketMethod::MethodDone](#) () [inline, protected]

7.221.3.7 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](#).

7.221.3.8 `bool OsciSocketMethod::StartMethod (int32 aTimeoutMsec)` [protected]

7.221.3.9 `TPVSocketEvent OsciSocketMethod::ThreadLogoff ()`

7.221.3.10 `TPVSocketEvent OsciSocketMethod::ThreadLogon ()`

7.221.4 Field Documentation

7.221.4.1 `OsciIPSocketI& OsciSocketMethod::iContainer`

7.221.4.2 `TPVSocketFxn OsciSocketMethod::iSocketFxn`

7.221.4.3 `OsciSocketRequestAO* OsciSocketMethod::iSocketRequestAO` [protected]

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

- [osci_socket_method.h](#)

7.222 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](#) ()

7.222.1 Detailed Description

Socket event observer. The client implements this to get asynchronous command completion.

7.222.2 Constructor & Destructor Documentation

7.222.2.1 virtual [OsclSocketObserver::~OsclSocketObserver](#) () [inline, virtual]

7.222.3 Member Function Documentation

7.222.3.1 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 EPVSocket-Failure.

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](#)

7.223 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

7.223.1 Detailed Description

This class defines the request interface to the PV socket server.

7.223.2 Constructor & Destructor Documentation

7.223.2.1 [OsclSocketRequest::OsclSocketRequest](#) () [inline]

7.223.3 Member Function Documentation

7.223.3.1 void [OsclSocketRequest::Activate](#) ([SocketRequestParam](#) * *iParam*, [OsclSocketRequestAO](#) & *a*)

7.223.3.2 void [OsclSocketRequest::CancelRequest](#) ()

7.223.3.3 void [OsclSocketRequest::Complete](#) ([OsclSocketServRequestQElem](#) *, int32 *aStatus*, int32 *aSockErr* = 0)

7.223.3.4 [TPVSocketFxn](#) [OsclSocketRequest::Fxn](#) () [inline]

7.223.4 Field Documentation

7.223.4.1 [SocketRequestParam](#)* [OsclSocketRequest::iParam](#)

7.223.4.2 [OsclSocketI](#)* [OsclSocketRequest::iSocketI](#)

7.223.4.3 [OsclSocketRequestAO](#)* [OsclSocketRequest::iSocketRequestAO](#)

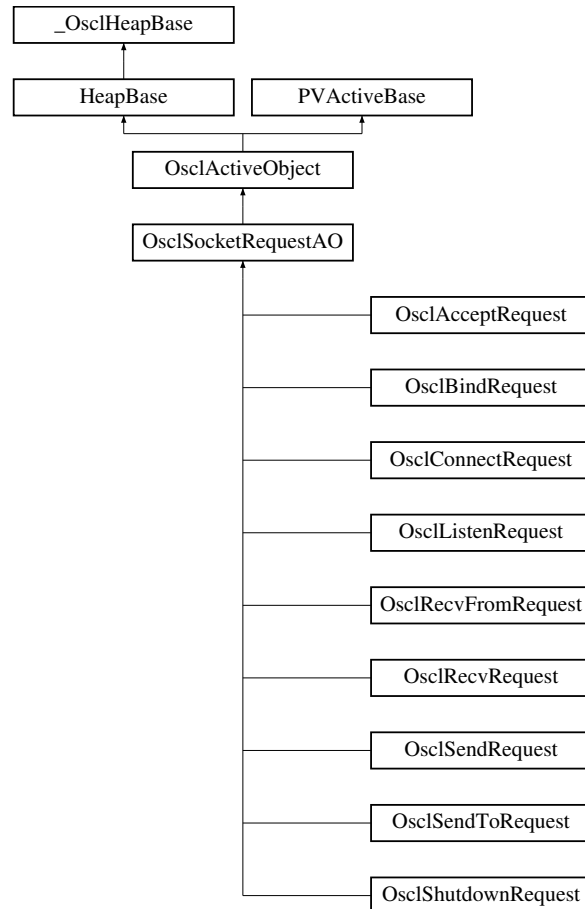
The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

7.224 OsciSocketRequestAO Class Reference

```
#include <osci_socket_method.h>
```

Inheritance diagram for OsciSocketRequestAO::



Public Methods

- void [ConstructL](#) ()

Protected Methods

- [OsciSocketRequestAO](#) ([OsciSocketMethod](#) &aContainer, const char *name)
- virtual [~OsciSocketRequestAO](#) ()
- [OsciAny](#) * [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](#)

7.224.1 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

7.224.2 Constructor & Destructor Documentation

7.224.2.1 [OsclSocketRequestAO::OsclSocketRequestAO](#) ([OsclSocketMethod](#) & *aContainer*, const char * *name*) [inline, protected]

7.224.2.2 virtual [OsclSocketRequestAO::~~OsclSocketRequestAO](#) () [inline, protected, virtual]

7.224.3 Member Function Documentation

7.224.3.1 void [OsclSocketRequestAO::Abort](#) () [inline, protected]

7.224.3.2 [Oscl_DefAlloc](#)& [OsclSocketRequestAO::Alloc](#) () [inline, protected]

7.224.3.3 void [OsclSocketRequestAO::CleanupParam](#) (bool *deallocate* = false) [protected]

7.224.3.4 void [OsclSocketRequestAO::ConstructL](#) () [inline]

7.224.3.5 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](#).

7.224.3.6 `int OsciSocketRequestAO::GetSocketError ()` [inline, protected]

7.224.3.7 `uint32 OsciSocketRequestAO::Id ()` [inline, protected]

7.224.3.8 `OsciAny* OsciSocketRequestAO::NewRequest (const uint32 size)` [protected]

7.224.3.9 `void OsciSocketRequestAO::RequestDone ()` [inline, protected]

7.224.3.10 `void OsciSocketRequestAO::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](#).

7.224.3.11 `OsciSocketI* OsciSocketRequestAO::SocketI ()` [inline, protected]

7.224.3.12 `OsciSocketObserver* OsciSocketRequestAO::SocketObserver ()` [inline, protected]

7.224.3.13 `virtual void OsciSocketRequestAO::Success ()` [inline, protected, virtual]

Reimplemented in [OsciRecvRequest](#), [OsciRecvFromRequest](#), [OsciSendRequest](#), and [OsciSendToRequest](#).

7.224.4 Friends And Related Function Documentation

7.224.4.1 friend class OsclSocketI [friend]

7.224.4.2 friend class OsclSocketMethod [friend]

7.224.4.3 friend class OsclSocketRequest [friend]

7.224.5 Field Documentation

7.224.5.1 [OsclSocketMethod](#)& OsclSocketRequestAO::iContainer [protected]

7.224.5.2 [SocketRequestParam](#)* OsclSocketRequestAO::iParam [protected]

7.224.5.3 uint32 OsclSocketRequestAO::iParamSize [protected]

7.224.5.4 int32 OsclSocketRequestAO::iSocketError [protected]

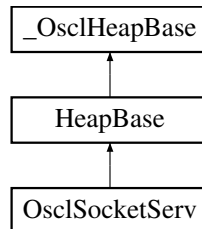
The documentation for this class was generated from the following file:

- [oscl_socket_method.h](#)

7.225 OslSocketServ Class Reference

```
#include <oscl_socket.h>
```

Inheritance diagram for OslSocketServ::



Public Methods

- OSCL_IMPORT_REF [~OslSocketServ](#) ()
- OSCL_IMPORT_REF int32 [Connect](#) (uint32 aMessageSlots=8, bool aShareSession=false)
- OSCL_IMPORT_REF void [Close](#) (bool aCleanup=true)

Static Public Methods

- OSCL_IMPORT_REF OslSocketServ * [NewL](#) (Osl_DefAlloc &alloc)

Friends

- class [OslTCPSocket](#)
- class [OslUDPSocket](#)
- class [OslDNS](#)

7.225.1 Constructor & Destructor Documentation

7.225.1.1 OSCL_IMPORT_REF OslSocketServ::~~OslSocketServ ()

Destructor. The server object must be deleted using the same allocator used in the NewL call.

7.225.2 Member Function Documentation

7.225.2.1 OSCL_IMPORT_REF void OslSocketServ::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.

7.225.2.2 OSCL_IMPORT_REF int32 OsciSocketServ::Connect (uint32 *aMessageSlots* = 8, bool *aShareSession* = false)

Connect to socket server. This is a synchronous method.

Parameters:

Number of message slots.

Returns:

Returns OsciErrNone for success, or a platform-specific code.

7.225.2.3 OSCL_IMPORT_REF OsciSocketServ* OsciSocketServ::NewL (Osci_DefAlloc & *alloc*) [static]

Create a socket server. May leave if failure.

Parameters:

alloc: Memory allocator.

Returns:

Returns pointer to socket server

7.225.3 Friends And Related Function Documentation

7.225.3.1 friend class OsciDNS [friend]

7.225.3.2 friend class OsciTCPSocket [friend]

7.225.3.3 friend class OsciUDPSocket [friend]

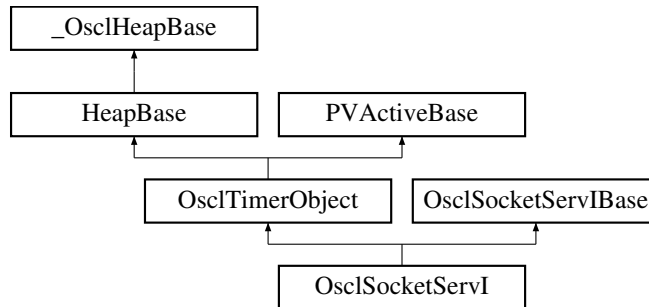
The documentation for this class was generated from the following file:

- [oscl_socket.h](#)

7.226 OsciSocketServI Class Reference

```
#include <osci_socket_serv_imp_pv.h>
```

Inheritance diagram for OsciSocketServI::



Public Methods

- int32 [Connect](#) (uint32 aMessageSlots, bool aShareSession)
- void [Close](#) (bool)
- bool [IsServerThread](#) ()

Static Public Methods

- OsciSocketServI * [NewL](#) (Osci_DefAlloc &a)

Friends

- class [OsciSocketServRequestList](#)
- class [LoopbackSocket](#)
- class [OsciTCPSocketI](#)
- class [OsciUDPSocketI](#)
- class [OsciSocketI](#)
- class [OsciDNSI](#)
- class [OsciSocketRequest](#)
- class [OsciSocketServ](#)

7.226.1 Detailed Description

PV socket server implementation

7.226.2 Member Function Documentation

7.226.2.1 void OsciSocketServI::Close (bool) [virtual]

Implements [OsciSocketServIBase](#).

7.226.2.2 `int32 OsciSocketServI::Connect (uint32 aMessageSlots, bool aShareSession)`
[virtual]

Implements [OsciSocketServIBase](#).

7.226.2.3 `bool OsciSocketServI::IsServerThread ()`

7.226.2.4 `OsciSocketServI* OsciSocketServI::NewL (Osci_DefAlloc & a)` [static]

7.226.3 Friends And Related Function Documentation

7.226.3.1 friend class `LoopbackSocket` [friend]

7.226.3.2 friend class `OsciDNSI` [friend]

7.226.3.3 friend class `OsciSocketI` [friend]

7.226.3.4 friend class `OsciSocketRequest` [friend]

7.226.3.5 friend class `OsciSocketServ` [friend]

7.226.3.6 friend class `OsciSocketServRequestList` [friend]

7.226.3.7 friend class `OsciTCPSocketI` [friend]

7.226.3.8 friend class `OsciUDPSocketI` [friend]

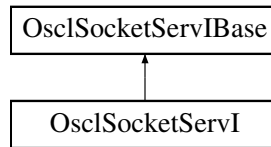
The documentation for this class was generated from the following file:

- [osci_socket_serv_imp_pv.h](#)

7.227 OscSocketServIBase Class Reference

```
#include <oscl_socket_serv_imp_base.h>
```

Inheritance diagram for OscSocketServIBase::



Public Methods

- virtual `~OscSocketServIBase()`
- virtual `int32 Connect` (`uint32 aMessageSlots`, `bool aShareSession`)=0
- virtual `void Close` (`bool`)=0

Data Fields

- `PVLogger * iLogger`

Protected Types

- enum `TSocketServState` { `ESocketServ_Idle`, `ESocketServ_Connected`, `ESocketServ_Error` }

Protected Methods

- `OscSocketServIBase` (`Osc_DefAlloc &a`)
- `TSocketServState State` () const
- `bool IsServConnected` () const

Protected Attributes

- `Osc_DefAlloc & iAlloc`
- `TSocketServState iServState`
- `int iServError`

7.227.1 Detailed Description

Base class common to all implementations

7.227.2 Member Enumeration Documentation

7.227.2.1 enum `OscSocketServIBase::TSocketServState` [protected]

Enumeration values:

`ESocketServ_Idle`

ESocketServ_Connected

ESocketServ_Error

7.227.3 Constructor & Destructor Documentation

7.227.3.1 virtual OsciSocketServIBase::~~OsciSocketServIBase () [inline, virtual]

7.227.3.2 OsciSocketServIBase::OsciSocketServIBase ([Osci_DefAlloc](#) & *a*) [inline, protected]

7.227.4 Member Function Documentation

7.227.4.1 virtual void OsciSocketServIBase::Close (bool) [pure virtual]

Implemented in [OsciSocketServI](#).

7.227.4.2 virtual int32 OsciSocketServIBase::Connect (uint32 *aMessageSlots*, bool *aShareSession*) [pure virtual]

Implemented in [OsciSocketServI](#).

7.227.4.3 bool OsciSocketServIBase::IsServConnected () const [inline, protected]

7.227.4.4 [TSocketServState](#) OsciSocketServIBase::State () const [inline, protected]

7.227.5 Field Documentation

7.227.5.1 [Osci_DefAlloc](#)& OsciSocketServIBase::iAlloc [protected]

7.227.5.2 [PVLogger](#)* OsciSocketServIBase::iLogger

7.227.5.3 int OsciSocketServIBase::iServError [protected]

7.227.5.4 [TSocketServState](#) OsciSocketServIBase::iServState [protected]

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

- [osci_socket_serv_imp_base.h](#)

7.228 OsciSocketServRequestList Class Reference

```
#include <osci_socket_serv_imp_reqlist.h>
```

Public Methods

- [OsciSocketServRequestList](#) ()
- void [Add](#) ([OsciSocketRequest](#) *)
- void [StartCancel](#) ([OsciSocketRequest](#) *)
- void [Open](#) ([OsciSocketServI](#) *s)
- void [Close](#) ()
- void [Wakeup](#) ()
- void [WaitOnRequests](#) ()
- void [Remove](#) ([OsciSocketServRequestQElem](#) *aElem)

Friends

- class [OsciSocketServI](#)

7.228.1 Detailed Description

PV socket server request queue

7.228.2 Constructor & Destructor Documentation

7.228.2.1 [OsciSocketServRequestList::OsciSocketServRequestList](#) ()

7.228.3 Member Function Documentation

7.228.3.1 void [OsciSocketServRequestList::Add](#) ([OsciSocketRequest](#) *)

7.228.3.2 void [OsciSocketServRequestList::Close](#) ()

7.228.3.3 void [OsciSocketServRequestList::Open](#) ([OsciSocketServI](#) *s)

7.228.3.4 void [OsciSocketServRequestList::Remove](#) ([OsciSocketServRequestQElem](#) * *aElem*)
[inline]

7.228.3.5 void [OsciSocketServRequestList::StartCancel](#) ([OsciSocketRequest](#) *)

7.228.3.6 void [OsciSocketServRequestList::WaitOnRequests](#) ()

7.228.3.7 void [OsciSocketServRequestList::Wakeup](#) ()

7.228.4 Friends And Related Function Documentation

7.228.4.1 friend class [OsciSocketServI](#) [friend]

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

- [oscl_socket_serv_imp_reqlist.h](#)

7.229 OsciSocketServRequestQElem Class Reference

```
#include <osci_socket_serv_imp_reqlist.h>
```

Public Methods

- [OsciSocketServRequestQElem](#) ([OsciSocketRequest](#) *r)

Data Fields

- [OsciSocketRequest](#) * iSocketRequest
- uint8 iSelect
- bool iCancel

7.229.1 Constructor & Destructor Documentation

7.229.1.1 [OsciSocketServRequestQElem::OsciSocketServRequestQElem](#) ([OsciSocketRequest](#) * r)
[inline]

7.229.2 Field Documentation

7.229.2.1 bool [OsciSocketServRequestQElem::iCancel](#)

7.229.2.2 uint8 [OsciSocketServRequestQElem::iSelect](#)

7.229.2.3 [OsciSocketRequest](#)* [OsciSocketServRequestQElem::iSocketRequest](#)

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

- [osci_socket_serv_imp_reqlist.h](#)

7.230 OsclSocketTOS Class Reference

```
#include <oscl_socket_types.h>
```

Public Types

- enum `TPVServicePrecedence` { `EPVRoutine` = 0, `EPVPriority` = 1, `EPVImmediate` = 2, `EPVFlash` = 3, `EPVOverrideFlash` = 4, `EPVCritic_Ecp` = 5, `EPVInetControl` = 6, `EPVNetControl` = 7 }
- enum `TPVServicePriority` { `EPVNoTOS` = 0x0, `EPVLDelay` = (1 << 4), `EPVHiThrpt` = (1 << 3), `EPVHiRel` = (1 << 2) }

Public Methods

- `OsclSocketTOS` ()
- void `SetPrecedence` (`TPVServicePrecedence` aPrecedence)
- void `SetPriority` (bool aMinimizeDelay, bool aMaximizeThroughput, bool MaximizeReliability)
- void `ClearTOS` ()
- uint8 `GetTOS` () const

7.230.1 Member Enumeration Documentation

7.230.1.1 enum OsclSocketTOS::TPVServicePrecedence

Enumeration values:

`EPVRoutine`
`EPVPriority`
`EPVImmediate`
`EPVFlash`
`EPVOverrideFlash`
`EPVCritic_Ecp`
`EPVInetControl`
`EPVNetControl`

7.230.1.2 enum OsclSocketTOS::TPVServicePriority

Enumeration values:

`EPVNoTOS`
`EPVLDelay`
`EPVHiThrpt`
`EPVHiRel`

7.230.2 Constructor & Destructor Documentation

7.230.2.1 `OsclSocketTOS::OsclSocketTOS ()` [inline]

7.230.3 Member Function Documentation

7.230.3.1 `void OsclSocketTOS::ClearTOS ()` [inline]

7.230.3.2 `uint8 OsclSocketTOS::GetTOS () const` [inline]

7.230.3.3 `void OsclSocketTOS::SetPrecedence (TPVServicePrecedence aPrecedence)` [inline]

7.230.3.4 `void OsclSocketTOS::SetPriority (bool aMinimizeDelay, bool aMaximizeThroughput, bool MaximizeReliability)` [inline]

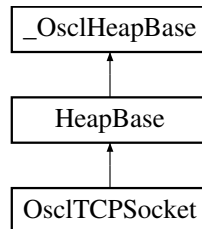
The documentation for this class was generated from the following file:

- [oscl_socket_types.h](#)

7.231 OsciTCPSocket Class Reference

```
#include <osci_socket.h>
```

Inheritance diagram for OsciTCPSocket::



Public Methods

- OSCL_IMPORT_REF [~OsciTCPSocket](#) ()
- OSCL_IMPORT_REF [TPVSocketEvent ThreadLogoff](#) ()
- OSCL_IMPORT_REF [TPVSocketEvent ThreadLogon](#) ([OsciSocketServ](#) &aServ, [OsciSocketObserver](#) *aObserver)
- OSCL_IMPORT_REF [int32 Close](#) ()
- OSCL_IMPORT_REF [int32 Bind](#) ([OsciNetworkAddress](#) &aAddress)
- OSCL_IMPORT_REF [TPVSocketEvent BindAsync](#) ([OsciNetworkAddress](#) &aAddress, [int32](#) aTimeoutMsec=(-1))
- OSCL_IMPORT_REF [void CancelBind](#) ()
- OSCL_IMPORT_REF [int32 SetOptionToReuseAddress](#) ()
- OSCL_IMPORT_REF [int32 SetTOS](#) (const [OsciSocketTOS](#) &aTOS)
- OSCL_IMPORT_REF [int32 GetPeerName](#) ([OsciNetworkAddress](#) &aPeerName)
- 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 [OsciTCPSocket * GetAcceptedSocketL](#) ([uint32](#) aId)
- OSCL_IMPORT_REF [uint8 * GetRecvData](#) ([int32](#) *aLength)
- OSCL_IMPORT_REF [uint8 * GetSendData](#) ([int32](#) *aLength)
- OSCL_IMPORT_REF [TPVSocketEvent Connect](#) ([OsciNetworkAddress](#) &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 OsciTCP socket * [NewL](#) ([OsciDefAlloc](#) &alloc, [OsciSocketServ](#) &aServ, [OsciSocketObserver](#) *aObserver, uint32 aId)

7.231.1 Detailed Description

The TCP Socket class

7.231.2 Constructor & Destructor Documentation

7.231.2.1 OSCL_IMPORT_REF OsciTCP socket::~~OsciTCP socket ()

Destructor. The object must be deleted using the same allocator used in the NewL call.

7.231.3 Member Function Documentation

7.231.3.1 OSCL_IMPORT_REF [TPVSocketEvent](#) OsciTCP socket::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.

7.231.3.2 OSCL_IMPORT_REF int32 OsciTCP socket::Bind ([OsciNetworkAddress](#) & aAddress)

Bind a TCP socket to an address. This is a synchronous method.

Parameters:

aAddress: Bind address.

Returns:

Returns OsciErrNone for success, or a platform-specific error code.

7.231.3.3 OSCL_IMPORT_REF [TPVSocketEvent](#) OsciTCP socket::BindAsync ([OsciNetworkAddress](#) & 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.

7.231.3.4 OSCL_IMPORT_REF void OslTCPSocket::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.

7.231.3.5 OSCL_IMPORT_REF void OslTCPSocket::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.

7.231.3.6 OSCL_IMPORT_REF void OslTCPSocket::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.

7.231.3.7 OSCL_IMPORT_REF void OslTCPSocket::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.

7.231.3.8 OSCL_IMPORT_REF void OslTCPSocket::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.

7.231.3.9 OSCL_IMPORT_REF void OslTCPSocket::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.

7.231.3.10 OSCL_IMPORT_REF void OsciTCPSocket::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.

7.231.3.11 OSCL_IMPORT_REF int32 OsciTCPSocket::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 OsciErrNone for success, or a platform-specific error code.

7.231.3.12 OSCL_IMPORT_REF TPVSocketEvent OsciTCPSocket::Connect (OsciNetworkAddress & 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.

7.231.3.13 OSCL_IMPORT_REF OsciTCPSocket* OsciTCPSocket::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.

7.231.3.14 OSCL_IMPORT_REF int32 OsciTCPSocket::GetPeerName (OsciNetworkAddress & aPeerName)

Retrieves the peer address of the socket

Parameters:

aPeerName: This will store the peer address when API returns successfully.

Returns:

Returns OsciErrNone for success, or a platform-specific error code. or PVSOCK_ERR_NOT_IMPLEMENTED, if this API is not implemented in OSCL for the underlying OS

7.231.3.15 OSCL_IMPORT_REF uint8* OsciTCPSocket::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.

7.231.3.16 OSCL_IMPORT_REF uint8* OsciTCPSocket::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.

7.231.3.17 OSCL_IMPORT_REF int32 OsciTCPSocket::Listen (int32 aQueueSize)

Listen. This is a synchronous method.

Parameters:

aQueueSize: Queue size.

Returns:

Returns OsciErrNone for success, or a platform-specific error code.

7.231.3.18 OSCL_IMPORT_REF TPVSocketEvent OsciTCPSocket::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.

7.231.3.19 `OSCL_IMPORT_REF OslTCPSocket* OslTCPSocket::NewL (Osl_DefAlloc & alloc, OslSocketServ & aServ, OslSocketObserver * 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.

7.231.3.20 `OSCL_IMPORT_REF TPVSocketEvent OslTCPSocket::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.

7.231.3.21 `OSCL_IMPORT_REF TPVSocketEvent OslTCPSocket::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.

7.231.3.22 OSCL_IMPORT_REF int32 OsciTCPSocket::SetOptionToReuseAddress ()

Allows the server to bind to an address which is in a TIME_WAIT state.

Returns:

Returns: OsciErrNone for success, or a platform-specific error code. or PVSOCK_ERR_NOT_SUPPORTED, if underlying OS doesn't support joining multicast group PVSOCK_ERR_BAD_PARAM, if config io file is not configured in accordance with underlying OS PVSOCK_ERR_NOT_IMPLEMENTED, if this API is not implemented in OSCL for the underlying OS

7.231.3.23 OSCL_IMPORT_REF int32 OsciTCPSocket::SetTOS (const OsciSocketTOS & aTOS)

Sets the Type of Service field of each outgoing IP datagram.

Parameters:

aTOS: Specifies the type of service requested.

Returns:

Returns: OsciErrNone for success, or a platform-specific error code. or PVSOCK_ERR_NOT_SUPPORTED, if underlying OS doesn't support joining multicast group PVSOCK_ERR_BAD_PARAM, if config io file is not configured in accordance with underlying OS PVSOCK_ERR_NOT_IMPLEMENTED, if this API is not implemented in OSCL for the underlying OS

7.231.3.24 OSCL_IMPORT_REF TPVSocketEvent OsciTCPSocket::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.

7.231.3.25 OSCL_IMPORT_REF TPVSocketEvent OsciTCPSocket::ThreadLogoff ()

Thread logoff routine. This will prepare for transfer and use of the socket in another thread. All socket requests must be complete prior to calling this routine. If any requests are still active, it will return EPVSocketFailure;

7.231.3.26 OSCL_IMPORT_REF TPVSocketEvent OsciTCPSocket::ThreadLogon (OsciSocketServ & aServ, OsciSocketObserver * aObserver)

Thread logon routine. This will complete the transfer of a socket from another thread for use in the current thread. The ThreadLogoff API must be called in the original thread prior to calling ThreadLogon.

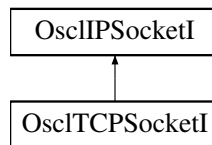
The documentation for this class was generated from the following file:

- [oscl_socket.h](#)

7.232 OsciTCPsSocketI Class Reference

```
#include <osci_tcp_socket.h>
```

Inheritance diagram for OsciTCPsSocketI::



Public Methods

- virtual `~OsciTCPsSocketI ()`
- `TPVSocketEvent ThreadLogoff ()`
- `TPVSocketEvent ThreadLogon (OsciSocketServI *aServ, OsciSocketObserver *aObserver)`
- `int32 Close ()`
- `int32 Listen (int aQueueSize)`
- `OsciTCPsSocketI * GetAcceptedSocketL (uint32 aId)`
- `uint8 * GetRecvData (int32 *aLength)`
- `uint8 * GetSendData (int32 *aLength)`
- `TPVSocketEvent BindAsync (OsciNetworkAddress &aAddress, int32 aTimeoutMsec=-1)`
- `void CancelBind ()`
- `TPVSocketEvent ListenAsync (uint32 qsize, int32 aTimeoutMsec=-1)`
- `void CancelListen ()`
- `TPVSocketEvent Connect (OsciNetworkAddress &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

- `OsciTCPsSocketI * NewL (Osci_DefAlloc &a, OsciSocketServI *aServ, OsciSocketObserver *a-Observer, uint32 aId)`

7.232.1 Detailed Description

Internal implementation class for `OsciTCPsSocket`

7.232.2 Constructor & Destructor Documentation

7.232.2.1 `virtual OsciTCPsocketI::~OsciTCPsocketI ()` [virtual]

7.232.3 Member Function Documentation

7.232.3.1 **TPVSocketEvent** `OsciTCPsocketI::Accept (int32 aTimeout = -1)` [inline]

7.232.3.2 **TPVSocketEvent** `OsciTCPsocketI::BindAsync (OsciNetworkAddress & aAddress, int32 aTimeoutMsec = -1)` [inline]

7.232.3.3 `void OsciTCPsocketI::CancelAccept ()` [inline]

7.232.3.4 `void OsciTCPsocketI::CancelBind ()` [inline]

7.232.3.5 `void OsciTCPsocketI::CancelConnect ()` [inline]

7.232.3.6 `void OsciTCPsocketI::CancelListen ()` [inline]

7.232.3.7 `void OsciTCPsocketI::CancelRecv ()` [inline]

7.232.3.8 `void OsciTCPsocketI::CancelSend ()` [inline]

7.232.3.9 `void OsciTCPsocketI::CancelShutdown ()` [inline]

7.232.3.10 `int32 OsciTCPsocketI::Close ()` [virtual]

Implements [OsciIPSocketI](#).

7.232.3.11 **TPVSocketEvent** `OsciTCPsocketI::Connect (OsciNetworkAddress & aAddress, int32 aTimeoutMsec = -1)` [inline]

7.232.3.12 `OsciTCPsocketI* OsciTCPsocketI::GetAcceptedSocketL (uint32 aId)`

7.232.3.13 `uint8 * OsciTCPsocketI::GetRecvData (int32 * aLength)` [inline, virtual]

Implements [OsciIPSocketI](#).

7.232.3.14 `uint8 * OsciTCPsocketI::GetSendData (int32 * aLength)` [inline, virtual]

Implements [OsciIPSocketI](#).

- 7.232.3.15 `int32 OslTCPSocketI::Listen (int aQueueSize) [inline]`
- 7.232.3.16 `TPVSocketEvent OslTCPSocketI::ListenAsync (uint32 qsize, int32 aTimeoutMsec = -1) [inline]`
- 7.232.3.17 `OslTCPSocketI* OslTCPSocketI::NewL (Osl_DefAlloc & a, OslSocketServI * aServ, OslSocketObserver * aObserver, uint32 aId) [static]`
- 7.232.3.18 `TPVSocketEvent OslTCPSocketI::Recv (uint8 *& aPtr, uint32 aMaxLen, int32 aTimeoutMsec = -1) [inline]`
- 7.232.3.19 `TPVSocketEvent OslTCPSocketI::Send (const uint8 *& aPtr, uint32 aLen, int32 aTimeoutMsec = -1) [inline]`
- 7.232.3.20 `TPVSocketEvent OslTCPSocketI::Shutdown (TPVSocketShutdown aHow, int32 aTimeoutMsec = -1) [inline]`
- 7.232.3.21 `TPVSocketEvent OslTCPSocketI::ThreadLogoff ()`

Reimplemented from [OslIPSocketI](#).

- 7.232.3.22 `TPVSocketEvent OslTCPSocketI::ThreadLogon (OslSocketServI * aServ, OslSocketObserver * aObserver)`

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

- [oscl_tcp_socket.h](#)

7.233 OsciThread Class Reference

```
#include <osci_thread.h>
```

Public Methods

- OSCL_IMPORT_REF [OsciThread \(\)](#)
- OSCL_IMPORT_REF [~OsciThread \(\)](#)
- OSCL_IMPORT_REF [OsciProcStatus::eOsciProcError Create \(TOsciThreadFuncPtr func, int32 stack_size, TOsciThreadFuncArg argument, OsciThread_State state=Start_on_creation, bool oIsJoinable=false\)](#)
- OSCL_IMPORT_REF [OsciProcStatus::eOsciProcError GetPriority \(OsciThreadPriority &refThreadPriority\)](#)
- OSCL_IMPORT_REF [OsciProcStatus::eOsciProcError SetPriority \(OsciThreadPriority ePriority\)](#)
- OSCL_IMPORT_REF [OsciProcStatus::eOsciProcError Suspend \(\)](#)
- OSCL_IMPORT_REF [OsciProcStatus::eOsciProcError Resume \(\)](#)
- OSCL_IMPORT_REF [OsciProcStatus::eOsciProcError Terminate \(OsciAny *exitcode\)](#)

Static Public Methods

- OSCL_IMPORT_REF void [Exit \(OsciAny *exitcode\)](#)
- OSCL_IMPORT_REF void [EnableKill \(\)](#)
- OSCL_IMPORT_REF [OsciProcStatus::eOsciProcError GetId \(TOsciThreadId &refThreadId\)](#)
- OSCL_IMPORT_REF bool [CompareId \(TOsciThreadId &t1, TOsciThreadId &t2\)](#)
- OSCL_IMPORT_REF void [SleepMillisec \(const int32 msec\)](#)

7.233.1 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.

7.233.2 Constructor & Destructor Documentation

7.233.2.1 OSCL_IMPORT_REF OsciThread::OsciThread ()

Class constructor

7.233.2.2 OSCL_IMPORT_REF OsciThread::~~OsciThread ()

Class destructor

7.233.3 Member Function Documentation

7.233.3.1 OSCL_IMPORT_REF bool OsciThread::CompareId (TOsciThreadId & t1, TOsciThreadId & 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.

7.233.3.2 OSCL_IMPORT_REF OsciProcStatus::eOsciProcError OsciThread::Create (TOsciThreadFuncPtr *func*, int32 *stack_size*, TOsciThreadFuncArg *argument*, OsciThread_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:

eOsciProcError

7.233.3.3 OSCL_IMPORT_REF void OsciThread::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

7.233.3.4 OSCL_IMPORT_REF void OsciThread::Exit (OsciAny * *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

**7.233.3.5 OSCL_IMPORT_REF OsciProcStatus::eOsciProcError OsciThread::GetId
(TOsciThreadId & refThreadId) [static]**

Static routine to retrieve ID of calling thread.

Parameters:

Thread ID passed by the application

Returns:

Error code

**7.233.3.6 OSCL_IMPORT_REF OsciProcStatus::eOsciProcError OsciThread::GetPriority
(OsciThreadPriority & 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

7.233.3.7 OSCL_IMPORT_REF OsciProcStatus::eOsciProcError OsciThread::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.

**7.233.3.8 OSCL_IMPORT_REF OsciProcStatus::eOsciProcError OsciThread::SetPriority
(OsciThreadPriority 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.

7.233.3.9 OSCL_IMPORT_REF void OsciThread::SleepMillisec (const int32 msec) [static]

Suspend current thread execution for specified time.

Parameters:

msec, t2: sleep time in milliseconds.

7.233.3.10 OSCL_IMPORT_REF OsciProcStatus::eOsciProcError OsciThread::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.

7.233.3.11 OSCL_IMPORT_REF OsciProcStatus::eOsciProcError OsciThread::Terminate (OsciAny * 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

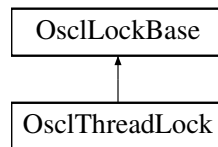
The documentation for this class was generated from the following file:

- [oscl_thread.h](#)

7.234 OsciThreadLock Class Reference

```
#include <osci_mutex.h>
```

Inheritance diagram for OsciThreadLock::



Public Methods

- OSCL_IMPORT_REF [OsciThreadLock \(\)](#)
- virtual OSCL_IMPORT_REF [~OsciThreadLock \(\)](#)
- OSCL_IMPORT_REF void [Lock \(\)](#)
- OSCL_IMPORT_REF void [Unlock \(\)](#)

7.234.1 Detailed Description

An implementation of [OsciLockBase](#) using a mutex

7.234.2 Constructor & Destructor Documentation

7.234.2.1 OSCL_IMPORT_REF [OsciThreadLock::OsciThreadLock \(\)](#)

7.234.2.2 virtual OSCL_IMPORT_REF [OsciThreadLock::~~OsciThreadLock \(\)](#) [virtual]

7.234.3 Member Function Documentation

7.234.3.1 OSCL_IMPORT_REF void [OsciThreadLock::Lock \(\)](#) [virtual]

Implements [OsciLockBase](#).

7.234.3.2 OSCL_IMPORT_REF void [OsciThreadLock::Unlock \(\)](#) [virtual]

Implements [OsciLockBase](#).

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

- [osci_mutex.h](#)

7.235 OsclTickCount Class Reference

```
#include <oscl_tickcount.h>
```

Static Public Methods

- uint32 [TickCount](#) ()
- uint32 [TickCountFrequency](#) ()
- uint32 [TickCountPeriod](#) ()
- uint32 [TicksToMsec](#) (uint32 ticks)
- uint32 [MsecToTicks](#) (uint32 msec)

7.235.1 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.

7.235.2 Member Function Documentation

7.235.2.1 uint32 OsclTickCount::MsecToTicks (uint32 msec) [static]

This function converts milliseconds to ticks

Returns:

ticks

7.235.2.2 uint32 OsclTickCount::TickCount () [static]

This function returns the current system tick count

Returns:

returns the tick count

7.235.2.3 uint32 OsclTickCount::TickCountFrequency () [static]

This function returns the tick frequency in ticks per second

Returns:

ticks per second

7.235.2.4 uint32 OsclTickCount::TickCountPeriod () [static]

This function returns the tick period in microseconds per tick

Returns:

microseconds per tick

7.235.2.5 uint32 OsciTickCount::TicksToMsec (uint32 *ticks*) [static]

This function converts ticks to milliseconds

Returns:

milliseconds

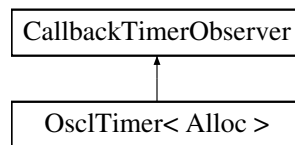
The documentation for this class was generated from the following file:

- [osci_tickcount.h](#)

7.236 `OscTimer< Alloc >` Class Template Reference

```
#include <oscl_timer.h>
```

Inheritance diagram for `OscTimer< Alloc >`:



Public Types

- typedef `CallbackTimer< Alloc >` `callback_timer_type`

Public Methods

- `OscTimer` (`const char *name`, `uint32 frequency=1`, `int32 priority=OscActiveObject::EPriority-Nominal`)
- virtual `~OscTimer` ()
- void `SetObserver` (`OscTimerObserver *obs`)
- void `SetFrequency` (`uint32 frequency`)
- void `SetExactFrequency` (`uint32 frequency`)
- void `Request` (`int32 timerID`, `int32 timeoutInfo`, `int32 cycles`, `OscTimerObserver *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 OsciTimer< Alloc >

7.236.1 Member Typedef Documentation

7.236.1.1 template<class Alloc> typedef [CallbackTimer](#)<Alloc> OsciTimer< Alloc >::callback_timer_type

7.236.2 Constructor & Destructor Documentation

7.236.2.1 template<class Alloc> OsciTimer< Alloc >::OsciTimer (const char * *name*, uint32 *frequency* = 1, int32 *priority* = OsciActiveObject::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.

7.236.2.2 template<class Alloc> OsciTimer< Alloc >::~~OsciTimer () [virtual]

7.236.3 Member Function Documentation

7.236.3.1 template<class Alloc> void OsciTimer< 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.

7.236.3.2 template<class Alloc> void OsciTimer< Alloc >::Clear ()

Cancel all pending timers.

7.236.3.3 template<class Alloc> void OsciTimer< Alloc >::Request (int32 *timerID*, int32 *timeoutInfo*, int32 *cycles*, [OsciTimerObserver](#) * *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 overrides the global observer if set.

7.236.3.4 `template<class Alloc> void OscTimer< 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.

7.236.3.5 `template<class Alloc> void OscTimer< 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.

7.236.3.6 `template<class Alloc> void OscTimer< Alloc >::SetObserver (OscTimerObserver * obs) [inline]`

Set the global observer. Each timer can request a local observer, which if set overrides the global observer.

Parameters:

obs observer object.

7.236.3.7 `template<class Alloc> void OscTimer< Alloc >::TimerBaseElapsed () [protected, virtual]`

Implements [CallbackTimerObserver](#).

7.236.4 Friends And Related Function Documentation**7.236.4.1** `template<class Alloc> friend class CallbackTimer< Alloc > [friend]`

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

- [oscl_timer.h](#)

7.237 OsciTimerCompare Class Reference

```
#include <osci_scheduler_readyq.h>
```

Static Public Methods

- `int compare (TOsciReady &a, TOsciReady &b)`

7.237.1 Member Function Documentation

7.237.1.1 `int OsciTimerCompare::compare (TOsciReady &a, TOsciReady &b)` [static]

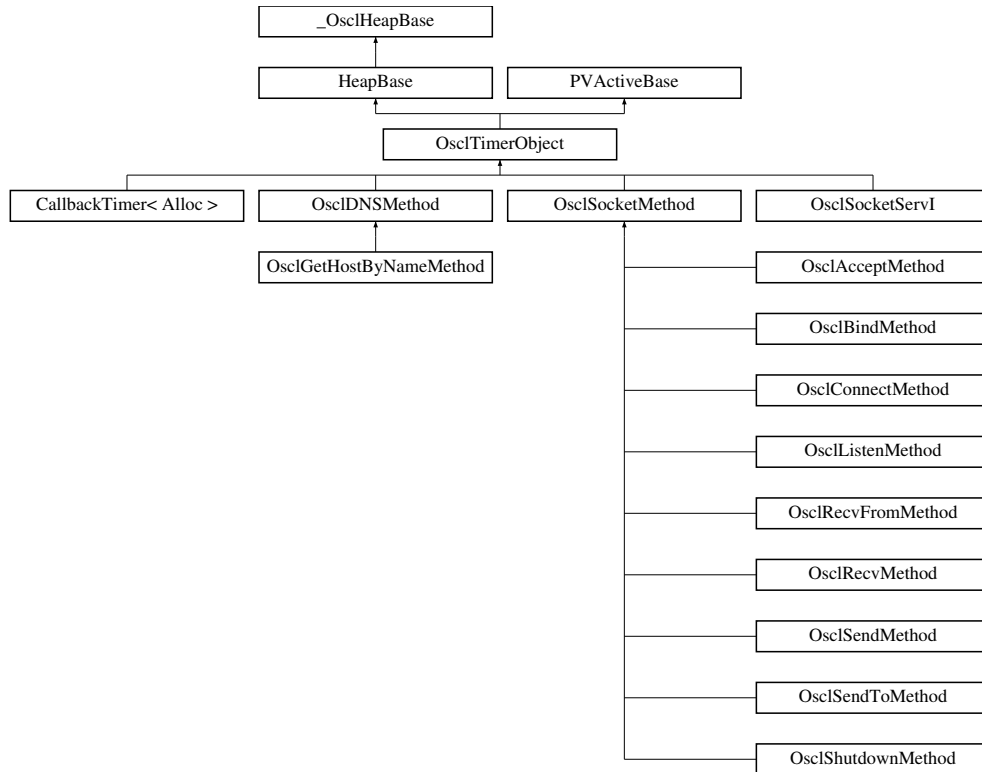
The documentation for this class was generated from the following file:

- [osci_scheduler_readyq.h](#)

7.238 OsciTimerObject Class Reference

```
#include <osci_scheduler_ao.h>
```

Inheritance diagram for OsciTimerObject::



Public Methods

- OSCL_IMPORT_REF [OsciTimerObject](#) (int32 aPriority, const char name[])
- virtual OSCL_IMPORT_REF [~OsciTimerObject](#) ()
- 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 [OsciAOSStatus](#) & [StatusRef](#) ()

Protected Methods

- virtual OSCL_IMPORT_REF void [DoCancel](#) ()
- virtual OSCL_IMPORT_REF int32 [RunError](#) (int32 aError)

7.238.1 Detailed Description

User base class for execution objects. OsciTimerObject defines an exec object with a timer.

7.238.2 Constructor & Destructor Documentation

7.238.2.1 OSCL_IMPORT_REF OsciTimerObject::OsciTimerObject (int32 *aPriority*, const char *name*[])

Constructor.

Parameters:

- aPriority* (input param): scheduling priority
- name* (input param): optional name for this AO.

7.238.2.2 virtual OSCL_IMPORT_REF OsciTimerObject::~~OsciTimerObject () [virtual]

Destructor.

7.238.3 Member Function Documentation

7.238.3.1 OSCL_IMPORT_REF void OsciTimerObject::AddToScheduler ()

Add this AO to the current thread's scheduler.

Reimplemented from [PVActiveBase](#).

7.238.3.2 OSCL_IMPORT_REF void OsciTimerObject::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.

7.238.3.3 OSCL_IMPORT_REF void OsciTimerObject::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](#).

7.238.3.4 `virtual OSCL_IMPORT_REF void OsciTimerObject::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 `OsciTimerObject::DoCancel` in its own DoCancel routine.

Implements [PVActiveBase](#).

7.238.3.5 `OSCL_IMPORT_REF bool OsciTimerObject::IsBusy ()`

Return true if this AO is active, false otherwise.

7.238.3.6 `OSCL_IMPORT_REF int32 OsciTimerObject::Priority ()`

Return scheduling priority of this exec object.

7.238.3.7 `OSCL_IMPORT_REF void OsciTimerObject::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](#).

7.238.3.8 `virtual OSCL_IMPORT_REF int32 OsciTimerObject::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 `OsciErrNone` if it handles the error, otherwise it should return the input error code.

Parameters:

aError: the leave code generated by the Run.

Implements [PVActiveBase](#).

7.238.3.9 `OSCL_IMPORT_REF void OsciTimerObject::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.

7.238.3.10 `OSCL_IMPORT_REF void OsciTimerObject::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.

7.238.3.11 OSCL_IMPORT_REF void OsciTimerObject::SetStatus (int32)

7.238.3.12 OSCL_IMPORT_REF int32 OsciTimerObject::Status ()

Request status access

7.238.3.13 OSCL_IMPORT_REF [OsciAOSStatus&](#) OsciTimerObject::StatusRef ()

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

- [osci_scheduler_ao.h](#)

7.239 OsciTimerObserver Class Reference

```
#include <osci_timer.h>
```

Public Methods

- virtual void [TimeoutOccurred](#) (int32 timerID, int32 timeoutInfo)=0
- virtual [~OsciTimerObserver](#) ()

7.239.1 Detailed Description

The observer class to receive timeout callbacks

7.239.2 Constructor & Destructor Documentation

7.239.2.1 virtual [OsciTimerObserver::~OsciTimerObserver](#) () [inline, virtual]

7.239.3 Member Function Documentation

7.239.3.1 virtual void [OsciTimerObserver::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.

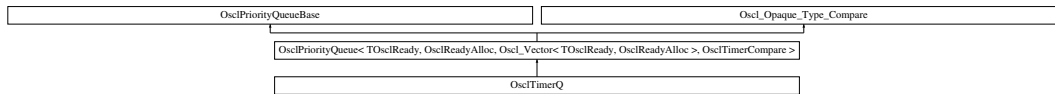
The documentation for this class was generated from the following file:

- [osci_timer.h](#)

7.240 OsciTimerQ Class Reference

```
#include <oscl_scheduler_readyq.h>
```

Inheritance diagram for OsciTimerQ::



Public Methods

- void [Construct](#) (int)
- void [Add](#) ([TOsciReady](#))
- void [Remove](#) ([TOsciReady](#))
- [TOsciReady](#) [PopTop](#) ()
- [TOsciReady](#) [Top](#) ()
- void [Pop](#) ([TOsciReady](#))
- bool [IsIn](#) ([TOsciReady](#))

7.240.1 Member Function Documentation

7.240.1.1 void [OsciTimerQ::Add](#) ([TOsciReady](#))

7.240.1.2 void [OsciTimerQ::Construct](#) (int)

7.240.1.3 bool [OsciTimerQ::IsIn](#) ([TOsciReady](#))

7.240.1.4 void [OsciTimerQ::Pop](#) ([TOsciReady](#))

7.240.1.5 [TOsciReady](#) [OsciTimerQ::PopTop](#) ()

7.240.1.6 void [OsciTimerQ::Remove](#) ([TOsciReady](#))

7.240.1.7 [TOsciReady](#) [OsciTimerQ::Top](#) ()

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

- [oscl_scheduler_readyq.h](#)

7.241 OsciTLS< T, ID, Registry > Class Template Reference

```
#include <osci_tls.h>
```

Public Methods

- `OsciTLS ()`
- `~OsciTLS ()`
- `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 = OsciTLSRegistry> class OsciTLS< T, ID, Registry >
```

7.241.1 Constructor & Destructor Documentation

7.241.1.1 `template<class T, uint32 ID, class Registry = OsciTLSRegistry> OsciTLS< T, ID, Registry >::OsciTLS () [inline]`

7.241.1.2 `template<class T, uint32 ID, class Registry = OsciTLSRegistry> OsciTLS< T, ID, Registry >::~~OsciTLS () [inline]`

7.241.2 Member Function Documentation

7.241.2.1 `template<class T, uint32 ID, class Registry = OsciTLSRegistry> T& OsciTLS< T, ID, Registry >::operator * () const [inline]`

The indirection operator (*) accesses a value indirectly, through a pointer.

This operator ensures that the OsciTLS can be used like the regular pointer that it was initialized with.

7.241.2.2 `template<class T, uint32 ID, class Registry = OsciTLSRegistry> T* OsciTLS< T, ID, Registry >::operator -> () const [inline]`

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the OsciTLS can be used like the regular pointer that it was initialized with.

7.241.2.3 `template<class T, uint32 ID, class Registry = OsciTLSRegistry> bool OsciTLS< 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.

7.241.3 Field Documentation

7.241.3.1 `template<class T, uint32 ID, class Registry = OsciTLSRegistry> T* OsciTLS< T, ID, Registry >::_Ptr [protected]`

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

- [osci_tls.h](#)

7.242 OsciTLSEx< T, ID, Registry > Class Template Reference

```
#include <osci_error.h>
```

Public Methods

- [OsciTLSEx \(\)](#)
- [~OsciTLSEx \(\)](#)
- [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 = OsciTLSRegistryEx> class OsciTLSEx< T, ID, Registry >
```

7.242.1 Constructor & Destructor Documentation

7.242.1.1 `template<class T, uint32 ID, class Registry = OsciTLSRegistryEx> OsciTLSEx< T, ID, Registry >::OsciTLSEx () [inline]`

7.242.1.2 `template<class T, uint32 ID, class Registry = OsciTLSRegistryEx> OsciTLSEx< T, ID, Registry >::~~OsciTLSEx () [inline]`

7.242.2 Member Function Documentation

7.242.2.1 `template<class T, uint32 ID, class Registry = OsciTLSRegistryEx> T& OsciTLSEx< T, ID, Registry >::operator * () const [inline]`

The indirection operator (*) accesses a value indirectly, through a pointer.

This operator ensures that the [OsciTLS](#) can be used like the regular pointer that it was initialized with.

7.242.2.2 `template<class T, uint32 ID, class Registry = OsciTLSRegistryEx> T* OsciTLSEx< T, ID, Registry >::operator → () const [inline]`

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the [OsciTLS](#) can be used like the regular pointer that it was initialized with.

7.242.2.3 `template<class T, uint32 ID, class Registry = OsciTLSRegistryEx> bool OsciTLSEx< 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.

7.242.3 Field Documentation

7.242.3.1 `template<class T, uint32 ID, class Registry = OsciTLSRegistryEx> T* OsciTLSEx< T, ID, Registry >::_Ptr [protected]`

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

- [oscl_error.h](#)

7.243 OsciTLSRegistry Class Reference

```
#include <osci_tls.h>
```

Static Public Methods

- OSCL_IMPORT_REF [OsciAny](#) * [getInstance](#) (uint32 ID, int32 &error)
- OSCL_IMPORT_REF void [registerInstance](#) ([OsciAny](#) *ptr, uint32 ID, int32 &error)

Friends

- class [OsciBase](#)

7.243.1 Member Function Documentation

7.243.1.1 OSCL_IMPORT_REF [OsciAny](#)* [OsciTLSRegistry::getInstance](#) (uint32 *ID*, int32 & *error*) [static]

7.243.1.2 OSCL_IMPORT_REF void [OsciTLSRegistry::registerInstance](#) ([OsciAny](#) * *ptr*, uint32 *ID*, int32 & *error*) [static]

7.243.2 Friends And Related Function Documentation

7.243.2.1 friend class [OsciBase](#) [friend]

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

- [osci_tls.h](#)

7.244 OsciTLSRegistryEx Class Reference

```
#include <osci_error.h>
```

Static Public Methods

- [OsciAny](#) * [getInstance](#) (uint32 ID)
- void [registerInstance](#) ([OsciAny](#) *ptr, uint32 ID)

7.244.1 Member Function Documentation

7.244.1.1 [OsciAny](#)* [OsciTLSRegistryEx::getInstance](#) (uint32 ID) [inline, static]

7.244.1.2 void [OsciTLSRegistryEx::registerInstance](#) ([OsciAny](#) * ptr, uint32 ID) [inline, static]

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

- [osci_error.h](#)

7.245 OsciTrapItem Class Reference

```
#include <osci_heapbase.h>
```

Public Methods

- OSCL_INLINE [OsciTrapItem](#) ([OsciTrapOperation](#) anOperation)
- OSCL_INLINE [OsciTrapItem](#) ([OsciTrapOperation](#) anOperation, [OsciAny](#) *aPtr)

Friends

- class [OsciTrapStackItem](#)
- class [OsciTrapStack](#)

7.245.1 Constructor & Destructor Documentation

7.245.1.1 OSCL_INLINE [OsciTrapItem::OsciTrapItem](#) ([OsciTrapOperation](#) *anOperation*)

7.245.1.2 OSCL_INLINE [OsciTrapItem::OsciTrapItem](#) ([OsciTrapOperation](#) *anOperation*, [OsciAny](#) * *aPtr*)

7.245.2 Friends And Related Function Documentation

7.245.2.1 friend class [OsciTrapStack](#) [friend]

7.245.2.2 friend class [OsciTrapStackItem](#) [friend]

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

- [osci_heapbase.h](#)

7.246 OsciTrapStack Class Reference

```
#include <osci_error_trapcleanup.h>
```

Friends

- class [OsciError](#)
- class [OsciErrorTrap](#)
- class [OsciErrorTrapImp](#)

7.246.1 Detailed Description

A common type for cleanup stack and trap mark stack. for internal use only.

7.246.2 Friends And Related Function Documentation

7.246.2.1 friend class [OsciError](#) [friend]

7.246.2.2 friend class [OsciErrorTrap](#) [friend]

7.246.2.3 friend class [OsciErrorTrapImp](#) [friend]

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

- [osci_error_trapcleanup.h](#)

7.247 OslTrapStackItem Class Reference

```
#include <oscl_error_trapcleanup.h>
```

Public Methods

- [OslTrapStackItem \(\)](#)
- [OslTrapStackItem \(_OslHeapBase *aCBase\)](#)
- [OslTrapStackItem \(OslAny *aTAny\)](#)
- [OslTrapStackItem \(OslTrapItem aItem\)](#)

Data Fields

- [_OslHeapBase * iCBase](#)
- [OslAny * iTAny](#)
- [OslTrapOperation iTrapOperation](#)
- [OslTrapStackItem * iNext](#)

7.247.1 Detailed Description

Internal cleanup stack item type.

7.247.2 Constructor & Destructor Documentation

7.247.2.1 [OslTrapStackItem::OslTrapStackItem \(\)](#) [inline]

7.247.2.2 [OslTrapStackItem::OslTrapStackItem \(_OslHeapBase * aCBase\)](#) [inline]

7.247.2.3 [OslTrapStackItem::OslTrapStackItem \(OslAny * aTAny\)](#) [inline]

7.247.2.4 [OslTrapStackItem::OslTrapStackItem \(OslTrapItem aItem\)](#) [inline]

7.247.3 Field Documentation

7.247.3.1 [_OslHeapBase*](#) [OslTrapStackItem::iCBase](#)

7.247.3.2 [OslTrapStackItem*](#) [OslTrapStackItem::iNext](#)

7.247.3.3 [OslAny*](#) [OslTrapStackItem::iTAny](#)

7.247.3.4 [OslTrapOperation](#) [OslTrapStackItem::iTrapOperation](#)

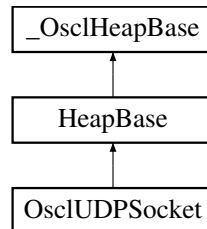
The documentation for this class was generated from the following file:

- [oscl_error_trapcleanup.h](#)

7.248 OsciUDPSocket Class Reference

```
#include <osci_socket.h>
```

Inheritance diagram for OsciUDPSocket::



Public Methods

- OSCL_IMPORT_REF [~OsciUDPSocket](#) ()
- OSCL_IMPORT_REF [TPVSocketEvent ThreadLogoff](#) ()
- OSCL_IMPORT_REF [TPVSocketEvent ThreadLogon](#) ([OsciSocketServ](#) &aServ, [OsciSocketObserver](#) *aObserver)
- OSCL_IMPORT_REF int32 [Close](#) ()
- OSCL_IMPORT_REF int32 [Bind](#) ([OsciNetworkAddress](#) &aAddress)
- OSCL_IMPORT_REF int32 [Join](#) ([OsciNetworkAddress](#) &aAddress)
- OSCL_IMPORT_REF int32 [JoinMulticastGroup](#) ([OsciIpMReq](#) &aMReq)
- OSCL_IMPORT_REF int32 [SetMulticastTTL](#) (int32 aTTL)
- OSCL_IMPORT_REF int32 [SetOptionToReuseAddress](#) ()
- OSCL_IMPORT_REF int32 [SetTOS](#) (const [OsciSocketTOS](#) &aTOS)
- OSCL_IMPORT_REF int32 [GetPeerName](#) ([OsciNetworkAddress](#) &aPeerName)
- OSCL_IMPORT_REF [TPVSocketEvent BindAsync](#) ([OsciNetworkAddress](#) &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, [OsciNetworkAddress](#) &aAddress, int32 aTimeoutMsec=-1)
- OSCL_IMPORT_REF void [CancelSendTo](#) ()
- OSCL_IMPORT_REF [TPVSocketEvent RecvFrom](#) (uint8 *aPtr, uint32 aMaxLen, [OsciNetworkAddress](#) &aAddress, int32 aTimeoutMsec=-1, uint32 aMultiRecvLimit=0, [Osci_Vector](#)< uint32, [OsciMemAllocator](#) > *aPacketLen=NULL, [Osci_Vector](#)< [OsciNetworkAddress](#), [OsciMemAllocator](#) > *aPacketSource=NULL)
- OSCL_IMPORT_REF void [CancelRecvFrom](#) ()
- OSCL_IMPORT_REF int32 [SetRecvBufferSize](#) (uint32 size)

Static Public Methods

- OSCL_IMPORT_REF [OsciUDPSocket *](#) [NewL](#) ([Osci_DefAlloc](#) &alloc, [OsciSocketServ](#) &aServ, [OsciSocketObserver](#) *aObserver, uint32 aId)

7.248.1 Detailed Description

The UDP Socket class

7.248.2 Constructor & Destructor Documentation

7.248.2.1 OSCL_IMPORT_REF OsciUDPSocket::~~OsciUDPSocket ()

Destructor. The object must be deleted using the same allocator used in the NewL call.

7.248.3 Member Function Documentation

7.248.3.1 OSCL_IMPORT_REF int32 OsciUDPSocket::Bind (OsciNetworkAddress & aAddress)

Bind a UDP socket to an address. This is a synchronous method.

Parameters:

aAddress: Bind address.

Returns:

Returns OsciErrNone for success, or a platform-specific error code.

7.248.3.2 OSCL_IMPORT_REF TPVSocketEvent OsciUDPSocket::BindAsync (OsciNetworkAddress & 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.

7.248.3.3 OSCL_IMPORT_REF void OsciUDPSocket::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.

7.248.3.4 OSCL_IMPORT_REF void OsciUDPSocket::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.

7.248.3.5 OSCL_IMPORT_REF void OsciUDPSocket::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.

7.248.3.6 OSCL_IMPORT_REF int32 OsciUDPSocket::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 OsciErrNone for success, or a platform-specific error code.

7.248.3.7 OSCL_IMPORT_REF int32 OsciUDPSocket::GetPeerName (OsciNetworkAddress & aPeerName)

Retrieves the peer address of the socket

Parameters:

aPeerName: This will store the peer address when API returns successfully.

Returns:

Returns OsciErrNone for success, or a platform-specific error code. or PVSOCK_ERR_NOT_IMPLEMENTED, if this API is not implemented in OSCL for the underlying OS

7.248.3.8 OSCL_IMPORT_REF uint8* OsciUDPSocket::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.

7.248.3.9 OSCL_IMPORT_REF uint8* OsciUDPSocket::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.

7.248.3.10 OSCL_IMPORT_REF int32 OsciUDPSocket::Join (OsciNetworkAddress & aAddress)

Bind a UDP socket to an address and Join the multicast group. This is a synchronous method.

Parameters:

aAddress: Bind address.

Returns:

Returns OsciErrNone for success, or a platform-specific error code. May throw an OsciErrNotSupported Exception

7.248.3.11 OSCL_IMPORT_REF int32 OsciUDPSocket::JoinMulticastGroup (OsciIpMReq & aMReq)

Join the multicast group.

Parameters:

aMReq: Multicast group information.

Returns:

Returns: OsciErrNone for success, or a platform-specific error code. or PVSOCK_ERR_NOT_SUPPORTED, if underlying OS doesn't support joining multicast group PVSOCK_ERR_BAD_PARAM, if config io file is not configured in accordance with underlying OS PVSOCK_ERR_NOT_IMPLEMENTED, if this API is not implemented in OSCL for the underlying OS

7.248.3.12 OSCL_IMPORT_REF OsciUDPSocket* OsciUDPSocket::NewL (Osci_DefAlloc & alloc, OsciSocketServ & aServ, OsciSocketObserver * 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.

7.248.3.13 OSCL_IMPORT_REF TPVSocketEvent OsciUDPSocket::RecvFrom (uint8 * aPtr, uint32 aMaxLen, OsciNetworkAddress & aAddress, int32 aTimeoutMsec = -1, uint32 aMultiRecvLimit = 0, Osci_Vector< uint32, OsciMemAllocator > * aPacketLen = NULL, Osci_Vector< OsciNetworkAddress, OsciMemAllocator > * 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.

7.248.3.14 OSCL_IMPORT_REF TPVSocketEvent OsciUDPSocket::SendTo (const uint8 * *aPtr*, uint32 *aLen*, OsciNetworkAddress & *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.

7.248.3.15 OSCL_IMPORT_REF int32 OsciUDPSocket::SetMulticastTTL (int32 *aTTL*)

Controls the number of intermediate systems through which a multicast datagram can be forwarded.

Parameters:

- aTTL:Specifies*** the time-to-live value for multicast datagrams sent through this socket.

Returns:

Returns: OsciErrNone for success, or a platform-specific error code. or PVSOCK_ERR_NOT_SUPPORTED, if underlying OS doesn't support joining multicast group PVSOCK_ERR_BAD_PARAM, if config io file is not configured in accordance with underlying OS PVSOCK_ERR_NOT_IMPLEMENTED, if this API is not implemented in OSCL for the underlying OS

7.248.3.16 OSCL_IMPORT_REF int32 OsciUDPSocket::SetOptionToReuseAddress ()

Allows the server to bind to an address which is in a TIME_WAIT state.

Returns:

Returns: OsciErrNone for success, or a platform-specific error code. or PVSOCK_ERR_NOT_SUPPORTED, if underlying OS doesn't support joining multicast group PVSOCK_ERR_BAD_PARAM, if config io file is not configured in accordance with underlying OS PVSOCK_ERR_NOT_IMPLEMENTED, if this API is not implemented in OSCL for the underlying OS

7.248.3.17 OSCL_IMPORT_REF int32 OsciUDPSocket::SetRecvBufferSize (uint32 size)

Set the buffer size of the socket This is a synchronous method.

Parameters:

size: buffer size

Returns:

Returns OsciErrNone for success, or a platform-specific error code. May throw an OsciErrNotSupported Exception.

7.248.3.18 OSCL_IMPORT_REF int32 OsciUDPSocket::SetTOS (const OsciSocketTOS & aTOS)

Sets the Type of Service field of each outgoing IP datagram.

Parameters:

aTOS: Specifies the type of service requested.

Returns:

Returns: OsciErrNone for success, or a platform-specific error code. or PVSOCK_ERR_NOT_SUPPORTED, if underlying OS doesn't support joining multicast group PVSOCK_ERR_BAD_PARAM, if config io file is not configured in accordance with underlying OS PVSOCK_ERR_NOT_IMPLEMENTED, if this API is not implemented in OSCL for the underlying OS

7.248.3.19 OSCL_IMPORT_REF TPVSocketEvent OsciUDPSocket::ThreadLogoff ()

Thread logoff routine. This will prepare for transfer and use of the socket in another thread. All socket requests must be complete prior to calling this routine. If any requests are still active, it will return EPVSocketFailure;

7.248.3.20 OSCL_IMPORT_REF TPVSocketEvent OsciUDPSocket::ThreadLogon (OsciSocketServ & aServ, OsciSocketObserver * aObserver)

Thread logon routine. This will complete the transfer of a socket from another thread for use in the current thread. The ThreadLogoff API must be called in the original thread prior to calling ThreadLogon.

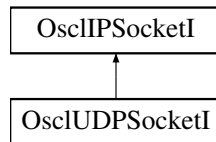
The documentation for this class was generated from the following file:

- [oscl_socket.h](#)

7.249 OsclUDPSocketI Class Reference

```
#include <oscl_udp_socket.h>
```

Inheritance diagram for OsclUDPSocketI::



Public Methods

- virtual `~OsclUDPSocketI ()`
- int32 `Close ()`
- int32 `JoinMulticastGroup (OsclIpMReq &aMReq)`
- int32 `SetMulticastTTL (int32 aTTL)`
- uint8 * `GetRecvData (int32 *aLength)`
- uint8 * `GetSendData (int32 *aLength)`
- `TPVSocketEvent ThreadLogoff ()`
- `TPVSocketEvent ThreadLogon (OsclSocketServI *aServ, OsclSocketObserver *aObserver)`
- `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)`

7.249.1 Detailed Description

Internal implementation class for `OsclUDPSocket`

7.249.2 Constructor & Destructor Documentation

7.249.2.1 `virtual OsciUDPSocketI::~~OsciUDPSocketI ()` [virtual]

7.249.3 Member Function Documentation

7.249.3.1 **TPVSocketEvent** `OsciUDPSocketI::BindAsync (OsciNetworkAddress & aAddress, int32 aTimeoutMsec = -1)` [inline]

7.249.3.2 `void OsciUDPSocketI::CancelBind ()` [inline]

7.249.3.3 `void OsciUDPSocketI::CancelRecvFrom ()` [inline]

7.249.3.4 `void OsciUDPSocketI::CancelSendTo ()` [inline]

7.249.3.5 `int32 OsciUDPSocketI::Close ()` [virtual]

Implements [OsciIPSocketI](#).

7.249.3.6 `uint8 * OsciUDPSocketI::GetRecvData (int32 * aLength)` [inline, virtual]

Implements [OsciIPSocketI](#).

7.249.3.7 `uint8 * OsciUDPSocketI::GetSendData (int32 * aLength)` [inline, virtual]

Implements [OsciIPSocketI](#).

7.249.3.8 `int32 OsciUDPSocketI::JoinMulticastGroup (OsciIpMReq & aMReq)`

7.249.3.9 `OsciUDPSocketI* OsciUDPSocketI::NewL (Osci_DefAlloc & a, OsciSocketServI * aServ, OsciSocketObserver * aObserver, uint32 aId)` [static]

7.249.3.10 **TPVSocketEvent** `OsciUDPSocketI::RecvFrom (uint8 *& aPtr, uint32 aMaxLen, OsciNetworkAddress & aAddress, int32 aTimeoutMsec = -1, uint32 aMultiMaxLen = 0, Osci_Vector< uint32, OsciMemAllocator > * aPacketLen = NULL, Osci_Vector< OsciNetworkAddress, OsciMemAllocator > * aPacketSource = NULL)` [inline]

7.249.3.11 **TPVSocketEvent** `OsciUDPSocketI::SendTo (const uint8 *& aPtr, uint32 aLen, OsciNetworkAddress & aAddress, int32 aTimeoutMsec = -1)` [inline]

7.249.3.12 `int32 OsciUDPSocketI::SetMulticastTTL (int32 aTTL)`

7.249.3.13 **TPVSocketEvent** `OsciUDPSocketI::ThreadLogoff ()`

Reimplemented from [OsciIPSocketI](#).

7.249.3.14 **TPVSocketEvent** `OsciUDPSocketI::ThreadLogon (OsciSocketServI * aServ, OsciSocketObserver * aObserver)`

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

- [oscl_udp_socket.h](#)

7.250 OsciUuid Struct Reference

```
#include <osci_uuid.h>
```

Public Methods

- [OsciUuid](#) ()
- [OsciUuid](#) (uint32 l, uint16 w1, uint16 w2, uint8 b1, uint8 b2, uint8 b3, uint8 b4, uint8 b5, uint8 b6, uint8 b7, uint8 b8)
- [OsciUuid](#) (const char *aUuidString)
- [OsciUuid](#) (const OsciUuid &uuid)
- OsciUuid & [operator=](#) (const OsciUuid &src)
- bool [operator==](#) (const OsciUuid &src) const
- bool [operator!=](#) (const OsciUuid &src) const

Data Fields

- uint32 [data1](#)
- uint16 [data2](#)
- uint16 [data3](#)
- uint8 [data4](#) [BYTES_IN_UUID_ARRAY]

7.250.1 Detailed Description

OSCL UUID structure used for unique identification of modules and interfaces.

7.250.2 Constructor & Destructor Documentation

7.250.2.1 `OsciUuid::OsciUuid()` [inline]

7.250.2.2 `OsciUuid::OsciUuid (uint32 l, uint16 w1, uint16 w2, uint8 b1, uint8 b2, uint8 b3, uint8 b4, uint8 b5, uint8 b6, uint8 b7, uint8 b8)` [inline]

7.250.2.3 `OsciUuid::OsciUuid (const char * aUuidString)` [inline]

7.250.2.4 `OsciUuid::OsciUuid (const OsciUuid & uuid)` [inline]

7.250.3 Member Function Documentation

7.250.3.1 `bool OsciUuid::operator!= (const OsciUuid & src) const` [inline]

7.250.3.2 `OsciUuid& OsciUuid::operator= (const OsciUuid & src)` [inline]

7.250.3.3 `bool OsciUuid::operator== (const OsciUuid & src) const` [inline]

7.250.4 Field Documentation

7.250.4.1 `uint32 OsciUuid::data1`

7.250.4.2 `uint16 OsciUuid::data2`

7.250.4.3 `uint16 OsciUuid::data3`

7.250.4.4 `uint8 OsciUuid::data4[BYTES_IN_UUID_ARRAY]`

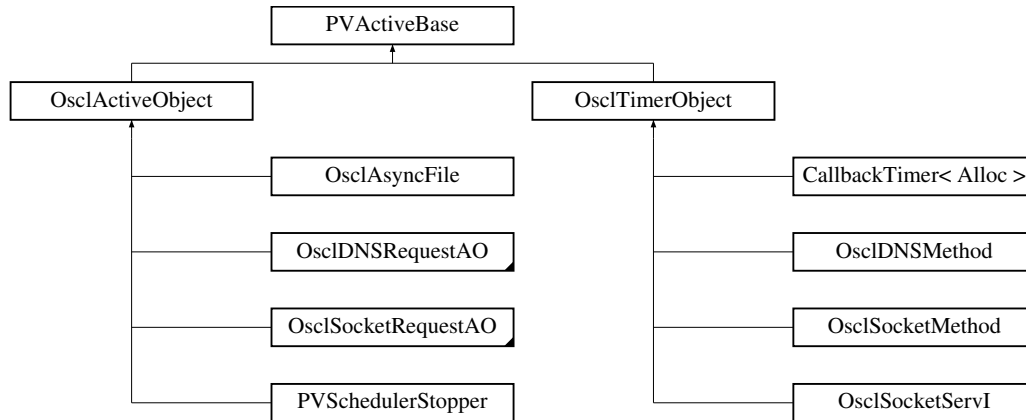
The documentation for this struct was generated from the following file:

- [osci_uuid.h](#)

7.251 PActiveBase Class Reference

```
#include <oscl_scheduler_aobase.h>
```

Inheritance diagram for PActiveBase::



Public Methods

- [PActiveBase](#) (const char name[], int32 pri)
- virtual [~PActiveBase](#) ()
- bool [IsInAnyQ](#) ()
- virtual int32 [RunError](#) (int32 aError)=0
- virtual void [Run](#) ()=0
- virtual void [DoCancel](#) ()=0
- void [AddToScheduler](#) ()
- void [RemoveFromScheduler](#) ()
- void [Destroy](#) ()
- void [Activate](#) ()
- OSCI_IMPORT_REF bool [IsAdded](#) () const
- void [Cancel](#) ()

Data Fields

- uint32 [iAddedNum](#)
- [OsciNameString](#)< PVEXECNAMELEN > [iName](#)
- [PVThreadContext](#) [iThreadContext](#)
- [PActiveStats](#) * [iPActiveStats](#)
- [TReadyQueLink](#) [iPVRReadyQLink](#)
- bool [iBusy](#)
- [OsciAOSStatus](#) [iStatus](#)

Friends

- class [PActiveStats](#)
- class [OsciSchedulerCommonBase](#)

- class [OscActiveObject](#)
- class [OscTimerObject](#)
- class [OscReadyQ](#)
- class [OscReadyCompare](#)
- class [OscReadySetPosition](#)
- class [OscExecScheduler](#)

7.251.1 Detailed Description

PV Scheduler internal AO base class. Both [OscActiveObject](#) and [OscTimerObject](#) derive from this class. For Symbian, this just container has the desired additions to the basic CTimer or OscActiveObj functionality. For non-Symbian, this class contains the entire AO implementation.

7.251.2 Constructor & Destructor Documentation

7.251.2.1 `PActiveBase::PActiveBase (const char name [], int32 pri)`

7.251.2.2 `virtual PActiveBase::~~PActiveBase ()` [virtual]

7.251.3 Member Function Documentation

7.251.3.1 `void PActiveBase::Activate ()`

7.251.3.2 `void PActiveBase::AddToScheduler ()`

Reimplemented in [OscActiveObject](#), and [OscTimerObject](#).

7.251.3.3 `void PActiveBase::Cancel ()`

Reimplemented in [OscActiveObject](#), and [OscTimerObject](#).

7.251.3.4 `void PActiveBase::Destroy ()`

7.251.3.5 `virtual void PActiveBase::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 [OscIDNSRequestAO](#), [OscSocketRequestAO](#), [OscActiveObject](#), and [OscTimerObject](#).

7.251.3.6 `OSCL_IMPORT_REF bool PActiveBase::IsAdded ()`

7.251.3.7 `bool PActiveBase::IsInAnyQ () [inline]`

7.251.3.8 `void PActiveBase::RemoveFromScheduler ()`

Reimplemented in [OscActiveObject](#), and [OscTimerObject](#).

7.251.3.9 `virtual void PActiveBase::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 [OscIDNSMethod](#), [OscIDNSRequestAO](#), [OscSocketMethod](#), [OscSocketRequestAO](#), and [CallbackTimer< Alloc >](#).

7.251.3.10 `virtual int32 PActiveBase::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 `OscErrNone` if the error was handled, or returns the input `aError` value if not handled.

Implemented in [OscActiveObject](#), and [OscTimerObject](#).

7.251.4 Friends And Related Function Documentation

- 7.251.4.1 friend class `OscActiveObject` [friend]
- 7.251.4.2 friend class `OscExecScheduler` [friend]
- 7.251.4.3 friend class `OscReadyCompare` [friend]
- 7.251.4.4 friend class `OscReadyQ` [friend]
- 7.251.4.5 friend class `OscReadySetPosition` [friend]
- 7.251.4.6 friend class `OscSchedulerCommonBase` [friend]
- 7.251.4.7 friend class `OscTimerObject` [friend]
- 7.251.4.8 friend class `PActiveStats` [friend]

7.251.5 Field Documentation

- 7.251.5.1 `uint32 PActiveBase::iAddedNum`
- 7.251.5.2 `bool PActiveBase::iBusy`
- 7.251.5.3 `OscNameString<PVEXECNAMELEN> PActiveBase::iName`
- 7.251.5.4 `PActiveStats* PActiveBase::iPActiveStats`
- 7.251.5.5 `TReadyQueLink PActiveBase::iPVReadyQLink`
- 7.251.5.6 `OscIAOStatus PActiveBase::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.

7.251.5.7 `PVThreadContext PActiveBase::iThreadContext`

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

- [oscl_scheduler_aobase.h](#)

7.252 PVActiveStats Class Reference

```
#include <oscl_scheduler_aobase.h>
```

Friends

- class [PVActiveBase](#)
- class [OscExecScheduler](#)
- class [OscExecSchedulerCommonBase](#)
- class [OscActiveObject](#)
- class [OscTimerObject](#)
- class [OscReadyQ](#)

7.252.1 Detailed Description

PV AO statistics

7.252.2 Friends And Related Function Documentation

7.252.2.1 friend class [OscActiveObject](#) [friend]

7.252.2.2 friend class [OscExecScheduler](#) [friend]

7.252.2.3 friend class [OscExecSchedulerCommonBase](#) [friend]

7.252.2.4 friend class [OscReadyQ](#) [friend]

7.252.2.5 friend class [OscTimerObject](#) [friend]

7.252.2.6 friend class [PVActiveBase](#) [friend]

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

- [oscl_scheduler_aobase.h](#)

7.253 PVLogger Class Reference

```
#include <pvlogger.h>
```

Public Types

- typedef int32 [log_level_type](#)
- typedef int32 [message_id_type](#)
- typedef int32 [filter_status_type](#)
- typedef [_OscBasicAllocator](#) [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](#) ([OscSharedPtr](#)< [PVLoggerAppender](#) > &appender)
- void [RemoveAppender](#) ([OscSharedPtr](#)< [PVLoggerAppender](#) > &appender)
- void [AddFilter](#) ([OscSharedPtr](#)< [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](#)

7.253.1 Member Typedef Documentation

7.253.1.1 `typedef _OscBasicAllocator PVLogger::alloc_type`

7.253.1.2 `typedef int32 PVLogger::filter_status_type`

7.253.1.3 `typedef int32 PVLogger::log_level_type`

7.253.1.4 `typedef int32 PVLogger::message_id_type`

7.253.2 Constructor & Destructor Documentation

7.253.2.1 `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

7.253.2.2 `virtual PVLogger::~~PVLogger () [inline, virtual]`

7.253.3 Member Function Documentation

7.253.3.1 `void PVLogger::AddAppender (OscSharedPtr< 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

7.253.3.2 `void PVLogger::AddFilter (OscSharedPtr< 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

7.253.3.3 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:**7.253.3.4 void PVLogger::DisableAppenderInheritance () [inline]**

This method disables appender inheritance for the logging control point

7.253.3.5 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

7.253.3.6 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

7.253.3.7 uint32 PVLogger::GetNumAppenders () [inline]

This method returns the number of appenders attached to the logging control point.

7.253.3.8 PVLogger* PVLogger::GetParent () [inline, protected]
7.253.3.9 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

7.253.3.10 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

7.253.3.11 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

7.253.3.12 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

7.253.3.13 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

7.253.3.14 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

7.253.3.15 void PVLogger::RemoveAppender (OsciSharedPtr< 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

7.253.3.16 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

7.253.3.17 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

7.253.3.18 void PVLogger::SetParent (PVLogger **parentLogger*) [inline, protected]**7.253.4 Friends And Related Function Documentation****7.253.4.1** friend class PVLoggerRegistry [friend]

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

- [pvlogger.h](#)

7.254 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

7.254.1 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.

7.254.2 Member Typedef Documentation

7.254.2.1 typedef PVLogger::message_id_type PVLoggerAppender::message_id_type

7.254.3 Constructor & Destructor Documentation

7.254.3.1 virtual PVLoggerAppender::~~PVLoggerAppender () [inline, virtual]

7.254.4 Member Function Documentation

7.254.4.1 virtual void PVLoggerAppender::AppendBuffers ([message_id_type](#) msgID, int32 numPairs, va_list va) [pure virtual]

7.254.4.2 virtual void PVLoggerAppender::AppendString ([message_id_type](#) msgID, const char *fmt, va_list va) [pure virtual]

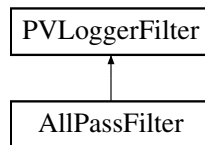
The documentation for this class was generated from the following file:

- [pvlogger_accessories.h](#)

7.255 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 FilterOpaqueMessage](#) (char *tag, [message_id_type](#) msgID, [log_level_type](#) level)=0

7.255.1 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.

7.255.2 Member Typedef Documentation

7.255.2.1 typedef PVLogger::filter_status_type PVLoggerFilter::filter_status_type

Reimplemented in [AllPassFilter](#).

7.255.2.2 typedef PVLogger::log_level_type PVLoggerFilter::log_level_type

Reimplemented in [AllPassFilter](#).

7.255.2.3 typedef PVLogger::message_id_type PVLoggerFilter::message_id_type

Reimplemented in [AllPassFilter](#).

7.255.3 Constructor & Destructor Documentation

7.255.3.1 virtual PVLoggerFilter::~PVLoggerFilter () [inline, virtual]

7.255.4 Member Function Documentation

7.255.4.1 virtual [filter_status_type](#) PVLoggerFilter::FilterOpaqueMessge (char * *tag*, [message_id_type](#) *msgID*, [log_level_type](#) *level*) [pure virtual]

Implemented in [AllPassFilter](#).

7.255.4.2 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](#)

7.256 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

7.256.1 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.

7.256.2 Member Typedef Documentation

7.256.2.1 typedef PVLogger::message_id_type PVLoggerLayout::message_id_type

7.256.3 Constructor & Destructor Documentation

7.256.3.1 virtual PVLoggerLayout::~~PVLoggerLayout () [inline, virtual]

7.256.4 Member Function Documentation

7.256.4.1 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.

7.256.4.2 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](#)

7.257 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](#) ()

7.257.1 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.

7.257.2 Member Typedef Documentation

7.257.2.1 typedef PVLogger::alloc_type PVLoggerRegistry::alloc_type

7.257.2.2 typedef PVLogger::log_level_type PVLoggerRegistry::log_level_type

7.257.3 Constructor & Destructor Documentation

7.257.3.1 OSCL_IMPORT_REF PVLoggerRegistry::PVLoggerRegistry ()

PVLoggerRegistry Constructor

7.257.3.2 virtual OSCL_IMPORT_REF PVLoggerRegistry::~~PVLoggerRegistry ()
[virtual]

PVLoggerRegistry Destructor

7.257.4 Member Function Documentation

7.257.4.1 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

7.257.4.2 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

7.257.4.3 OSCL_IMPORT_REF PVLoggerRegistry* PVLoggerRegistry::GetPVLoggerRegistry () [static]

Get the logger registry. There is only one logger registry instance per thread.

7.257.4.4 OSCL_IMPORT_REF void PVLoggerRegistry::SetNodeLogLevelExplicit (Osci_TagTree< PVLogger *, alloc_type >::node_type * *node*, log_level_type *level*)

This method recursively propagates the log level to all the descendents, 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

7.257.4.5 OSCL_IMPORT_REF bool PVLoggerRegistry::SetNodeLogLevelExplicit (char * *tagIn*, *log_level_type level*)

This method propagates the log level to all the descendents 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.

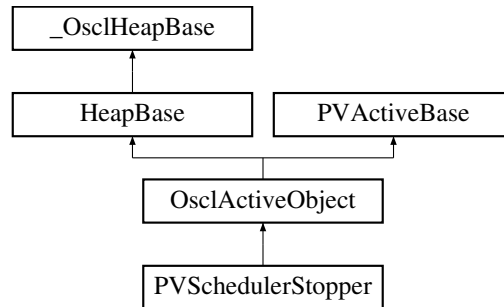
The documentation for this class was generated from the following file:

- [pvlogger_registry.h](#)

7.258 PVSchedulerStopper Class Reference

```
#include <oscl_scheduler.h>
```

Inheritance diagram for PVSchedulerStopper::



Public Methods

- [PVSchedulerStopper \(\)](#)
- [~PVSchedulerStopper \(\)](#)

7.258.1 Detailed Description

Scheduler stopper AO class, for internal use by scheduler.

7.258.2 Constructor & Destructor Documentation

7.258.2.1 PVSchedulerStopper::PVSchedulerStopper ()

7.258.2.2 PVSchedulerStopper::~~PVSchedulerStopper ()

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

- [oscl_scheduler.h](#)

7.259 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](#)

7.259.1 Constructor & Destructor Documentation

7.259.1.1 [PVSockBufRecv::PVSockBufRecv \(\)](#) [inline]

7.259.1.2 [PVSockBufRecv::PVSockBufRecv \(uint8 * aPtr, uint32 aLen, uint32 aMax\)](#) [inline]

7.259.1.3 [PVSockBufRecv::PVSockBufRecv \(const PVSockBufRecv & a\)](#) [inline]

7.259.2 Field Documentation

7.259.2.1 [uint32 PVSockBufRecv::iLen](#)

7.259.2.2 [uint32 PVSockBufRecv::iMaxLen](#)

7.259.2.3 [uint8* PVSockBufRecv::iPtr](#)

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

- [oscl_socket_request.h](#)

7.260 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](#)

7.260.1 Constructor & Destructor Documentation

7.260.1.1 [PVSockBufSend::PVSockBufSend \(\)](#) [inline]

7.260.1.2 [PVSockBufSend::PVSockBufSend \(const uint8 * aPtr, uint32 aLen\)](#) [inline]

7.260.1.3 [PVSockBufSend::PVSockBufSend \(const PVSockBufSend & a\)](#) [inline]

7.260.2 Field Documentation

7.260.2.1 [uint32 PVSockBufSend::iLen](#)

7.260.2.2 [const uint8* PVSockBufSend::iPtr](#)

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

- [oscl_socket_request.h](#)

7.261 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 [OscActiveObject](#)
- class [OscTimerObject](#)
- class [OscExecScheduler](#)
- class [OscCoeActiveScheduler](#)
- class [OscExecSchedulerCommonBase](#)
- class [OscExecSchedulerBase](#)
- class [OscCoeActiveSchedulerBase](#)

7.261.1 Constructor & Destructor Documentation

7.261.1.1 OSCL_IMPORT_REF [PVThreadContext::PVThreadContext](#) ()

7.261.1.2 OSCL_IMPORT_REF [PVThreadContext::~~PVThreadContext](#) ()

7.261.2 Member Function Documentation

7.261.2.1 OSCL_IMPORT_REF void [PVThreadContext::EnterThreadContext](#) ()

enter and exit thread context.

7.261.2.2 OSCL_IMPORT_REF void [PVThreadContext::ExitThreadContext](#) ()

7.261.2.3 OSCL_IMPORT_REF uint32 [PVThreadContext::Id](#) () [static]

static routine to get a unique thread ID for caller's thread context.

7.261.2.4 OSCL_IMPORT_REF bool PVThreadContext::IsSameThreadContext ()

compare caller's thread context to this one.

7.261.2.5 OSCL_IMPORT_REF bool PVThreadContext::ThreadHasScheduler () [static]

a static utility to tell whether the calling thread has any scheduler– either Osci scheduler or native scheduler.

7.261.3 Friends And Related Function Documentation

7.261.3.1 friend class OsciActiveObject [friend]

7.261.3.2 friend class OsciCoeActiveScheduler [friend]

7.261.3.3 friend class OsciCoeActiveSchedulerBase [friend]

7.261.3.4 friend class OsciExecScheduler [friend]

7.261.3.5 friend class OsciExecSchedulerBase [friend]

7.261.3.6 friend class OsciExecSchedulerCommonBase [friend]

7.261.3.7 friend class OsciTimerObject [friend]

7.261.3.8 friend class PVActiveBase [friend]

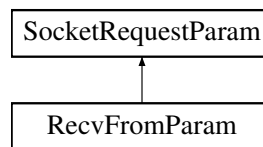
The documentation for this class was generated from the following file:

- [oscl_scheduler_threadcontext.h](#)

7.262 RecvFromParam Class Reference

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

Inheritance diagram for RecvFromParam::



Public Methods

- `RecvFromParam` (`uint8 * &aPtr`, `uint32 aMaxLen`, `OscNetworkAddress &aAddress`, `uint32 flags`, `uint32 aMultiMax`, `Osc_Vector< uint32, OscMemAllocator > *aPacketLen`, `Osc_Vector< OscNetworkAddress, OscMemAllocator > *aPacketSource`)

Data Fields

- `PVSockBufRecv iBufRecv`
- `uint32 iFlags`
- `OscNetworkAddress & iAddr`
- `uint32 iMultiMaxLen`
- `Osc_Vector< uint32, OscMemAllocator > * iPacketLen`
- `Osc_Vector< OscNetworkAddress, OscMemAllocator > * iPacketSource`

7.262.1 Constructor & Destructor Documentation

7.262.1.1 `RecvFromParam::RecvFromParam` (`uint8 * & aPtr`, `uint32 aMaxLen`, `OscNetworkAddress & aAddress`, `uint32 flags`, `uint32 aMultiMax`, `Osc_Vector< uint32, OscMemAllocator > * aPacketLen`, `Osc_Vector< OscNetworkAddress, OscMemAllocator > * aPacketSource`) [inline]

7.262.2 Field Documentation

7.262.2.1 `OscNetworkAddress& RecvFromParam::iAddr`

7.262.2.2 `PVSockBufRecv RecvFromParam::iBufRecv`

7.262.2.3 `uint32 RecvFromParam::iFlags`

7.262.2.4 `uint32 RecvFromParam::iMultiMaxLen`

7.262.2.5 `Osc_Vector<uint32, OscMemAllocator>* RecvFromParam::iPacketLen`

7.262.2.6 `Osc_Vector<OscNetworkAddress, OscMemAllocator>* RecvFromParam::iPacketSource`

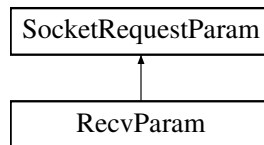
The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

7.263 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

7.263.1 Constructor & Destructor Documentation

7.263.1.1 `RecvParam::RecvParam (uint8 *& aPtr, uint32 aMaxLen, uint32 flags) [inline]`

7.263.2 Field Documentation

7.263.2.1 [PVSockBufRecv](#) `RecvParam::iBufRecv`

7.263.2.2 `uint32 RecvParam::iFlags`

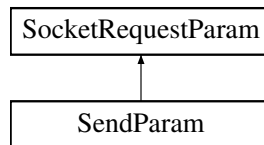
The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

7.264 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

7.264.1 Detailed Description

Socket method parameter sets

7.264.2 Constructor & Destructor Documentation

7.264.2.1 [SendParam::SendParam](#) (const uint8 *& aPtr, uint32 aLen, uint32 aFlags) [inline]

7.264.3 Field Documentation

7.264.3.1 [PVSockBufSend](#) SendParam::iBufSend

7.264.3.2 uint32 SendParam::iFlags

7.264.3.3 uint32 SendParam::iXferLen

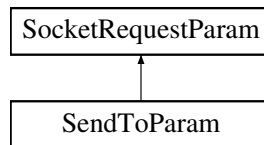
The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

7.265 SendToParam Class Reference

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

Inheritance diagram for SendToParam::



Public Methods

- [SendToParam](#) (const uint8 *&aPtr, uint32 aLen, [OscNetworkAddress](#) &anAddr, uint32 flags)
- [~SendToParam](#) ()

Data Fields

- [PVSockBufSend](#) iBufSend
- uint32 iFlags
- [OscNetworkAddress](#) iAddr
- uint32 iXferLen

7.265.1 Constructor & Destructor Documentation

7.265.1.1 [SendToParam::SendToParam](#) (const uint8 *& aPtr, uint32 aLen, [OscNetworkAddress](#) & anAddr, uint32 flags) [inline]

7.265.1.2 [SendToParam::~~SendToParam](#) () [inline]

7.265.2 Field Documentation

7.265.2.1 [OscNetworkAddress](#) [SendToParam::iAddr](#)

7.265.2.2 [PVSockBufSend](#) [SendToParam::iBufSend](#)

7.265.2.3 uint32 [SendToParam::iFlags](#)

7.265.2.4 uint32 [SendToParam::iXferLen](#)

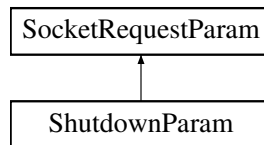
The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

7.266 ShutdownParam Class Reference

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

Inheritance diagram for ShutdownParam::



Public Methods

- [ShutdownParam](#) ([TPVSocketShutdown](#) aHow)

Data Fields

- [TPVSocketShutdown](#) iHow

7.266.1 Constructor & Destructor Documentation

7.266.1.1 [ShutdownParam::ShutdownParam](#) ([TPVSocketShutdown](#) aHow) [inline]

7.266.2 Field Documentation

7.266.2.1 [TPVSocketShutdown](#) [ShutdownParam::iHow](#)

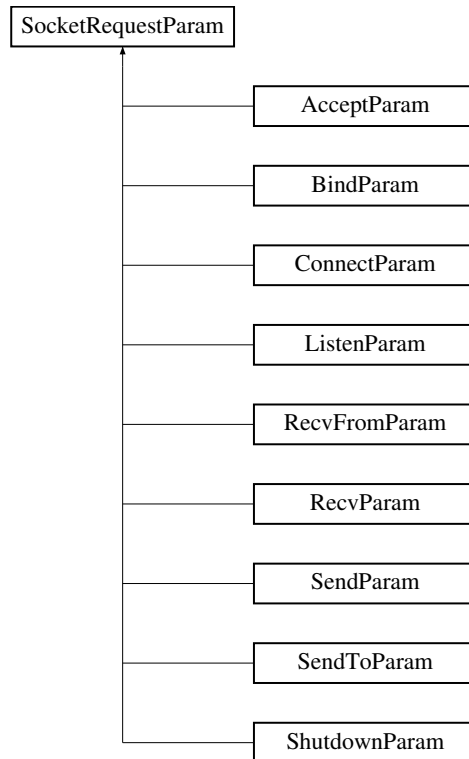
The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

7.267 SocketRequestParam Class Reference

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

Inheritance diagram for SocketRequestParam::



Public Methods

- [SocketRequestParam](#) (TPVSocketFxn aFxn)

Data Fields

- [TPVSocketFxn](#) iFxn

7.267.1 Detailed Description

Base class for all socket method parameter sets

7.267.2 Constructor & Destructor Documentation

7.267.2.1 [SocketRequestParam::SocketRequestParam \(TPVSocketFxn aFxn\)](#) [inline]

7.267.3 Field Documentation

7.267.3.1 [TPVSocketFxn](#) SocketRequestParam::iFxn

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

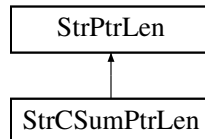
- [oscl_socket_request.h](#)

7.268 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](#)

7.268.1 Detailed Description

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

7.268.2 Member Typedef Documentation

7.268.2.1 `typedef int16 StrCsumPtrLen::ChecksumType`

7.268.3 Constructor & Destructor Documentation

7.268.3.1 `StrCsumPtrLen::StrCsumPtrLen ()` [inline]

7.268.3.2 `StrCsumPtrLen::StrCsumPtrLen (const char * newPtr)` [inline]

7.268.3.3 `StrCsumPtrLen::StrCsumPtrLen (const char * newPtr, uint32 newLen)` [inline]

7.268.3.4 `StrCsumPtrLen::StrCsumPtrLen (const StrCsumPtrLen & rhs)` [inline]

7.268.3.5 `StrCsumPtrLen::StrCsumPtrLen (const StrPtrLen & rhs)` [inline]

7.268.4 Member Function Documentation

7.268.4.1 `ChecksumType StrCsumPtrLen::getChecksum () const` [inline]

7.268.4.2 `c_bool StrCsumPtrLen::isCIEquivalentTo (const StrCsumPtrLen & rhs) const`
[inline]

7.268.4.3 `c_bool StrCsumPtrLen::operator!= (const StrCsumPtrLen & rhs) const` [inline]

7.268.4.4 `StrCsumPtrLen& StrCsumPtrLen::operator= (const char * rhs)` [inline]

Reimplemented from [StrPtrLen](#).

7.268.4.5 `StrCsumPtrLen& StrCsumPtrLen::operator= (const StrPtrLen & rhs)` [inline]

Reimplemented from [StrPtrLen](#).

7.268.4.6 `StrCsumPtrLen& StrCsumPtrLen::operator= (const StrCsumPtrLen & rhs)`
[inline]

7.268.4.7 `c_bool StrCsumPtrLen::operator== (const StrCsumPtrLen & rhs) const` [inline]

7.268.4.8 `OSCL_IMPORT_REF void StrCsumPtrLen::setChecksum ()`

7.268.4.9 `void StrCsumPtrLen::setPtrLen (const char * newPtr, uint32 newLen)` [inline]

Reimplemented from [StrPtrLen](#).

7.268.5 Field Documentation

7.268.5.1 `ChecksumType StrCsumPtrLen::checksum` [protected]

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

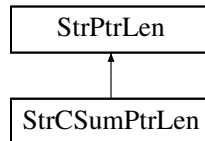
- [oscl_str_ptr_len.h](#)

7.269 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](#)

7.269.1 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.

7.269.2 Constructor & Destructor Documentation

7.269.2.1 `StrPtrLen::StrPtrLen (const char * newPtr)` [inline]

7.269.2.2 `StrPtrLen::StrPtrLen (const char * newPtr, uint32 newLen)` [inline]

7.269.2.3 `StrPtrLen::StrPtrLen ()` [inline]

7.269.2.4 `StrPtrLen::StrPtrLen (const StrPtrLen & rhs)` [inline]

7.269.3 Member Function Documentation

7.269.3.1 `const char* StrPtrLen::c_str () const` [inline]

7.269.3.2 `c_bool StrPtrLen::isCIEquivalentTo (const StrPtrLen & rhs) const` [inline]

7.269.3.3 `c_bool StrPtrLen::isCIPrefixOf (const StrPtrLen & rhs) const` [inline]

7.269.3.4 `bool StrPtrLen::isLetter (const char c) const` [inline, protected]

7.269.3.5 `int32 StrPtrLen::length () const` [inline]

7.269.3.6 `int32 StrPtrLen::operator!= (const StrPtrLen & rhs) const` [inline]

7.269.3.7 `StrPtrLen& StrPtrLen::operator= (const char * rhs)` [inline]

Reimplemented in [StrCSumPtrLen](#).

7.269.3.8 `StrPtrLen& StrPtrLen::operator= (const StrPtrLen & rhs)` [inline]

Reimplemented in [StrCSumPtrLen](#).

7.269.3.9 `int32 StrPtrLen::operator== (const StrPtrLen & rhs) const` [inline]

7.269.3.10 `void StrPtrLen::setPtrLen (const char * newPtr, uint32 newLen)` [inline]

Reimplemented in [StrCSumPtrLen](#).

7.269.3.11 `int32 StrPtrLen::size () const` [inline]

7.269.4 Field Documentation

7.269.4.1 `int32 StrPtrLen::len` [protected]

7.269.4.2 `const char* StrPtrLen::ptr` [protected]

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

- [oscl_str_ptr_len.h](#)

7.270 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 [OsciBasicTimeStruct](#) &in_tv)
Create a TimeValue representing the absolute time specified by the BasicTimeStruct.
- OSCL_COND_IMPORT_REF [TimeValue](#) (const [ISO8601timeStrBuf](#) time_strbuf)
- OSCL_COND_IMPORT_REF [TimeValue](#) (uint16 aYear, uint16 aMonth, uint16 aDay, uint16 aHour, uint16 aMinute, uint16 aSecond, uint16 aMilliseconds)
- OSCL_COND_IMPORT_REF [TimeValue](#) ([OsciBasicDateTimeStruct](#) 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 coverts 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
- OSCL_COND_IMPORT_REF uint32 [get_usec](#) () const
Get a 32 bit value representing the number of microseconds in the time value.
- OSCL_COND_IMPORT_REF uint64 [get_timevalue_in_usec](#) () const
Get a 64 bit value representing the time value converted to microseconds.
- 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 PV 8601 format.
- OSCL_IMPORT_REF int [get_ISO8601_str_time](#) (ISO8601timeStrBuf 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 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 bool [is_zulu](#) () const
Manipulate internal flags to mark the time value as being in "zulu" (GMT) time.
- OSCL_COND_IMPORT_REF void [set_zulu](#) (bool is_zulu)
- OSCL_COND_IMPORT_REF TimeValue & [operator=](#) (const TimeValue &a)
Assignment operator.
- OSCL_COND_IMPORT_REF TimeValue & [operator+=](#) (const TimeValue &a)
+= operator
- OSCL_COND_IMPORT_REF TimeValue & [operator-=](#) (const TimeValue &a)
-= operator
- OSCL_COND_IMPORT_REF TimeValue & [operator *=](#) (const int scale)
This operator scales the time value by a constant.
- OSCL_COND_IMPORT_REF [OsciBasicTimeStruct](#) * [get_timeval_ptr](#) ()
- OSCL_COND_IMPORT_REF TimeValue & [operator+=](#) (const int32 aSeconds)
- OSCL_COND_IMPORT_REF TimeValue & [operator-=](#) (const int32 aSeconds)

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)

7.270.1 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`

7.270.2 Constructor & Destructor Documentation

7.270.2.1 `OSCL_COND_IMPORT_REF TimeValue::TimeValue ()`

Create a TimeValue representing the current time.

7.270.2.2 `OSCL_COND_IMPORT_REF TimeValue::TimeValue (const TimeValue & Tv)`

Copy constructor.

7.270.2.3 `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.

7.270.2.4 `OSCL_COND_IMPORT_REF TimeValue::TimeValue (const OsciBasicTimeStruct & in_tv)`

Create a TimeValue representing the absolute time specified by the BasicTimeStruct.

Parameters:

- in_tv* OsciBasicTimeStruct as defined for each platform.

7.270.2.5 `OSCL_COND_IMPORT_REF TimeValue::TimeValue (const ISO8601timeStrBuf time_strbuf)`

7.270.2.6 `OSCL_COND_IMPORT_REF TimeValue::TimeValue (uint16 aYear, uint16 aMonth, uint16 aDay, uint16 aHour, uint16 aMinute, uint16 aSecond, uint16 aMilliseconds)`

TimeValue constructor that sets time according to following input parameter for a specific date time. Please note that no argument is check for its validity (range etc) It might assert incase wrong argument are passed by user of this api.

Parameters:

- in*] uint16 wYear;
- in*] uint16 wMonth; Jan = 1 to Dec = 12
- in*] uint16 wDay; 1-28/29/30/31
- in*] uint16 wHour; 0 to 23
- in*] uint16 wMinute; 0 to 59
- in*] uint16 wSecond; 0 to 59
- in*] uint16 wMilliseconds; 0 to 999

7.270.2.7 OSCL_COND_IMPORT_REF TimeValue::TimeValue (OscBasicDateTimeStruct *in_ts*)

Create a TimeValue representing the absolute time specified by the BasicDateTimeStruct.

Parameters:

- in_ts* OscBasicDateTimeStruct 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

7.270.3 Member Function Documentation
7.270.3.1 OSCL_IMPORT_REF int TimeValue::get_ISO8601_str_time (ISO8601timeStrBuf *time_strbuf*)

Get a PV extended text representation of the Time based on the ISO 8601 format.

Parameters:

- time_strbuf* A ISO8601timeStrBuf 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 "1985-04-12 10:15:30Z".

7.270.3.2 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

7.270.3.3 OSCL_IMPORT_REF int TimeValue::get_pv8601_str_time (PV8601timeStrBuf *time_strbuf*)

Get a PV extended text representation of the Time based on the PV 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".

7.270.3.4 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".

7.270.3.5 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.)

7.270.3.6 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".

7.270.3.7 OSCL_COND_IMPORT_REF OsciBasicTimeStruct* TimeValue::get_timeval_ptr ()**7.270.3.8 OSCL_COND_IMPORT_REF uint64 TimeValue::get_timevalue_in_usec ()**

Get a 64 bit value representing the time value converted to microseconds.

Returns:

Returns a uint64 value representing the time value in terms of microseconds. The time origin is dependent on platform for which OSCL is compiled. For example for symbian it is midnight, January 1st, 0 AD for windows it is January 1, 1601 (UTC)

7.270.3.9 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.

7.270.3.10 OSCL_COND_IMPORT_REF bool TimeValue::is_zero ()

Determine if the time value is zero.

7.270.3.11 OSCL_COND_IMPORT_REF bool TimeValue::is_zulu ()

Manipulate internal flags to mark the time value as being in "zulu" (GMT) time.

7.270.3.12 OSCL_COND_IMPORT_REF TimeValue& TimeValue::operator *= (const int *scale*)

This operator scales the time value by a constant.

7.270.3.13 OSCL_COND_IMPORT_REF TimeValue& TimeValue::operator += (const int32 *aSeconds*)
7.270.3.14 OSCL_COND_IMPORT_REF TimeValue& TimeValue::operator += (const TimeValue & *a*)

+= operator

7.270.3.15 OSCL_COND_IMPORT_REF TimeValue& TimeValue::operator -= (const int32 *aSeconds*)
7.270.3.16 OSCL_COND_IMPORT_REF TimeValue& TimeValue::operator -= (const TimeValue & *a*)

-= operator

7.270.3.17 OSCL_COND_IMPORT_REF TimeValue& TimeValue::operator = (const TimeValue & *a*)

Assignment operator.

7.270.3.18 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

7.270.3.19 OSCL_COND_IMPORT_REF void TimeValue::set_to_current_time ()

Set the time value to the current system time.

7.270.3.20 OSCL_COND_IMPORT_REF void TimeValue::set_to_zero ()

Set the time value to zero (represents a zero interval).

7.270.3.21 OSCL_COND_IMPORT_REF void TimeValue::set_zulu (bool is_zulu)**7.270.3.22 OSCL_COND_IMPORT_REF int32 TimeValue::to_msec ()****7.270.4 Friends And Related Function Documentation****7.270.4.1 friend class NTPTime [friend]****7.270.4.2 OSCL_COND_IMPORT_REF friend bool operator!= (const TimeValue & a, const TimeValue & b) [friend]****7.270.4.3 OSCL_COND_IMPORT_REF friend bool operator< (const TimeValue & a, const TimeValue & b) [friend]****7.270.4.4 OSCL_COND_IMPORT_REF friend bool operator<= (const TimeValue & a, const TimeValue & b) [friend]****7.270.4.5 OSCL_COND_IMPORT_REF friend bool operator== (const TimeValue & a, const TimeValue & b) [friend]****7.270.4.6 OSCL_COND_IMPORT_REF friend bool operator> (const TimeValue & a, const TimeValue & b) [friend]****7.270.4.7 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](#)

7.271 TLSStorageOps Class Reference

```
#include <oscl_tls.h>
```

Static Public Methods

- OSCL_IMPORT_REF void [save_registry](#) (TOscITlsKey *key, OsciAny *ptr, int32 &)
- OSCL_IMPORT_REF OsciAny * [get_registry](#) (TOscITlsKey *key)

7.271.1 Member Function Documentation

7.271.1.1 OSCL_IMPORT_REF OsciAny* TLSStorageOps::get_registry (TOscITlsKey * key)
[static]

7.271.1.2 OSCL_IMPORT_REF void TLSStorageOps::save_registry (TOscITlsKey * key,
OsciAny * ptr, int32 &) [static]

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

- [oscl_tls.h](#)

7.272 TReadyQueLink Class Reference

```
#include <oscl_scheduler_readyq.h>
```

Public Methods

- [TReadyQueLink \(\)](#)

Data Fields

- [int32 iAOPriority](#)
- [uint32 iTimeToRunTicks](#)
- [uint32 iTimeQueuedTicks](#)
- [uint32 iSeqNum](#)
- [OsclAny * iIsIn](#)

7.272.1 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.

7.272.2 Constructor & Destructor Documentation

7.272.2.1 [TReadyQueLink::TReadyQueLink \(\)](#) [inline]

7.272.3 Field Documentation

7.272.3.1 [int32 TReadyQueLink::iAOPriority](#)

7.272.3.2 [OsclAny* TReadyQueLink::iIsIn](#)

7.272.3.3 [uint32 TReadyQueLink::iSeqNum](#)

7.272.3.4 [uint32 TReadyQueLink::iTimeQueuedTicks](#)

7.272.3.5 [uint32 TReadyQueLink::iTimeToRunTicks](#)

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

- [oscl_scheduler_readyq.h](#)

7.273 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](#)

7.273.1 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.

7.273.2 Constructor & Destructor Documentation

- 7.273.2.1 `WStrPtrLen::WStrPtrLen (const oscl_wchar * newPtr)` [inline]
- 7.273.2.2 `WStrPtrLen::WStrPtrLen (const oscl_wchar * newPtr, uint32 newLen)` [inline]
- 7.273.2.3 `WStrPtrLen::WStrPtrLen ()` [inline]
- 7.273.2.4 `WStrPtrLen::WStrPtrLen (const WStrPtrLen & rhs)` [inline]

7.273.3 Member Function Documentation

- 7.273.3.1 `const oscl_wchar* WStrPtrLen::c_str () const` [inline]
- 7.273.3.2 `c_bool WStrPtrLen::isCIEquivalentTo (const WStrPtrLen & rhs) const` [inline]
- 7.273.3.3 `int32 WStrPtrLen::length () const` [inline]
- 7.273.3.4 `int32 WStrPtrLen::operator!= (const WStrPtrLen & rhs) const` [inline]
- 7.273.3.5 `WStrPtrLen& WStrPtrLen::operator= (const oscl_wchar * rhs)` [inline]
- 7.273.3.6 `WStrPtrLen& WStrPtrLen::operator= (const WStrPtrLen & rhs)` [inline]
- 7.273.3.7 `int32 WStrPtrLen::operator== (const WStrPtrLen & rhs) const` [inline]
- 7.273.3.8 `void WStrPtrLen::setPtrLen (const oscl_wchar * newPtr, uint32 newLen)` [inline]
- 7.273.3.9 `int32 WStrPtrLen::size () const` [inline]

7.273.4 Field Documentation

- 7.273.4.1 `int32 WStrPtrLen::len` [protected]
- 7.273.4.2 `const oscl_wchar* WStrPtrLen::ptr` [protected]

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

- [oscl_str_ptr_len.h](#)

Chapter 8

oscl File Documentation

8.1 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 [OsclAOSStatus](#)

Variables

- const int32 [OSCL_REQUEST_ERR_NONE](#) = 0
- const int32 [OSCL_REQUEST_PENDING](#) = (-0x7ffffff)
- const int32 [OSCL_REQUEST_ERR_CANCEL](#) = (-1)
- const int32 [OSCL_REQUEST_ERR_GENERAL](#) = (-2)

8.1.1 Detailed Description

Some basic types used with active objects.

8.2 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.

8.2.1 Detailed Description

The file [oscl_assert.h](#) provides an OSCL_ASSERT macro to document assumptions and test them during development.

8.3 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](#) ()

8.3.1 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.

8.4 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](#)

8.4.1 Detailed Description

A basic allocator that does not rely on other modules.

8.5 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_VIRTUAL_BASE](#)(type) type
- #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"

8.5.1 Detailed Description

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

8.6 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 [OscBinIStream](#)
- class [OscBinIStreamBigEndian](#)
- class [OscBinIStreamLittleEndian](#)
- class [OscBinOStream](#)
 - Class [OscBinOStream](#) implements the basic stream functions for an output stream.*
- class [OscBinOStreamBigEndian](#)
 - Class [OscBinOStreamBigEndian](#) implements a binary output stream using big endian byte ordering.*
- class [OscBinOStreamLittleEndian](#)
 - Class [OscBinOStreamLittleEndian](#) implements a binary output stream using little endian byte ordering.*
- class [OscBinStream](#)

8.6.1 Detailed Description

Defines a set of binary stream classes which handle portable input / output of binary data regardless of the native byte order.

8.7 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.

8.7.1 Detailed Description

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

8.8 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__)

8.8.1 Detailed Description

The file defines simple default memory allocator classes. These allocators are used by the [OscL_Vector](#) and [OscL_Map](#) class, etc.

8.9 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](#)()

8.9.1 Detailed Description

Defines a DLL entry point.

8.10 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 [OscIDNS](#)
- class [OscIDNSObserver](#)

Enumerations

- enum [TPVDNSFxn](#) { [EPVDNSGetHostByName](#) }
- enum [TPVDNSEvent](#) { [EPVDNSSuccess](#), [EPVDNSPending](#), [EPVDNSTimeout](#), [EPVDNSFailure](#), [EPVDNSCancel](#) }

8.10.1 Detailed Description

The file [oscl_socket.h](#) defines the OSCL DNS APIs.

8.11 oscl_dns_gethostbyname.h File Reference

```
#include "oscl_dns_method.h"  
#include "oscl_dns.h"  
#include "osclconfig_io.h"
```

Data Structures

- class [OscGetHostByNameMethod](#)
- class [OscGetHostByNameRequest](#)

8.12 oscl_dns_imp.h File Reference

```
#include "oscl_dns_tuneables.h"  
#include "oscl_dns_imp_pv.h"
```

8.13 oscl_dns_imp_base.h File Reference

```
#include "oscl_socket_imp.h"  
#include "oscl_dns_request.h"  
#include "oscl_dns.h"
```

Data Structures

- class [OscDNSIBase](#)

8.14 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 [OscIDNSI](#)

8.15 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 [OscDNSMethod](#)
- class [OscDNSRequestAO](#)

8.16 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 [OscMemAllocator](#) [TDNSRequestParamAllocator](#)

8.16.1 Typedef Documentation

8.16.1.1 typedef [OscMemAllocator](#) [TDNSRequestParamAllocator](#)

8.17 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](#)

8.18 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

8.18.1 Define Documentation

8.18.1.1 #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.

8.18.1.2 #define [PV_DNS_SERVER](#) 1

Enable/disable the PV DNS server here.

8.19 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)

8.19.1 Detailed Description

Internal use types for scheduler.

8.20 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.

8.20.1 Detailed Description

Defines functions to access additional information on errors where supported through an errno or similar service.

8.21 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 [OscError](#)
- class [OscErrorTrap](#)
- class [OscTLSEx](#)
- class [OscTLSRegistryEx](#)

Defines

- #define [OSCL_TRAPSTACK_PUSH\(a\)](#) OscError::PushL(a)
- #define [OSCL_TRAPSTACK_POP\(\)](#) OscError::Pop()
- #define [OSCL_TRAPSTACK_POPDEALLOC\(\)](#) OscError::PopDealloc()

8.21.1 Detailed Description

OSCL Error trap and cleanup include file.

8.22 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.

8.22.1 Detailed Description

Defines a memory allocation class used by the oscl error layer.

8.23 oscl_error_codes.h File Reference

Defines basic error and leave codes.

Defines

- #define [OscErrNone](#) 0
- #define [OscErrGeneral](#) 100
- #define [OscErrNoMemory](#) 101
- #define [OscErrCancelled](#) 102
- #define [OscErrNotSupported](#) 103
- #define [OscErrArgument](#) 104
- #define [OscErrBadHandle](#) 105
- #define [OscErrAlreadyExists](#) 106
- #define [OscErrBusy](#) 107
- #define [OscErrNotReady](#) 108
- #define [OscErrCorrupt](#) 109
- #define [OscErrTimeout](#) 110
- #define [OscErrOverflow](#) 111
- #define [OscErrUnderflow](#) 112
- #define [OscErrInvalidState](#) 113
- #define [OscErrNoResources](#) 114
- #define [OscErrNotInstalled](#) 115
- #define [OscErrAlreadyInstalled](#) 116
- #define [OscErrSystemCallFailed](#) 117
- #define [OscErrNoHandler](#) 118
- #define [OscErrThreadContextIncorrect](#) 119
- #define [OSCL_ERR_NONE](#) [OscErrNone](#)
- #define [OSCL_BAD_ALLOC_EXCEPTION_CODE](#) [OscErrNoMemory](#)
- #define [OscSuccess](#) 0
- #define [OscPending](#) 1
- #define [OscFailure](#) -1

Typedefs

- typedef int32 [OscLeaveCode](#)
- typedef int32 [OscReturnCode](#)

8.23.1 Detailed Description

Defines basic error and leave codes.

8.24 oscl_error_imp.h File Reference

Internal error implementation support.

```
#include "osclconfig_error.h"  
#include "oscl_error_imp_jumps.h"
```

Defines

- #define [PVEERROR_IMP_JUMPS](#)

8.24.1 Detailed Description

Internal error implementation support.

8.25 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)

8.25.1 Detailed Description

Implementation File for Leave using C++ exceptions.

8.26 oscl_error_imp_fatalerror.h File Reference

Implementation File for Leave using system fatal error.

```
#include "oscl_assert.h"
```

Defines

- #define [PVErrDoLeave\(\)](#) [_OSCL_Abort\(\)](#)
- #define [_PV_TRAP\(__r, __s\)](#)
- #define [_PV_TRAP_NO_TLS\(__tr, __r, __s\)](#)

8.26.1 Detailed Description

Implementation File for Leave using system fatal error.

8.26.2 Define Documentation

8.26.2.1 #define [_PV_TRAP\(__r, __s\)](#)

Value:

```
__r=OscErrNone;\n  {__s;}
```

8.26.2.2 #define [_PV_TRAP_NO_TLS\(__tr, __r, __s\)](#)

Value:

```
__r=OscErrNone;\n  {__s;}
```

8.26.2.3 #define [PVErrDoLeave\(\)](#) [_OSCL_Abort\(\)](#)

8.27 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\(__trapimp, __r, __s\)](#)

8.27.1 Detailed Description

Implementation of using Setjmp / Longjmp.

8.27.2 Define Documentation

8.27.2.1 #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();\
    }\
}
```

8.27.2.2 #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();\
    }\
}
```

8.27.2.3 #define PVErrDoLeave() OsclJump::StaticJump([internalLeave](#))

8.28 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 [OscErrorTrapImp](#)
- class [OscTrapStack](#)
- class [OscTrapStackItem](#)

Defines

- #define [OSCL_MAX_TRAP_LEVELS](#) 20
- #define [PVEERRORTRAP_REGISTRY_ID](#) [OSCL_TLS_ID_PVEERRORTRAP](#)
- #define [PVEERRORTRAP_REGISTRY](#) [OscTLSRegistry](#)

8.28.1 Detailed Description

OSCL Error trap and cleanup implementation include file.

8.29 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.

8.29.1 Detailed Description

contains all the exception handling macros and classes

8.30 oscl_exclusive_ptr.h File Reference

This file defines the [OscExclusivePtr](#) 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 [OscExclusiveArrayPtr](#)

The [OscExclusiveArrayPtr](#) 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 [OscExclusiveArrayPtr](#) expires, its destructor uses delete to free the memory.

- class [OscExclusivePtr](#)

The [OscExclusivePtr](#) 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 [OscExclusivePtr](#) expires, its destructor uses delete to free the memory.

- class [OscExclusivePtrA](#)

The [OscExclusivePtrA](#) 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 [OscExclusivePtrA](#) expires, Alloc is used to free the memory.

8.30.1 Detailed Description

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

8.31 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](#)

8.32 oscl_file_cache.h File Reference

The file [oscl_file_cache.h](#) defines the class [OscFileCache](#).

```
#include "osclconfig_io.h"
#include "oscl_base.h"
#include "oscl_file_io.h"
```

Data Structures

- class [OscFileCache](#)
- class [OscFileCacheBuffer](#)

8.32.1 Detailed Description

The file [oscl_file_cache.h](#) defines the class [OscFileCache](#).

8.33 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)

8.33.1 Detailed Description

The file [oscl_file_dir_utils.h](#) defines some unix-style directory ops.

8.34 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](#)

8.34.1 Detailed Description

The file [oscl_file_find.h](#) defines the class [OscL_FileFind](#).

8.35 oscl_file_handle.h File Reference

The file [oscl_file_handle.h](#) defines the class [OscFileHandle](#).

```
#include "osclconfig_io.h"
#include "oscl_base.h"
```

Data Structures

- class [OscFileHandle](#)

Typedefs

- typedef FILE * [TOscFileHandle](#)

8.35.1 Detailed Description

The file [oscl_file_handle.h](#) defines the class [OscFileHandle](#).

8.36 oscl_file_io.h File Reference

The file [oscl_file_io.h](#) defines the class [OscFile](#). 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 [OscFile](#)
- class [OscFixedCacheParam](#)
- class [OscCacheObserver](#)

Defines

- #define [TOscFileOffsetInt32](#) int32

8.36.1 Detailed Description

The file [oscl_file_io.h](#) defines the class [OscFile](#). This is the public API to the basic file I/O operations.

8.37 oscl_file_manager.h File Reference

File management class.

```
#include "oscl_base.h"
```

Data Structures

- class [OscFileManager](#)

8.37.1 Detailed Description

File management class.

8.38 oscl_file_native.h File Reference

The file [oscl_file_native.h](#) defines the class [OscNativeFile](#). 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 [OscNativeFile](#)

8.38.1 Detailed Description

The file [oscl_file_native.h](#) defines the class [OscNativeFile](#). This is the porting layer for basic file I/O operations.

8.39 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](#)

8.39.1 Detailed Description

The file [oscl_file_server.h](#) defines the class [OscL_FileServer](#). This is the porting layer for file server implementations.

8.40 oscl_file_stats.h File Reference

File stats class.

```
#include "oscl_base.h"
#include "osclconfig_io.h"
```

Data Structures

- class [OscFileStats](#)
- class [OscFileStatsItem](#)

Defines

- #define [OSCL_FILE_STATS_LOGGER_NODE](#) "OscFileStats"

Enumerations

- enum [TOscFileOp](#) { [EOscFileOp_Open](#), [EOscFileOp_Close](#), [EOscFileOp_Read](#), [EOscFileOp_Write](#), [EOscFileOp_Seek](#), [EOscFileOp_Tell](#), [EOscFileOp_Size](#), [EOscFileOp_Flush](#), [EOscFileOp_EndOfFile](#), [EOscFileOp_SetSize](#), [EOscFileOp_NativeOpen](#), [EOscFileOp_NativeClose](#), [EOscFileOp_NativeRead](#), [EOscFileOp_NativeWrite](#), [EOscFileOp_NativeSeek](#), [EOscFileOp_NativeTell](#), [EOscFileOp_NativeSize](#), [EOscFileOp_NativeFlush](#), [EOscFileOp_NativeEndOfFile](#), [EOscFileOp_NativeSetSize](#), [EOscFileOp_Last](#) }

8.40.1 Detailed Description

File stats class.

8.41 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 [OscNativeFileParams](#)

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("/")

8.41.1 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.

8.42 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 [_OscHeapBase](#)
- class [OscTrapItem](#)

Typedefs

- typedef void(* [OscTrapOperation](#))(OscAny *)

8.42.1 Detailed Description

OSCL Heap Base include file.

8.43 oscl_init.h File Reference

Global oscl initialization.

```
#include "oscl_base.h"  
#include <stdio.h>
```

Data Structures

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

8.43.1 Detailed Description

Global oscl initialization.

8.44 oscl_int64_utils.h File Reference

```
#include "oscl_base.h"
```

Data Structures

- class [Osc_Int64_Utils](#)
The `Osc_Int64_Utils` class provides a wrapper for commonly used `int64/uint64` operations.
- struct [OscInteger64Transport](#)

Typedefs

- typedef [OscInteger64Transport](#) [_OscInteger64Transport](#)

8.44.1 Typedef Documentation

8.44.1.1 typedef struct [OscInteger64Transport](#) [_OscInteger64Transport](#)

[OscInteger64Transport](#) 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.

8.45 oscl_ip_socket.h File Reference

```
#include "oscl_socket_types.h"  
#include "oscl_vector.h"  
#include "oscl_mem.h"
```

Data Structures

- class [OsclIPSocketI](#)

8.46 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](#)

8.46.1 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.

8.47 oscl_lock_base.h File Reference

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

Data Structures

- class [OscLockBase](#)
- class [OscNullLock](#)
- class [OscScopedLock](#)

The [OscScopedLock](#) 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 [OscScopedLock](#) goes out of scope.

8.47.1 Detailed Description

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

8.48 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_SelectIst`
- class `value_compare`

Defines

- `#define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE`

8.48.1 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.

8.48.2 Define Documentation

8.48.2.1 `#define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE`

8.49 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)

8.49.1 Detailed Description

Provides math functions.

8.50 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 [BufFragGroup](#)
- class [MediaData](#)
- class [MemAllocator](#)

Typedefs

- typedef void(* [BufferFreeFuncPtr](#))(void *)
- typedef uint32 [MediaTimestamp](#)

8.50.1 Detailed Description

Defines a container class for media data made up of a collection of memory fragments.

8.51 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

8.51.1 Detailed Description

Defines a status values for the [MediaData](#) containers.

8.52 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 [OscIAuditCB](#)
- class [OscIMem](#)
- class [OscIMemAllocator](#)
- class [OscIMemAllocDestructDealloc](#)
- class [OscIMemBasicAllocator](#)
- class [OscIMemBasicAllocDestructDealloc](#)
- class [OscIMemGlobalAuditObject](#)

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_malloc(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_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_malloc (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_malloc (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)`

8.52.1 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.

8.52.2 Define Documentation

8.52.2.1 `#define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE`

Previously this was in `oscl_mem_imp.h`

8.52.3 Function Documentation

8.52.3.1 `void operator delete (void * aPtr)` [inline]

8.52.3.2 `void* operator new (size_t aSize)` [inline]

8.53 oscl_mem_align.h File Reference

8.54 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< OsciMemStatsNode, Osci_TAlloc< OsciMemStatsNode, OsciMemBasicAllocator > > OsciMemStatsNodeAutoPtr`
- typedef `Osci_TAlloc< MM_StatsNodeTagTreeType, OsciMemBasicAllocator > TagTree_Allocator`
- typedef `Osci_TagTree< MM_StatsNodeTagTreeType, TagTree_Allocator > OsciTagTreeType`

8.54.1 Detailed Description

This file contains the definition and partial implementation of MM_Audit class.

8.54.2 Define Documentation

8.54.2.1 `#define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE`

8.55 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

8.55.1 Detailed Description

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

8.55.2 Define Documentation

8.55.2.1 #define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE

8.56 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 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](#)

8.56.1 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.

8.56.2 Define Documentation

8.56.2.1 `#define` [OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE](#)

8.57 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)

8.57.1 Detailed Description

This file contains prototypes for the basic memory functions.

8.58 oscl_mem_inst.h File Reference

The file defines default memory instrumentation level.

```
#include "osclconfig_memory.h"
```

Defines

- #define [PVMEM_INST_LEVEL](#) 1

8.58.1 Detailed Description

The file defines default memory instrumentation level.

8.59 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 [OscMemPoolFixedChunkAllocator](#)
- class [OscMemPoolFixedChunkAllocatorObserver](#)
- class [OscMemPoolResizableAllocator](#)
- class [OscMemPoolResizableAllocatorMemoryObserver](#)
- class [OscMemPoolResizableAllocatorObserver](#)

8.59.1 Detailed Description

This file contains the definition of memory pool allocators.

8.60 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 [OscMemPoolAllocator](#)

8.60.1 Detailed Description

This file contains the definition of memory pool allocator for leave/trap.

8.61 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 [OscMutex](#)
- class [OscThreadLock](#)

Typedefs

- typedef [OscMutex](#) [OscNoYieldMutex](#)

8.61.1 Detailed Description

This file provides implementation of mutex.

8.61.2 Typedef Documentation

8.61.2.1 typedef [OscMutex](#) [OscNoYieldMutex](#)

Class [OscNoYieldMutex](#) can be used in use cases where there will be no CPU-yielding operation done while the [Mutex](#) is locked.

CPU-yielding operations include [OscMutex::Lock](#), [OscSemaphore::Wait](#), [OscThread::Sleep](#), and [OscBrewThreadUtil::BThreadYield](#).

The behavior of [OscNoYieldMutex](#) depends on whether the threading model is pre-emptive or not. When threading is pre-emptive, it is identical to [OscMutex](#). When threading is non-pre-emptive, it is a NO-OP.

An example of this type of use case is for simple data protection.

8.62 oscl_namestring.h File Reference

Name string class include file.

```
#include "oscl_base.h"
```

Data Structures

- class [OsclNameString](#)

8.62.1 Detailed Description

Name string class include file.

8.63 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](#)

8.63.1 Detailed Description

The file [oscl_opaque_type.h](#) defines pure virtual classes for working with opaque types.

8.64 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 [OscCompareLess](#)
- class [OscPriorityQueue](#)
- class [OscPriorityQueueBase](#)

8.64.1 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 $O(\log_2(N))$ insertion and deletion complexity and $O(1)$ complexity to access the top priority item.

8.65 oscl_procstatus.h File Reference

Data Structures

- class [OscProcStatus](#)

8.66 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](#)

8.66.1 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.

8.67 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 [OscRand](#)

8.67.1 Detailed Description

Provides pseudo-random number generation.

8.68 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_DefAllocWithRefCount](#)
- class [OscLRefCount](#)
- class [OscLRefCountDA](#)
- class [OscLRefCountMTDA](#)
- class [OscLRefCountMTSA](#)
- class [OscLRefCountSA](#)

8.68.1 Detailed Description

A general purpose reference counter to object lifetimes.

8.69 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 manage its lifetime through the refcount.

```
#include "oscl_base.h"
#include "oscl_refcounter.h"
```

Data Structures

- class [OscRefCounterMemFrag](#)

8.69.1 Detailed Description

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

8.70 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](#)

8.70.1 Detailed Description

Client-side implementation Registry Access implementation.

8.71 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](#)

8.71.1 Detailed Description

Client-side implementation of OsclRegistry.

8.72 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](#)

8.72.1 Detailed Description

Client-side implementation of OsclRegistryInterface.

8.73 oscl_registry_serv_impl.h File Reference

Server-side implementation of OslRegistry 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 [OslComponentRegistry](#)
- class [OslComponentRegistryData](#)
- class [OslComponentRegistryElement](#)

8.73.1 Detailed Description

Server-side implementation of OslRegistry interfaces.

8.74 oscl_registry_serv_impl_global.h File Reference

```
#include "osclconfig_proc.h"  
#include "oscl_base.h"
```


8.75 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](#)

8.76 oscl_registry_types.h File Reference

Common types used in Osl registry interfaces.

```
#include "oscl_types.h"
#include "oscl_string_containers.h"
```

Data Structures

- class [OslRegistryAccessElement](#)

Typedefs

- typedef [OslAny](#) * [OslComponentFactory](#)

8.76.1 Detailed Description

Common types used in Osl registry interfaces.

8.77 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 [OscExecScheduler](#)
- class [OscExecSchedulerCommonBase](#)
- class [OscScheduler](#)
- class [OscSchedulerObserver](#)
- class [PVSchedulerStopper](#)

Defines

- #define [PVSCHEDNAMELEN](#) 30

8.78 oscl_scheduler_ao.h File Reference

Osc Scheduler user execution object classes.

```
#include "oscl_scheduler_aobase.h"  
#include "oscl_mem.h"  
#include "oscl_scheduler_types.h"
```

Data Structures

- class [OscActiveObject](#)
- class [OscTimerObject](#)

8.78.1 Detailed Description

Osc Scheduler user execution object classes.

8.79 oscl_scheduler_aobase.h File Reference

Osc 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 [PVEXECNAMELEN](#) 30

8.79.1 Detailed Description

Osc Scheduler internal active object classes.

8.80 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 [OscReadyAlloc](#)
- class [OscReadyCompare](#)
- class [OscReadyQ](#)
- class [OscTimerCompare](#)
- class [OscTimerQ](#)
- class [TReadyQueLink](#)

Typedefs

- typedef [PActiveBase](#) * [TOscReady](#)

8.80.1 Detailed Description

ready q types for oscl scheduler

8.81 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_OscIThread](#), [EPVThreadContext_NonOscIThread](#), [EPVThreadContext_Undetermined](#) }

8.81.1 Detailed Description

Thread context functions needed by oscl scheduler.

8.82 oscl_scheduler_tuneables.h File Reference

Tunable settings for Osl 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

8.82.1 Detailed Description

Tunable settings for Osl Scheduler.

8.83 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 [OscExecSchedulerBase](#)

8.83.1 Detailed Description

Scheduler common types include file.

8.84 oscl_semaphore.h File Reference

This file provides implementation of mutex.

```
#include "osclconfig_proc.h"  
#include "oscl_thread.h"
```

Data Structures

- class [OsclSemaphore](#)

8.84.1 Detailed Description

This file provides implementation of mutex.

8.85 oscl_shared_ptr.h File Reference

This file defines a template class [OscSharedPtr](#) 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 [OscSharedPtr](#)
A parameterized smart pointer class.

Defines

- #define [OSCL_DISABLE_WARNING_RETURN_TYPE_NOT_UDT](#)

8.85.1 Detailed Description

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

8.86 oscl_singleton.h File Reference

This file defines the [OscSingleton](#) 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 [OscSingleton](#)
- class [OscSingletonRegistry](#)
- 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_PVMFRECIGNIZER](#) = 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_WMDRMLOCK](#) = 13
- const uint32 [OSCL_SINGLETON_ID_LAST](#) = 14

8.86.1 Detailed Description

This file defines the [OscSingleton](#) 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.

[OscSingleton](#) is initialized in `OscBase::Init`.

8.86.2 Variable Documentation

- 8.86.2.1 `const uint32 OSCL_SINGLETON_ID_CPM_PLUGIN = 7`
- 8.86.2.2 `const uint32 OSCL_SINGLETON_ID_LAST = 14`
- 8.86.2.3 `const uint32 OSCL_SINGLETON_ID_OMX = 10`
- 8.86.2.4 `const uint32 OSCL_SINGLETON_ID_OMXMASTERCORE = 11`
- 8.86.2.5 `const uint32 OSCL_SINGLETON_ID_OSCLMEM = 1`
- 8.86.2.6 `const uint32 OSCL_SINGLETON_ID_OSCLREGISTRY = 9`
- 8.86.2.7 `const uint32 OSCL_SINGLETON_ID_PAYLOADPARSER = 6`
- 8.86.2.8 `const uint32 OSCL_SINGLETON_ID_PVERRORTRAP = 4`
- 8.86.2.9 `const uint32 OSCL_SINGLETON_ID_PVLOGGER = 2`
- 8.86.2.10 `const uint32 OSCL_SINGLETON_ID_PVMFRECOGNIZER = 8`
- 8.86.2.11 `const uint32 OSCL_SINGLETON_ID_PVSCHEDULER = 3`
- 8.86.2.12 `const uint32 OSCL_SINGLETON_ID_SDPMEDIAPARSER = 5`
- 8.86.2.13 `const uint32 OSCL_SINGLETON_ID_TEST = 0`
- 8.86.2.14 `const uint32 OSCL_SINGLETON_ID_TICKCOUNT = 12`
- 8.86.2.15 `const uint32 OSCL_SINGLETON_ID_WMDRMLOCK = 13`

8.87 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)

8.87.1 Detailed Description

Provides a portable implementation of snprintf.

8.88 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 [OscSocketServ](#)
- class [OscTCPSocket](#)
- class [OscUDPSocket](#)

8.88.1 Detailed Description

The file [oscl_socket.h](#) defines the OSCL Socket APIs.

8.89 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](#)

8.90 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 [OscBindMethod](#)
- class [OscBindRequest](#)

8.91 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 [OscConnectMethod](#)
- class [OscConnectRequest](#)

8.92 oscl_socket_imp.h File Reference

```
#include "oscl_socket_tuneables.h"  
#include "oscl_socket_imp_pv.h"
```

8.93 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](#)

8.94 oscl_socket_imp_pv.h File Reference

```
#include "oscl_socket_imp_base.h"
```

Data Structures

- class [OscSocketI](#)

Defines

- #define [PVSOCK_ERR_BAD_PARAM](#) (-1)
- #define [PVSOCK_ERR_SOCKET_NOT_OPEN](#) (-2)
- #define [PVSOCK_ERR_SOCKET_NO_SERV](#) (-3)
- #define [PVSOCK_ERR_SERV_NOT_CONNECTED](#) (-4)
- #define [PVSOCK_ERR_SOCKET_NOT_CONNECTED](#) (-5)
- #define [PVSOCK_ERR_NOT_IMPLEMENTED](#) (-6)
- #define [PVSOCK_ERR_NOT_SUPPORTED](#) (-7)

8.94.1 Define Documentation

8.94.1.1 #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.

8.94.1.2 #define PVSOCK_ERR_NOT_IMPLEMENTED (-6)

8.94.1.3 #define PVSOCK_ERR_NOT_SUPPORTED (-7)

8.94.1.4 #define PVSOCK_ERR_SERV_NOT_CONNECTED (-4)

8.94.1.5 #define PVSOCK_ERR_SOCKET_NO_SERV (-3)

8.94.1.6 #define PVSOCK_ERR_SOCKET_NOT_CONNECTED (-5)

8.94.1.7 #define PVSOCK_ERR_SOCKET_NOT_OPEN (-2)

8.95 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 [OscListenMethod](#)
- class [OscListenRequest](#)

Defines

- #define [OSCL_SOCKET_LISTEN_H_INCLUDEDd](#)

8.95.1 Define Documentation

8.95.1.1 #define OSCL_SOCKET_LISTEN_H_INCLUDEDd

8.96 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 [OscSocketMethod](#)
- class [OscSocketRequestAO](#)

Defines

- #define [MSEC_TO_MICROSEC](#) 1000

8.96.1 Define Documentation

8.96.1.1 #define MSEC_TO_MICROSEC 1000

8.97 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 [OscRecvMethod](#)
- class [OscRecvRequest](#)

8.98 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 [OscRecvFromMethod](#)
- class [OscRecvFromRequest](#)

8.99 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 [OscSocketRequest](#)
- class [PVSockBufRecv](#)
- class [PVSockBufSend](#)
- class [RecvFromParam](#)
- class [RecvParam](#)
- class [SendParam](#)
- class [SendToParam](#)
- class [ShutdownParam](#)
- class [SocketRequestParam](#)

8.100 oscl_socket_send.h File Reference

```
#include "oscl_socket_types.h"  
#include "oscl_socket_method.h"
```

Data Structures

- class [OscSendMethod](#)
- class [OscSendRequest](#)

8.101 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 [OscSendToMethod](#)
- class [OscSendToRequest](#)

8.102 oscl_socket_serv_imp.h File Reference

```
#include "osclconfig_io.h"  
#include "oscl_socket_tuneables.h"  
#include "oscl_socket_serv_imp_pv.h"
```

8.103 oscl_socket_serv_imp_base.h File Reference

```
#include "oscl_base.h"  
#include "oscl_socket_stats.h"
```

Data Structures

- class [OscSocketServIBase](#)

8.104 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

8.104.1 Define Documentation

8.104.1.1 #define OSCL_EXCEPTSET_FLAG 0x01

8.104.1.2 #define OSCL_READSET_FLAG 0x04

A bitmask for socket select operations

8.104.1.3 #define OSCL_WRITESET_FLAG 0x02

8.105 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](#)

8.106 oscl_socket_shutdown.h File Reference

```
#include "oscl_socket_types.h"  
#include "oscl_socket_method.h"
```

Data Structures

- class [OscShutdownMethod](#)
- class [OscShutdownRequest](#)

8.107 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` { `EOsclSocket_RequestAO_Success`, `EOsclSocket_RequestAO_Canceled`, `EOsclSocket_RequestAO_Error`, `EOsclSocket_RequestAO_Timeout`, `EOsclSocket_ServRequestIssued`, `EOsclSocket_ServPoll`, `EOsclSocket_OS`, `EOsclSocket_Readable`, `EOsclSocket_Writable`, `EOsclSocket_Except`, `EOsclSocket_DataRecv`, `EOsclSocket_DataSent`, `EOsclSocket_ServRequestComplete`, `EOsclSocket_ServRequestCancelIssued`, `EOsclSocketServ_LoopssockOk`, `EOsclSocketServ_LoopssockError` }
- enum `TOsclSocketServStatEvent` { `EOsclSocketServ_SelectNoActivity` = 0, `EOsclSocketServ_SelectActivity`, `EOsclSocketServ_SelectRescheduleAsap`, `EOsclSocketServ_SelectReschedulePoll`, `EOsclSocketServ_LastEvent` }

8.107.1 Enumeration Type Documentation

8.107.1.1 enum `TOsclSocketServStatEvent`

Enumeration values:

`EOsclSocketServ_SelectNoActivity`
`EOsclSocketServ_SelectActivity`
`EOsclSocketServ_SelectRescheduleAsap`
`EOsclSocketServ_SelectReschedulePoll`
`EOsclSocketServ_LastEvent`

8.107.1.2 enum `TOsclSocketStatEvent`

Socket diagnostics.

Enumeration values:

`EOsclSocket_RequestAO_Success`
`EOsclSocket_RequestAO_Canceled`
`EOsclSocket_RequestAO_Error`
`EOsclSocket_RequestAO_Timeout`
`EOsclSocket_ServRequestIssued`
`EOsclSocket_ServPoll`
`EOsclSocket_OS`
`EOsclSocket_Readable`
`EOsclSocket_Writable`

EOsclSocket_Except
EOsclSocket_DataRecv
EOsclSocket_DataSent
EOsclSocket_ServRequestComplete
EOsclSocket_ServRequestCancelIssued
EOsclSocketServ_LoopsockOk
EOsclSocketServ_LoopsockError

8.108 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

8.108.1 Define Documentation

8.108.1.1 #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.

8.108.1.2 #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.

8.108.1.3 #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.

8.108.1.4 #define PV_SOCKET_REQUEST_AO_PRIORITY OsclActiveObject::EPriority-Nominal

`PV_SOCKET_REQUEST_AO_PRIORITY` sets the priority of the socket request completion AOs.

8.108.1.5 #define PV_SOCKET_SERVER 1

Enable/disable the PV socket server here.

8.108.1.6 #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.

8.108.1.7 #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.

8.108.1.8 #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

8.108.1.9 #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.

8.108.1.10 #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 loop option of threaded server mode. This option is forced to "0" in AO mode.

8.108.1.11 #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.

8.108.1.12 #define PV_SOCKET_SERVER_THREAD_PRIORITY ThreadPriorityAboveNormal

PV_SOCKET_SERVER_THREAD_PRIORITY sets the priority of the PV socket server thread.

8.108.1.13 #define PV_SOCKET_SERVER_STATS 0

For detailed performance breakdown of time spend in [OsclSocketServI](#) AO. Output is logged under "Oscl-SchedulerPerfStats" node. Should be off in production code. This option is forced to "0" in threaded mode.

8.109 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 [OscIpMReq](#)
- class [OscNetworkAddress](#)
- class [OscSocketObserver](#)
- class [OscSocketTOS](#)

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](#), [EPVSocketNotImplemented](#) }
- enum [TPVSocketShutdown](#) { [EPVSocketSendShutdown](#), [EPVSocketRecvShutdown](#), [EPVSocketBothShutdown](#) }
- enum [TPVSocketOptionName](#) { [EPVIPMulticastTTL](#), [EPVIPAddMembership](#), [EPVIPTOS](#), [EPVSockReuseAddr](#) }
- enum [TPVSocketOptionLevel](#) { [EPVIPProtoIP](#), [EPVIPProtoTCP](#), [EPVSocket](#) }

8.109.1 Define Documentation

8.109.1.1 #define PVNETWORKADDRESS_LEN 50

8.109.2 Enumeration Type Documentation

8.109.2.1 enum TPVSocketEvent

Return codes for asynchronous APIs

Enumeration values:

EPVSocketSuccess

EPVSocketPending

EPVSocketTimeout

EPVSocketFailure

EPVSocketCancel
EPVSocketNotImplemented

8.109.2.2 enum TPVSocketFxn

Enumeration values:

EPVSocketSend
EPVSocketSendTo
EPVSocketRecv
EPVSocketRecvFrom
EPVSocketConnect
EPVSocketAccept
EPVSocketShutdown
EPVSocketBind
EPVSocketListen
EPVSocket_Last

8.109.2.3 enum TPVSocketOptionLevel

Enumeration values:

EPVIPProtoIP
EPVIPProtoTCP
EPVSocket

8.109.2.4 enum TPVSocketOptionName

Enumeration values:

EPVIPMulticastTTL
EPVIPAddMembership
EPVIPTOS
EPVSockReuseAddr

8.109.2.5 enum TPVSocketShutdown

Enumeration values:

EPVSocketSendShutdown
EPVSocketRecvShutdown
EPVSocketBothShutdown

8.110 oscl_stdstring.h File Reference

This file provides standard string operations such as strlen, strcpy, etc. ANSI defines undefined behavior when the destination pointer is null for operations such as strcpy, strcat, 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_stremp](#) (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_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 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_CIstremp](#) (const char *str1, const char *str2)
- OSCL_IMPORT_REF int32 [oscl_CIstremp](#) (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_strcat](#) (char *dest, const char *src)
- OSCL_IMPORT_REF [oscl_wchar](#) * [oscl_strcat](#) ([oscl_wchar](#) *dest, const [oscl_wchar](#) *src)

8.110.1 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.

8.111 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

8.111.1 Detailed Description

Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.

8.112 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](#)

Enumerations

- enum [TOSCL_StringOp](#) { [EOSCL_StringOp_CompressASCII](#), [EOSCL_StringOp_UTF16ToUTF8](#) }
- enum [TOSCL_wStringOp](#) { [EOSCL_wStringOp_ExpandASCII](#), [EOSCL_wStringOp_UTF8ToUTF16](#) }

8.112.1 Detailed Description

Provides a standardized set of string containers that can be used in place of character arrays.

8.113 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](#)

8.113.1 Detailed Description

Provides a standardized set of string containers that can be used in place of character arrays.

8.114 oscl_string_rep.h File Reference

Contains some internal implementation for string containers.

```
#include "oscl_defalloc.h"
```

Data Structures

- class [CFastRep](#)
- class [CHeapRep](#)
- class [CStackRep](#)

8.114.1 Detailed Description

Contains some internal implementation for string containers.

8.115 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

8.115.1 Detailed Description

Utilities to unescape URIs.

8.116 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.

8.116.1 Detailed Description

Utilities to validate and truncate UTF-8 encoded strings.

8.117 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, `OscFloat` &value)
- OSCL_IMPORT_REF bool `PV_atof` (const char *buf, int length, `OscFloat` &value)
- OSCL_IMPORT_REF int `oscl_abs` (int aVal)

8.117.1 Detailed Description

Utilities to parse and convert strings.

8.118 oscl_string_xml.h File Reference

Utilities to escape special characters in XML strings.

```
#include "oscl_base.h"
```

Functions

- OSDL_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.
- OSDL_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: &, <, >, ', ".

8.118.1 Detailed Description

Utilities to escape special characters in XML strings.

8.119 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](#)

8.119.1 Detailed Description

The file [oscl_tagtree.h](#) ...

8.119.2 Define Documentation

8.119.2.1 `#define` [OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE](#)

8.120 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 [OscITCPSocketf](#)

8.121 oscl_thread.h File Reference

```
#include "osclconfig_proc.h"
#include "oscl_procstatus.h"
#include "oscl_base.h"
```

Data Structures

- class [OscThread](#)

Typedefs

- typedef [TOscThreadFuncRet](#)(OSCL_THREAD_DECL * [TOscThreadFuncPtr](#))(TOscThreadFuncArg)

Enumerations

- enum [OscThread_State](#) { [Start_on_creation](#), [Suspend_on_creation](#) }
- enum [OscThreadPriority](#) { [ThreadPriorityLowest](#), [ThreadPriorityLow](#), [ThreadPriorityBelowNormal](#), [ThreadPriorityNormal](#), [ThreadPriorityAboveNormal](#), [ThreadPriorityHighest](#), [ThreadPriorityTimeCritical](#) }

8.121.1 Detailed Description

This file provides THREAD implementation that can be ported to three OS LINUX, SYMBIAN, WIN32

8.121.2 Typedef Documentation

8.121.2.1 typedef [TOscThreadFuncRet](#)(OSCL_THREAD_DECL * [TOscThreadFuncPtr](#))([TOscThreadFuncArg](#))

8.121.3 Enumeration Type Documentation

8.121.3.1 enum [OscThread_State](#)

Enumeration values:

[Start_on_creation](#)

[Suspend_on_creation](#)

8.121.3.2 enum [OscThreadPriority](#)

Enumeration values:

[ThreadPriorityLowest](#)

[ThreadPriorityLow](#)

[ThreadPriorityBelowNormal](#)

ThreadPriorityNormal

ThreadPriorityAboveNormal

ThreadPriorityHighest

ThreadPriorityTimeCritical

8.122 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 [OscTickCount](#)

Defines

- #define [OSCLTICKCOUNT_MAX_TICKS](#) 0xffffffff

8.122.1 Detailed Description

Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.

8.123 oscl_time.h File Reference

The file `oscl_time.h` defines to 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`]
- typedef char `ISO8601timeStrBuf` [`ISO8601TIME_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 `ISO8601ToRFC822` (`ISO8601timeStrBuf` iso8601_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)
- OSCL_COND_IMPORT_REF `TimeValue` operator+ (const `TimeValue` &a, const int32 bSeconds)
- OSCL_COND_IMPORT_REF `TimeValue` operator+ (const int32 aSeconds, const `TimeValue` &b)
- OSCL_COND_IMPORT_REF `TimeValue` operator- (const `TimeValue` &a, const int32 bSeconds)
- OSCL_COND_IMPORT_REF `TimeValue` operator- (const int32 aSeconds, const `TimeValue` &b)

Variables

- const int `CTIME_BUFFER_SIZE` = 26
- const int `PV8601TIME_BUFFER_SIZE` = 21
- const int `ISO8601TIME_BUFFER_SIZE` = 21
- const long `USEC_PER_SEC` = 1000000
- const long `MSEC_PER_SEC` = 1000
- const uint32 `unix_ntp_offset` = 2208988800U

8.123.1 Detailed Description

The file `oscl_time.h` defines to 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.

8.124 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 [OscTimer](#)
- class [OscTimerObserver](#)

8.125 oscl_tls.h File Reference

```
#include "oscl_base.h"
#include "oscl_defalloc.h"
```

Data Structures

- class [OscTLS](#)
- class [OscTLSRegistry](#)
- 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 [OscAny](#) [TOscTlsKey](#)

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

8.126 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](#)

8.126.1 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.

8.126.2 Define Documentation

8.126.2.1 #define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE

8.127 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

8.127.1 Detailed Description

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

8.128 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 [OscUDPSocketI](#)

8.129 oscl_utf8conv.h File Reference

Utilities to convert unicode to utf8 and vice versa.

```
#include "oscl_base.h"
```

Defines

- #define [MAX_NUMBER_OF_BYTE_PER_UTF8](#) 3

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_UnicodeToUTF8](#) (const [oscl_wchar](#) *input, int32 inLength, char *output, int32 outLength)
Convert Unicode string to UTF8 byte sequence.

8.129.1 Detailed Description

Utilities to convert unicode to utf8 and vice versa.

8.130 oscl_uuid.h File Reference

This file defines the OSCL UUID structure used for unique identifiers as well as the short (32-bit) identifiers OslUuid32.

```
#include "oscl_base.h"
#include "oscl_mem_basic_functions.h"
#include "oscl_string_utils.h"
#include "oscl_stdstring.h"
```

Data Structures

- struct [OslUuid](#)

Defines

- #define [EMPTY_UUID](#) PVUuid(0,0,0,0,0,0,0,0,0,0)
- #define [BYTES_IN_UUID_ARRAY](#) 8

Typedefs

- typedef uint32 [OslUuid32](#)

Variables

- const char [PV_CHAR_CLOSE_BRACKET](#) = `)`
- const char [PV_CHAR_COMMA](#) = `,`

8.130.1 Detailed Description

This file defines the OSCL UUID structure used for unique identifiers as well as the short (32-bit) identifiers OslUuid32.

8.130.2 Define Documentation

8.130.2.1 `#define BYTES_IN_UUID_ARRAY 8`

8.130.2.2 `#define EMPTY_UUID PVUuid(0,0,0,0,0,0,0,0,0,0)`

8.130.3 Typedef Documentation

8.130.3.1 `typedef uint32 OsclUid32`

8.130.4 Variable Documentation

8.130.4.1 `const char PV_CHAR_CLOSE_BRACKET = ')`

8.130.4.2 `const char PV_CHAR_COMMA = ','`

8.131 oscl_vector.h File Reference

The file [oscl_vector.h](#) defines the template class [OscVector](#) 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 [OscVector](#)
- class [OscVectorBase](#)

8.131.1 Detailed Description

The file [oscl_vector.h](#) defines the template class [OscVector](#) 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.

8.132 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_EXPORT_REF](#) __attribute__((visibility("default")))
- #define [OSCL_IMPORT_REF](#) __attribute__((visibility("default")))
- #define [OSCL_RELEASE_BUILD](#) 0
- #define [PVLOGGER_INST_LEVEL](#) 5
- #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_HAS_PRAGMA_PACK](#) 0
- #define [OSCL_HAS_PACKED_STRUCT](#) 1
- #define [OSCL_PACKED_VAR\(x\) x __attribute__\(\(packed\)\)](#)
- #define [OSCL_PACKED_STRUCT_BEGIN](#)
- #define [OSCL_PACKED_STRUCT_END](#) __attribute__((packed))
- #define [OSCL_ASSERT_ALWAYS](#) 0

8.132.1 Detailed Description

This file contains configuration information for the linux platform.

8.132.2 Define Documentation

8.132.2.1 **#define** `__TFS__` `<>`

8.132.2.2 **#define** `OSCL_EXPORT_REF __attribute__((visibility("default")))`

8.132.2.3 **#define** `OSCL_HAS_ANDROID_FILE_IO_SUPPORT` `1`

8.132.2.4 **#define** `OSCL_HAS_ANDROID_SUPPORT` `1`

8.132.2.5 **#define** `OSCL_HAS_PACKED_STRUCT` `1`

8.132.2.6 **#define** `OSCL_IMPORT_REF __attribute__((visibility("default")))`

8.132.2.7 **#define** `OSCL_NATIVE_UINT64_TYPE` `u_int64_t`

8.132.2.8 **#define** `OSCL_PACKED_STRUCT_BEGIN`

8.132.2.9 **#define** `OSCL_PACKED_STRUCT_END __attribute__((packed))`

8.132.2.10 **#define** `OSCL_PACKED_VAR(x) x __attribute__((packed))`

8.132.2.11 **#define** `OSCL_RELEASE_BUILD` `0`

8.132.2.12 **#define** `OSCL_TEMPLATED_DESTRUCTOR_CALL(type, simple_type) ~type ()`

8.132.2.13 **#define** `OSCL_UNSIGNED_CONST(x) x##u`

8.132.2.14 **#define** `PVLOGGER_INST_LEVEL` `5`

8.133 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](#)

8.133.1 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 OSLC integer types.

8.133.2 Define Documentation

8.133.2.1 #define [OSCL_HAS_ANSI_MEMORY_FUNCS](#) 1

8.133.3 Typedef Documentation

8.133.3.1 typedef size_t [oscl_memsize_t](#)

8.134 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__](#)

8.135 osclconfig_compiler_warnings.h File Reference

This file contains the ability to turn off/on compiler warnings.

Defines

- #define `OSCL_FUNCTION_PTR(x) (&x)`

8.135.1 Detailed Description

This file contains the ability to turn off/on compiler warnings.

8.135.2 Define Documentation

8.135.2.1 #define `OSCL_FUNCTION_PTR(x) (&x)`

8.136 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

8.136.1 Detailed Description

This file contains the common typedefs and header files needed to compile osclerror.

8.136.2 Define Documentation

8.136.2.1 #define [OSCL_HAS_ERRNO_H](#) 1

8.136.2.2 #define [OSCL_HAS_EXCEPTIONS](#) 1

8.136.2.3 #define [OSCL_HAS_SETJMP_H](#) 1

8.136.2.4 #define [OSCL_HAS_SYMBIAN_ERRORTRAP](#) 0

8.137 osclconfig_error_check.h File Reference

8.138 osclconfig_global_new_delete.h File Reference

Functions

- void * [operator new](#) (size_t)
- void [operator delete](#) (void *)

8.139 osclconfig_global_placement_new.h File Reference

Functions

- void * [operator new](#) (size_t, void *ptr)

8.139.1 Function Documentation

8.139.1.1 void* [operator new](#) (size_t, void * *ptr*) [inline]

8.140 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 [OsciValidInetAddr\(addr\)](#) (inet_addr(addr)!=INADDR_NONE)
- #define [OsciMakeSockAddr\(sockaddr, port, addrstr, ok\)](#)
- #define [OsciUnMakeSockAddr\(sockaddr, addrstr\)](#) addrstr=inet_ntoa(sockaddr.sin_addr);
- #define [OsciMakeInAddr\(in_addr, addrstr, ok\)](#)
- #define [OsciUnMakeInAddr\(in_addr, addrstr\)](#) addrstr=inet_ntoa(in_addr);
- #define [OsciSetRecvBufferSize\(s, val, ok, err\)](#)
- #define [OsciBind\(s, addr, ok, err\)](#)

- #define [OscSetSockOpt](#)(s, optLevel, optName, optVal, optLen, ok, err)
- #define [OscJoin](#)(s, addr, ok, err)
- #define [OscListen](#)(s, size, ok, err)
- #define [OscAccept](#)(s, accept_s, ok, err, wouldblock)
- #define [OscSetNonBlocking](#)(s, ok, err)
- #define [OscShutdown](#)(s, how, ok, err)
- #define [OscSocket](#)(s, fam, type, prot, ok, err)
- #define [OscSendTo](#)(s, buf, len, addr, ok, err, nbytes, wouldblock)
- #define [OscSend](#)(s, buf, len, ok, err, nbytes, wouldblock)
- #define [OscCloseSocket](#)(s, ok, err)
- #define [OscConnect](#)(s, addr, ok, err, wouldblock)
- #define [OscGetPeerName](#)(s, name, namelen, ok, err)
- #define [OscGetAsyncSockErr](#)(s, ok, err)
- #define [OscPipe](#)(x) pipe(x)
- #define [OscReadFD](#)(fd, buf, cnt) read(fd,buf,cnt)
- #define [OscWriteFD](#)(fd, buf, cnt) write(fd,buf,cnt)
- #define [OscConnectComplete](#)(s, wset, eset, success, fail, ok, err)
- #define [OscRecv](#)(s, buf, len, ok, err, nbytes, wouldblock)
- #define [OscRecvFrom](#)(s, buf, len, paddr, paddrlen, ok, err, nbytes, wouldblock)
- #define [OscSocketSelect](#)(nfd, rd, wr, ex, timeout, ok, err, nhandles)
- #define [OscSocketStartup](#)(ok)
- #define [OscSocketCleanup](#)(ok)
- #define [OscGethostbyname](#)(name, hostent, ok, err)
- #define [OscGetDottedAddr](#)(hostent, dottedaddr, ok)
- #define [OscGetDottedAddrVector](#)(hostent, dottedaddr, dottedaddrvect, 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_IP](#) IPPROTO_IP
- #define [OSCL_IPPROTO_TCP](#) IPPROTO_TCP
- #define [OSCL_IPPROTO_UDP](#) IPPROTO_UDP
- #define [OSCL_SOL_SOCKET](#) SOL_SOCKET
- #define [OSCL_SOL_IP](#) IPPROTO_IP
- #define [OSCL_SOL_TCP](#) IPPROTO_TCP
- #define [OSCL_SOL_UDP](#) IPPROTO_UDP
- #define [OSCL SOCKOPT_IP_MULTICAST_TTL](#) IP_MULTICAST_TTL
- #define [OSCL SOCKOPT_IP_ADDMEMBERSHIP](#) IP_ADD_MEMBERSHIP
- #define [OSCL SOCKOPT_IP_TOS](#) IP_TOS
- #define [OSCL SOCKOPT_SOL_REUSEADDR](#) SO_REUSEADDR
- #define [_FILE_OFFSET_BITS](#) 64

Typedefs

- typedef int [TOscSocket](#)
- typedef sockaddr_in [TOscSockAddr](#)
- typedef socklen_t [TOscSockAddrLen](#)
- typedef ip_mreq [TipMReq](#)
- typedef hostent [TOscHostent](#)
- typedef off_t [TOscFileOffset](#)

8.140.1 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.

8.140.2 Define Documentation

8.140.2.1 `#define _FILE_OFFSET_BITS 64`

8.140.2.2 `#define OSCL_AF_INET AF_INET`

8.140.2.3 `#define OSCL_FILE_BUFFER_MAX_SIZE 32768`

8.140.2.4 `#define OSCL_HAS_ANSI_64BIT_FILE_IO_SUPPORT 0`

8.140.2.5 `#define OSCL_HAS_ANSI_FILE_IO_SUPPORT 1`

8.140.2.6 `#define OSCL_HAS_BERKELEY_SOCKETS 1`

8.140.2.7 `#define OSCL_HAS_GLOB 0`

8.140.2.8 `#define OSCL_HAS_LARGE_FILE_SUPPORT 1`

8.140.2.9 `#define OSCL_HAS_MSWIN_FILE_IO_SUPPORT 0`

8.140.2.10 `#define OSCL_HAS_NATIVE_FILE_CACHE_ENABLE 1`

8.140.2.11 `#define OSCL_HAS_PV_FILE_CACHE 0`

8.140.2.12 `#define OSCL_HAS_SOCKET_SUPPORT 1`

8.140.2.13 `#define OSCL_HAS_SYMBIAN_COMPATIBLE_IO_FUNCTION 0`

8.140.2.14 `#define OSCL_HAS_SYMBIAN_DNS_SERVER 0`

8.140.2.15 `#define OSCL_HAS_SYMBIAN_SOCKET_SERVER 0`

8.140.2.16 `#define OSCL_IPPROTO_IP IPPROTO_IP`

8.140.2.17 `#define OSCL_IPPROTO_TCP IPPROTO_TCP`

8.140.2.18 `#define OSCL_IPPROTO_UDP IPPROTO_UDP`

8.140.2.19 `#define OSCL_SD_BOTH SHUT_RDWR`

8.140.2.20 `#define OSCL_SD_RECEIVE SHUT_RD`

8.140.2.21 `#define OSCL_SD_SEND SHUT_WR`

8.140.2.22 `#define OSCL SOCK_DGRAM SOCK_DGRAM`

8.140.2.23 `#define OSCL SOCK_STREAM SOCK_STREAM`

8.140.2.24 `#define OSCL SOCKOPT_IP_ADDMEMBERSHIP IP_ADD_MEMBERSHIP`

8.140.2.25 `#define OSCL SOCKOPT_IP_MULTICAST_TTL IP_MULTICAST_TTL`

8.140.2.26 `#define OSCL SOCKOPT_IP_TOS IP_TOS`

8.140.2.27 `#define OSCL SOCKOPT_SOL_REUSEADDR SO_REUSEADDR`

```
accept_s=accept(s,NULL,NULL);\n    ok=(accept_s!=(-1));\n    if (!ok){err=errno;wouldblock=(err==EAGAIN||err==EWOULDBLOCK);}
```

8.140.2.33 #define OsciBind(s, addr, ok, err)

Value:

```
TOsciSockAddr* tmpadr = &addr;\n    sockaddr* saddr = OSCI_STATIC_CAST(sockaddr*, tmpadr);\n    ok=(bind(s,saddr,sizeof(addr))!=(-1));\n    if (!ok)err=errno
```

8.140.2.34 #define OsciCloseSocket(s, ok, err)

Value:

```
ok=(close(s)!=(-1));\n    if (!ok)err=errno
```

8.140.2.35 #define OsciConnect(s, addr, ok, err, wouldblock)

Value:

```
TOsciSockAddr* tmpadr = &addr;\n    sockaddr* saddr = OSCI_STATIC_CAST(sockaddr*, tmpadr);\n    ok=(connect(s,saddr,sizeof(addr))!=(-1));\n    if (!ok){err=errno;wouldblock=(err==EINPROGRESS);}
```

8.140.2.36 #define OsciConnectComplete(s, wset, eset, success, fail, ok, err)

Value:

```
success=fail=false;\n    if (FD_ISSET(s,&eset))\n        {fail=true;OsciGetAsyncSockErr(s,ok,err);}\n    else if (FD_ISSET(s,&wset))\n        {OsciGetAsyncSockErr(s,ok,err);if (ok && err==0)success=true;else fail=true;}
```

8.140.2.37 #define OsciGetAsyncSockErr(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;
```


8.140.2.38 #define OsciGetDottedAddr(hostent, dottedaddr, ok)

Value:

```
long *_hostaddr=(long*)hostent->h_addr_list[0];\
struct in_addr _inaddr;\
_inaddr.s_addr=*_hostaddr;\
dottedaddr=inet_ntoa(_inaddr);\
ok=(dottedaddr!=NULL);
```

8.140.2.39 #define OsciGetDottedAddrVector(hostent, dottedaddr, dottedaddrvect, ok)

Value:

```
if(dottedaddrvect)\
{\
long **_addrlist=(long**)hostent->h_addr_list;\
for(int i = 0; _addrlist[i] != NULL; i++){\
struct in_addr _inaddr;\
_inaddr.s_addr=*_addrlist[i];\
OsciNetworkAddress addr(inet_ntoa(_inaddr), 0);\
dottedaddrvect->push_back(addr);\
}\
if (!dottedaddrvect->empty())\
{dottedaddr->port = dottedaddrvect->front().port; dottedaddr->ipAddr.Set(dottedaddrvect->front().ipAddr);\
ok=(!dottedaddrvect->empty() && ((*dottedaddrvect)[0]).ipAddr.Str() != NULL);\
}\
else\
{\
char *add;\
OsciGetDottedAddr(hostent,add,ok);\
if(ok) dottedaddr->ipAddr.Set(add);\
}\
}
```

8.140.2.40 #define OsciGethostbyname(name, hostent, ok, err)

Value:

```
hostent=gethostbyname((const char*)name);\
ok=(hostent!=NULL);\
if (!ok)err=errno;
```

8.140.2.41 #define OsciGetPeerName(s, name, namelen, ok, err)

Value:

```
ok=(getpeername(s,(sockaddr*)&name,(socklen_t*)&namelen) != (-1) );\
if (!ok)err=errno
```

8.140.2.42 #define OsciJoin(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;\
}
```

8.140.2.43 #define OsciListen(s, size, ok, err)

Value:

```
ok=(listen(iSocket, qSize)!=(-1));\
    if (!ok)err=errno
```

8.140.2.44 #define OsciMakeInAddr(in_addr, addrstr, ok)

Value:

```
int32 result = inet_aton((const char*)addrstr, &in_addr);\
    ok=(result!=0);
```

8.140.2.45 #define OsciMakeSockAddr(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);
```

8.140.2.46 #define OsciPipe(x) pipe(x)

8.140.2.47 #define OsciReadFD(fd, buf, cnt) read(fd, buf, cnt)

8.140.2.48 #define OsciRecv(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);}
```

8.140.2.49 #define OsciRecvFrom(s, buf, len, paddr, paddrlen, ok, err, nbytes, wouldblock)

Value:

```
{\
void* p=paddr;\
nbytes=recvfrom(s,(void*)(buf),(size_t)(len),0,(struct sockaddr*)p,paddrlen);\
    ok=(nbytes!=(-1));\
    if (!ok){err=errno;wouldblock=(err==EAGAIN);}\
}
```

8.140.2.50 #define OsciSend(s, buf, len, ok, err, nbytes, wouldblock)

Value:

```
nbytes=send(s,(const void*)(buf),(size_t)(len),0);\
    ok=(nbytes!=(-1));\
    if (!ok){err=errno;wouldblock=(err==EAGAIN || err==EWOULDBLOCK);}
```

8.140.2.51 #define OsciSendTo(s, buf, len, addr, ok, err, nbytes, wouldblock)

Value:

```
TOsciSockAddr* tmpadr = &addr;\
    sockaddr* sadr = OSCI_STATIC_CAST(sockaddr*, tmpadr);\
    nbytes=sendto(s,(const void*)(buf),(size_t)(len),0,sadr,(socklen_t)sizeof(addr));\
    ok=(nbytes!=(-1));\
    if (!ok){err=errno;wouldblock=(err==EAGAIN || err==EWOULDBLOCK);}
```

8.140.2.52 #define OsciSetNonBlocking(s, ok, err)

Value:

```
ok=(fcntl(s,F_SETFL,O_NONBLOCK)!=(-1));\
    if (!ok)err=errno
```

8.140.2.53 #define OsciSetRecvBufferSize(s, val, ok, err)

Value:

```
ok=(setsockopt(s,SOL_SOCKET,SO_RCVBUF,(char*)&val, sizeof(int)) !=-1);\
    if (!ok)err=errno
```

8.140.2.54 #define OsciSetSockOpt(s, optLevel, optName, optVal, optLen, ok, err)

Value:

```
ok=(setsockopt(s,optLevel,optName,OSCI_STATIC_CAST(const char*,optVal),optLen) != (-1));\
    if (!ok)err=errno
```

8.140.2.55 #define OsciShutdown(s, how, ok, err)**Value:**

```
ok=(shutdown(iSocket,how)!=(-1));\  
if (!ok)err=errno
```

8.140.2.56 #define OsciSocket(s, fam, type, prot, ok, err)**Value:**

```
s=socket(fam,type,prot);\  
ok=(s!=(-1));\  
if (!ok)err=errno
```

8.140.2.57 #define OsciSocketCleanup(ok)**Value:**

```
signal(SIGPIPE,SIG_DFL);\  
ok=true
```

8.140.2.58 #define OsciSocketSelect(nfds, rd, wr, ex, timeout, ok, err, nhandles)**Value:**

```
nhandles=select(nfds,&rd,&wr,&ex,&timeout);\  
ok=(nhandles!=(-1));\  
if (!ok)err=errno
```

8.140.2.59 #define OsciSocketStartup(ok)**Value:**

```
signal(SIGPIPE,SIG_IGN);\  
ok=true
```

8.140.2.60 `#define OsciUnMakeInAddr(in_addr, addrstr) addrstr=inet_ntoa(in_addr);`

8.140.2.61 `#define OsciUnMakeSockAddr(sockaddr, addrstr) addrstr=inet_ntoa(sockaddr.sin_addr);`

8.140.2.62 `#define OsciValidInetAddr(addr) (inet_addr(addr)!=INADDR_NONE)`

8.140.2.63 `#define OsciWriteFD(fd, buf, cnt) write(fd,buf,cnt)`

8.140.3 Typedef Documentation

8.140.3.1 `typedef struct ip_mreq TIpMReq`

8.140.3.2 `typedef off_t TOsciFileOffset`

8.140.3.3 `typedef struct hostent TOsciHostent`

8.140.3.4 `typedef struct sockaddr_in TOsciSockAddr`

8.140.3.5 `typedef socklen_t TOsciSockAddrLen`

8.140.3.6 `typedef int TOsciSocket`

8.141 osclconfig_io_check.h File Reference

Typedefs

- typedef [TOscFileOffset](#) [__verify__TOscFileOffset__defined__](#)

8.141.1 Typedef Documentation

8.141.1.1 typedef [TOscFileOffset](#) [__verify__TOscFileOffset__defined__](#)

type `TOscFileOffset` should be defined as the type used for file size and offsets on the target platform.
Example: `typedef size_t TOscFileOffset;`

8.142 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

8.142.1 Detailed Description

This file contains configuration information for the ix86 processor family.

8.143 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` "./"

8.143.1 Detailed Description

This file contains configuration information for the ANSI build.

8.143.2 Define Documentation

8.143.2.1 #define `OSCL_HAS_RUNTIME_LIB_LOADING_SUPPORT` 1

8.143.2.2 #define `OSCL_LIB_READ_DEBUG_LIBS` 1

8.143.2.3 #define `PV_DYNAMIC_LOADING_CONFIG_FILE_PATH` "./"

8.143.2.4 #define `PV_RUNTIME_LIB_FILENAME_EXTENSION` "so"

8.144 osclconfig_lib_check.h File Reference

8.145 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

8.145.1 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.

8.145.2 Define Documentation

8.145.2.1 #define [OSCL_CHAR_IS_SIGNED](#) 0

8.145.2.2 #define [OSCL_CHAR_IS_UNSIGNED](#) 1

8.146 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

8.146.1 Define Documentation

8.146.1.1 #define OSCL_BYPASS_MEMMGT 1

8.146.1.2 #define OSCL_HAS_GLOBAL_NEW_DELETE 1

8.146.1.3 #define OSCL_HAS_HEAP_BASE_SUPPORT 1

8.146.1.4 #define OSCL_HAS_SYMBIAN_MEMORY_FUNCS 0

8.146.1.5 #define PVMEM_INST_LEVEL 1

8.147 osclconfig_memory_check.h File Reference

8.148 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_MSWIN_TIME_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

8.149 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"
```

8.149.1 Detailed Description

This file contains configuration information for the linux platform.

8.150 osclconfig_proc_check.h File Reference

Typedefs

- typedef [TOscIThreadId](#) `__verify__TOscIThreadId__defined__`
- typedef [TOscIThreadFuncRet](#) `__verify__TOscIThreadFuncRet__defined__`
- typedef [TOscIThreadFuncArg](#) `__verify__TOscIThreadFuncArg__defined__`
- typedef [TOscIThreadObject](#) `__verify__TOscIThreadObject__defined__`
- typedef [TOscIMutexObject](#) `__verify__TOscIMutexObject__defined__`
- typedef [TOscISemaphoreObject](#) `__verify__TOscISemaphoreObject__defined__`
- typedef [TOscIConditionObject](#) `__verify__TOscIConditionObject__defined__`

8.150.1 Typedef Documentation

8.150.1.1 typedef [TOscIConditionObject](#) `__verify__TOscIConditionObject__defined__`

type [TOscIConditionObject](#) should be defined as the type used as a condition variable on the target platform. Example: `typedef pthread_cond_t TOscIConditionObject;`

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 TOscIConditionObject; //not used`

8.150.1.2 typedef [TOscIMutexObject](#) `__verify__TOscIMutexObject__defined__`

type [TOscIMutexObject](#) should be defined as the type used as a mutex object or handle on the target platform. Example: `typedef pthread_mutex_t TOscIMutexObject;`

8.150.1.3 typedef [TOscISemaphoreObject](#) `__verify__TOscISemaphoreObject__defined__`

type [TOscISemaphoreObject](#) should be defined as the type used as a mutex object or handle on the target platform. Example: `typedef sem_t TOscISemaphoreObject;`

8.150.1.4 typedef [TOscIThreadFuncArg](#) `__verify__TOscIThreadFuncArg__defined__`

type [TOscIThreadFuncArg](#) should be defined as the type used as a thread function argument on the target platform. Example: `typedef LPVOID TOscIThreadFuncArg;`

8.150.1.5 typedef [TOscIThreadFuncRet](#) `__verify__TOscIThreadFuncRet__defined__`

type [TOscIThreadFuncRet](#) should be defined as the type used as a thread function return value on the target platform. Example: `typedef DWORD TOscIThreadFuncRet;`

8.150.1.6 typedef [TOscIThreadId](#) `__verify__TOscIThreadId__defined__`

type [TOscIThreadId](#) should be defined as the type used as a thread ID on the target platform. Example: `typedef DWORD TOscIThreadId;`

8.150.1.7 typedef `TOscIThreadObject` `__verify__TOscIThreadObject__defined__`

type `TOscIThreadObject` should be defined as the type used as a thread object or handle on the target platform. Example: `typedef pthread_t TOscIThreadObject;`

8.151 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 [TOscIThreadId](#)
- typedef void * [TOscIThreadFuncArg](#)
- typedef void * [TOscIThreadFuncRet](#)
- typedef pthread_t [TOscIThreadObject](#)
- typedef pthread_mutex_t [TOscIMutexObject](#)
- typedef int [TOscISemaphoreObject](#)
- typedef pthread_cond_t [TOscIConditionObject](#)

8.151.1 Define Documentation

8.151.1.1 `#define OSCL_HAS_NON_PREEMPTIVE_THREAD_SUPPORT 0`

8.151.1.2 `#define OSCL_HAS_PTHREAD_SUPPORT 1`

8.151.1.3 `#define OSCL_HAS_SEM_TIMEDWAIT_SUPPORT 0`

8.151.1.4 `#define OSCL_HAS_SYMBIAN_SCHEDULER 0`

8.151.1.5 `#define OSCL_HAS_THREAD_SUPPORT 1`

8.151.1.6 `#define OSCL_THREAD_DECL`

8.151.2 Typedef Documentation

8.151.2.1 `typedef pthread_cond_t TOsclConditionObject`

8.151.2.2 `typedef pthread_mutex_t TOsclMutexObject`

8.151.2.3 `typedef int TOsclSemaphoreObject`

8.151.2.4 `typedef void* TOsclThreadFuncArg`

8.151.2.5 `typedef void* TOsclThreadFuncRet`

8.151.2.6 `typedef pthread_t TOsclThreadId`

8.151.2.7 `typedef pthread_t TOsclThreadObject`

8.152 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 [TOscIThreadId](#)
- typedef void * [TOscIThreadFuncArg](#)
- typedef void * [TOscIThreadFuncRet](#)
- typedef pthread_t [TOscIThreadObject](#)
- typedef pthread_mutex_t [TOscIMutexObject](#)
- typedef sem_t [TOscISemaphoreObject](#)
- typedef pthread_cond_t [TOscIConditionObject](#)

8.152.1 Define Documentation

8.152.1.1 `#define OSCL_HAS_NON_PREEMPTIVE_THREAD_SUPPORT 0`

8.152.1.2 `#define OSCL_HAS_PTHREAD_SUPPORT 1`

8.152.1.3 `#define OSCL_HAS_SEM_TIMEDWAIT_SUPPORT 1`

8.152.1.4 `#define OSCL_HAS_SYMBIAN_SCHEDULER 0`

8.152.1.5 `#define OSCL_HAS_THREAD_SUPPORT 1`

8.152.1.6 `#define OSCL_THREAD_DECL`

8.152.2 Typedef Documentation

8.152.2.1 `typedef pthread_cond_t TOsclConditionObject`

8.152.2.2 `typedef pthread_mutex_t TOsclMutexObject`

8.152.2.3 `typedef sem_t TOsclSemaphoreObject`

8.152.2.4 `typedef void* TOsclThreadFuncArg`

8.152.2.5 `typedef void* TOsclThreadFuncRet`

8.152.2.6 `typedef pthread_t TOsclThreadId`

8.152.2.7 `typedef pthread_t TOsclThreadObject`

8.153 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 [OscBasicTimeStruct](#)
- typedef tm [OscBasicDateTimeStruct](#)

8.153.1 Define Documentation

8.153.1.1 #define [OSCL_HAS_UNIX_TIME_FUNCS](#) 1

8.153.2 Typedef Documentation

8.153.2.1 typedef tm [OscBasicDateTimeStruct](#)

8.153.2.2 typedef struct timeval [OscBasicTimeStruct](#)

8.154 osclconfig_time_check.h File Reference

Typedefs

- typedef [OscBasicTimeStruct __Validate__BasicTimeStruct__](#)
- typedef [OscBasicDateTimeStruct __Validate__BasicTimeDateStruct__](#)

8.154.1 Typedef Documentation

8.154.1.1 typedef [OscBasicDateTimeStruct __Validate__BasicTimeDateStruct__](#)

OscBasicDateTimeStruct type should be defined to the platform-specific date + time type.

8.154.1.2 typedef [OscBasicTimeStruct __Validate__BasicTimeStruct__](#)

OscBasicTimeStruct type should be defined to the platform-specific time of day type.

8.155 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_MSWIN_TIME_SUPPORT](#) 0
- #define [OSCL_HAS_MSWIN_PARTIAL_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 [TOscITlsKey](#)
- typedef pthread_mutex_t [TOscBasicLockObject](#)

8.155.1 Define Documentation

- 8.155.1.1 `#define _STRLIT(x) L ## x`
- 8.155.1.2 `#define _STRLIT_CHAR(x) x`
- 8.155.1.3 `#define _STRLIT_WCHAR(x) L ## x`
- 8.155.1.4 `#define INT64(x) x##LL`
- 8.155.1.5 `#define INT64_HILO(high, low) (((high##LL)<<32)|low)`
- 8.155.1.6 `#define OSCL_DISABLE_INLINES 0`
- 8.155.1.7 `#define OSCL_HAS_ANSI_MATH_SUPPORT 1`
- 8.155.1.8 `#define OSCL_HAS_ANSI_STDIO_SUPPORT 1`
- 8.155.1.9 `#define OSCL_HAS_ANSI_STDLIB_SUPPORT 1`
- 8.155.1.10 `#define OSCL_HAS_ANSI_STRING_SUPPORT 1`
- 8.155.1.11 `#define OSCL_HAS_ANSI_WIDE_STRING_SUPPORT 0`
- 8.155.1.12 `#define OSCL_HAS_BASIC_LOCK 1`
- 8.155.1.13 `#define OSCL_HAS_GLOBAL_VARIABLE_SUPPORT 1`
- 8.155.1.14 `#define OSCL_HAS_MSWIN_PARTIAL_SUPPORT 0`
- 8.155.1.15 `#define OSCL_HAS_MSWIN_SUPPORT 0`
- 8.155.1.16 `#define OSCL_HAS_MSWIN_TIME_SUPPORT 0`
- 8.155.1.17 `#define OSCL_HAS_NATIVE_INT64_TYPE 1`
- 8.155.1.18 `#define OSCL_HAS_NATIVE_UINT64_TYPE 1`
- 8.155.1.19 `#define OSCL_HAS_SYMBIAN_SUPPORT 0`
- 8.155.1.20 `#define OSCL_HAS_TLS_SUPPORT 1`
- 8.155.1.21 `#define OSCL_HAS_UNICODE_SUPPORT 1`
- 8.155.1.22 `#define OSCL_HAS_UNIX_SUPPORT 1`
- 8.155.1.23 `#define OSCL_MEMFRAG_PTR_BEFORE_LEN 1`
- 8.155.1.24 `#define OSCL_NATIVE_INT64_TYPE int64_t`
- 8.155.1.25 `#define OSCL_NATIVE_UINT64_TYPE uint64_t`
- 8.155.1.26 `#define OSCL_NATIVE_WCHAR_TYPE wchar_t`
- 8.155.1.27 `#define OSCL_TLS_GET_FUNC(key) pthread_getspecific(key)`

8.156 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_STUDIO_SUPPORT](#) 1
- #define [OSCL_MEMFRAG_PTR_BEFORE_LEN](#) 1
- #define [OSCL_HAS_UNIX_SUPPORT](#) 1
- #define [OSCL_HAS_MSWIN_SUPPORT](#) 0
- #define [OSCL_HAS_MSWIN_TIME_SUPPORT](#) 0
- #define [OSCL_HAS_MSWIN_PARTIAL_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 [TOscITlsKey](#)
- typedef pthread_mutex_t [TOscBasicLockObject](#)

8.156.1 Define Documentation

- 8.156.1.1 `#define _STRLIT(x) L ## x`
- 8.156.1.2 `#define _STRLIT_CHAR(x) x`
- 8.156.1.3 `#define _STRLIT_WCHAR(x) L ## x`
- 8.156.1.4 `#define INT64(x) x##LL`
- 8.156.1.5 `#define INT64_HILO(high, low) (((high##LL)<<32)|low)`
- 8.156.1.6 `#define OSCL_DISABLE_INLINES 0`
- 8.156.1.7 `#define OSCL_HAS_ANSI_MATH_SUPPORT 1`
- 8.156.1.8 `#define OSCL_HAS_ANSI_STDIO_SUPPORT 1`
- 8.156.1.9 `#define OSCL_HAS_ANSI_STDLIB_SUPPORT 1`
- 8.156.1.10 `#define OSCL_HAS_ANSI_STRING_SUPPORT 1`
- 8.156.1.11 `#define OSCL_HAS_ANSI_WIDE_STRING_SUPPORT 1`
- 8.156.1.12 `#define OSCL_HAS_BASIC_LOCK 1`
- 8.156.1.13 `#define OSCL_HAS_GLOBAL_VARIABLE_SUPPORT 1`
- 8.156.1.14 `#define OSCL_HAS_MSWIN_PARTIAL_SUPPORT 0`
- 8.156.1.15 `#define OSCL_HAS_MSWIN_SUPPORT 0`
- 8.156.1.16 `#define OSCL_HAS_MSWIN_TIME_SUPPORT 0`
- 8.156.1.17 `#define OSCL_HAS_NATIVE_INT64_TYPE 1`
- 8.156.1.18 `#define OSCL_HAS_NATIVE_UINT64_TYPE 1`
- 8.156.1.19 `#define OSCL_HAS_SYMBIAN_SUPPORT 0`
- 8.156.1.20 `#define OSCL_HAS_TLS_SUPPORT 1`
- 8.156.1.21 `#define OSCL_HAS_UNICODE_SUPPORT 1`
- 8.156.1.22 `#define OSCL_HAS_UNIX_SUPPORT 1`
- 8.156.1.23 `#define OSCL_MEMFRAG_PTR_BEFORE_LEN 1`
- 8.156.1.24 `#define OSCL_NATIVE_INT64_TYPE int64_t`
- 8.156.1.25 `#define OSCL_NATIVE_UINT64_TYPE uint64_t`
- 8.156.1.26 `#define OSCL_NATIVE_WCHAR_TYPE wchar_t`
- 8.156.1.27 `#define OSCL_TLS_GET_FUNC(key) pthread_getspecific(key)`

8.157 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)

8.157.1 Define Documentation

8.157.1.1 #define OSCL_CLOCK_HAS_DRIFT_CORRECTION 0

8.157.1.2 #define OSCL_HAS_SYMBIAN_MATH 0

8.157.1.3 #define OSCL_HAS_SYMBIAN_TIMERS 0

8.157.1.4 #define OSCL_RAND_MAX RAND_MAX

8.157.1.5 #define SLEEP_ONE_SEC sleep(1)

8.158 osclconfig_util_check.h File Reference

8.159 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](#)

8.159.1 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.

8.159.2 Define Documentation

8.159.2.1 #define _PVLOGGER_LOGBIN(LOGGER, LEVEL, MESSAGE)

Value:

```
{\
  if (LOGGER)\
  {\
    if (LOGGER->IsActive(LEVEL))\
    {\
      LOGGER->LogMsgBuffers MESSAGE;\
    }\
  }\
}
```

8.159.2.2 #define _PVLOGGER_LOGBIN_V(LOGGER, LEVEL, MESSAGE)

Value:

```
{\
  if (LOGGER)\
  {\
    if (LOGGER->IsActive(LEVEL))\
    {\
      LOGGER->LogMsgBuffersV MESSAGE;\
    }\
  }\
}
```

8.159.2.3 #define _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)

Value:

```
{\
  if (LOGGER)\
  {\
    if (LOGGER->IsActive(LEVEL))\
    {\
      LOGGER->LogMsgString MESSAGE;\
    }\
  }\
}
```

8.159.2.4 #define _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)

Value:

```
{\
  if (LOGGER)\
  {\
    if (LOGGER->IsActive(LEVEL))\
    {\
      LOGGER->LogMsgStringV MESSAGE;\
    }\
  }\
}
```

8.159.2.5 #define PVLOGGER_ENABLE 1

In case logging is compiled out, there is no need to compile the logger runtime code either.

8.159.2.6 #define PVLOGGER_INST_LEVEL 5**8.159.2.7 #define PVLOGGER_INST_LEVEL_SUPPORT 1****8.159.2.8 #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

**8.159.2.9 #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.

- 8.159.2.10 `#define PVLOGGER_LOGBIN_PVLOGMSG_INST_HLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN(LOGGER, LEVEL, MESSAGE)`
- 8.159.2.11 `#define PVLOGGER_LOGBIN_PVLOGMSG_INST_LLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN(LOGGER, LEVEL, MESSAGE)`
- 8.159.2.12 `#define PVLOGGER_LOGBIN_PVLOGMSG_INST_MLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN(LOGGER, LEVEL, MESSAGE)`
- 8.159.2.13 `#define PVLOGGER_LOGBIN_PVLOGMSG_INST_PROF(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN(LOGGER, LEVEL, MESSAGE)`
- 8.159.2.14 `#define PVLOGGER_LOGBIN_PVLOGMSG_INST_REL(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN(LOGGER, LEVEL, MESSAGE)`
- 8.159.2.15 `#define PVLOGGER_LOGBIN_V(IL, LOGGER, LEVEL, MESSAGE) PVLOGGER_LOGBIN_V_ ## IL (LOGGER, LEVEL, MESSAGE)`
- 8.159.2.16 `#define PVLOGGER_LOGBIN_V_PVLOGMSG_INST_HLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN_V(LOGGER, LEVEL, MESSAGE)`
- 8.159.2.17 `#define PVLOGGER_LOGBIN_V_PVLOGMSG_INST_LLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN_V(LOGGER, LEVEL, MESSAGE)`
- 8.159.2.18 `#define PVLOGGER_LOGBIN_V_PVLOGMSG_INST_PROF(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN_V(LOGGER, LEVEL, MESSAGE)`
- 8.159.2.19 `#define PVLOGGER_LOGBIN_V_PVLOGMSG_INST_REL(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN_V(LOGGER, LEVEL, MESSAGE)`
- 8.159.2.20 `#define PVLOGGER_LOGBIN_V_PVLOGMSG_V_INST_MLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN_V(LOGGER, LEVEL, MESSAGE)`
- 8.159.2.21 `#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 Message to Node 1`

`"));` -This message of log level `PVLOGMSG_WARNING`, and has a message id of 13

- 8.159.2.22** `#define PVLOGGER_LOGMSG_PVLOGMSG_INST_HLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)`
- 8.159.2.23** `#define PVLOGGER_LOGMSG_PVLOGMSG_INST_LLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)`
- 8.159.2.24** `#define PVLOGGER_LOGMSG_PVLOGMSG_INST_MLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)`
- 8.159.2.25** `#define PVLOGGER_LOGMSG_PVLOGMSG_INST_PROF(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)`
- 8.159.2.26** `#define PVLOGGER_LOGMSG_PVLOGMSG_INST_REL(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)`
- 8.159.2.27** `#define PVLOGGER_LOGMSG_V(IL, LOGGER, LEVEL, MESSAGE) PVLOGGER_LOGMSG_V_ ## IL (LOGGER, LEVEL, MESSAGE)`
- 8.159.2.28** `#define PVLOGGER_LOGMSG_V_PVLOGMSG_INST_HLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)`
- 8.159.2.29** `#define PVLOGGER_LOGMSG_V_PVLOGMSG_INST_LLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)`
- 8.159.2.30** `#define PVLOGGER_LOGMSG_V_PVLOGMSG_INST_MLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)`
- 8.159.2.31** `#define PVLOGGER_LOGMSG_V_PVLOGMSG_INST_PROF(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)`
- 8.159.2.32** `#define PVLOGGER_LOGMSG_V_PVLOGMSG_INST_REL(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)`
- 8.159.2.33** `#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.

8.159.2.34 `#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.

8.159.2.35 #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.

8.159.2.36 #define PVLOGMSG_INST_PROF 1

Profile Layer

The profile layer is used for messages and information related to measuring and reporting performance-related information.

8.159.2.37 #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.

8.159.3 Variable Documentation**8.159.3.1 const int32 PVLOGGER_LEVEL_UNINITIALIZED = -1****8.159.3.2 const PVLogger::log_level_type PVLOGMSG_ALERT = 1**

action must be taken immediately

8.159.3.3 const PVLogger::log_level_type PVLOGMSG_CRIT = 2

critical conditions

8.159.3.4 const PVLogger::log_level_type PVLOGMSG_DEBUG = 8

debug-level messages

8.159.3.5 const PVLogger::log_level_type PVLOGMSG_EMERG = 0

system is unusable

8.159.3.6 const PVLogger::log_level_type PVLOGMSG_ERR = 3

error conditions

8.159.3.7 const **PVLogger::log_level_type** **PVLOGMSG_FATAL_ERROR = PVLOGMSG_EMERG**

8.159.3.8 const **PVLogger::log_level_type** **PVLOGMSG_INFO = 6**

informational

8.159.3.9 const **PVLogger::log_level_type** **PVLOGMSG_NONFATAL_ERROR = PVLOGMSG_ERR**

8.159.3.10 const **PVLogger::log_level_type** **PVLOGMSG_NOTICE = 5**

normal but significant condition

8.159.3.11 const **PVLogger::log_level_type** **PVLOGMSG_STACK_TRACE = 7**

function enter and exit

8.159.3.12 const **PVLogger::log_level_type** **PVLOGMSG_STATISTIC = PVLOGMSG_INFO**

8.159.3.13 const **PVLogger::log_level_type** **PVLOGMSG_VERBOSE = PVLOGMSG_DEBUG**

8.159.3.14 const **PVLogger::log_level_type** **PVLOGMSG_WARNING = 4**

warning conditions

8.160 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

8.160.1 Variable Documentation

8.160.1.1 const [PVLoggerFilter::filter_status_type](#) [PVLOGGER_FILTER_ACCEPT](#) = 1

8.160.1.2 const [PVLoggerFilter::filter_status_type](#) [PVLOGGER_FILTER_NEUTRAL](#) = 3

8.160.1.3 const [PVLoggerFilter::filter_status_type](#) [PVLOGGER_FILTER_REJECT](#) = 2

8.161 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,...)

8.161.1 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.

8.161.2 Define Documentation

8.161.2.1 `#define PVLOGGER_C_INST_LEVEL 5`

8.161.2.2 `#define PVLOGMSG_C_ALERT 1`

8.161.2.3 `#define PVLOGMSG_C_CRIT 2`

8.161.2.4 `#define PVLOGMSG_C_EMERG 0`

8.161.2.5 `#define PVLOGMSG_C_ERR 3`

8.161.2.6 `#define PVLOGMSG_C_INFO 6`

8.161.2.7 `#define PVLOGMSG_C_INST_HLDBG 2`

8.161.2.8 `#define PVLOGMSG_C_INST_LLDBG 4`

8.161.2.9 `#define PVLOGMSG_C_INST_MLDBG 3`

8.161.2.10 `#define PVLOGMSG_C_INST_PROF 1`

8.161.2.11 `#define PVLOGMSG_C_INST_REL 0`

8.161.2.12 `#define PVLOGMSG_C_NOTICE 5`

8.161.2.13 `#define PVLOGMSG_C_STACK_DEBUG 8`

8.161.2.14 `#define PVLOGMSG_C_STACK_TRACE 7`

8.161.2.15 `#define PVLOGMSG_C_WARNING 4`

8.161.3 Function Documentation

8.161.3.1 `OSCL_IMPORT_REF void* pvLogger_GetLoggerObject (const char * tag)`

8.161.3.2 `OSCL_IMPORT_REF int pvLogger_IsActive (void * logger, int log_level)`

8.161.3.3 `OSCL_IMPORT_REF void pvLogger_LogMsgString (void * logger, int msgID, const char * fmt, ...)`

8.162 pvlogger_registry.h File Reference

```
#include "pvlogger.h"  
#include "oscl_tagtree.h"
```

Data Structures

- class [PVLoggerRegistry](#)

Chapter 9

oscl Page Documentation

9.1 Todo List

Global [MAX_NUMBER_OF_BYTE_PER_UTF8](#) Handle 4-byte surrogate pair representation

Index

- ~AllPassFilter
 - AllPassFilter, [114](#)
- ~BufFragGroup
 - BufFragGroup, [120](#)
- ~BufferMgr
 - BufferMgr, [117](#)
- ~CallbackTimer
 - CallbackTimer, [123](#)
- ~CallbackTimerObserver
 - CallbackTimerObserver, [125](#)
- ~DNSRequestParam
 - DNSRequestParam, [132](#)
- ~GetHostByNameParam
 - GetHostByNameParam, [135](#)
- ~HeapBase
 - HeapBase, [137](#)
- ~MM_AllocInfo
 - MM_AllocInfo, [149](#)
- ~MM_AllocNode
 - MM_AllocNode, [150](#)
- ~MM_Audit_Imp
 - MM_Audit_Imp, [153](#)
- ~MediaData
 - MediaData, [142](#)
- ~MemAllocator
 - MemAllocator, [145](#)
- ~OSCLMemAutoPtr
 - OSCLMemAutoPtr, [434](#)
- ~OSCL_FastString
 - OSCL_FastString, [175](#)
- ~OSCL_HeapString
 - osclutil, [83](#)
- ~OSCL_HeapStringA
 - OSCL_HeapStringA, [199](#)
- ~OSCL_StackString
 - osclutil, [83](#)
- ~OSCL_String
 - OSCL_String, [259](#)
- ~OSCL_wFastString
 - OSCL_wFastString, [293](#)
- ~OSCL_wHeapString
 - osclutil, [83](#)
- ~OSCL_wHeapStringA
 - OSCL_wHeapStringA, [298](#)
- ~OSCL_wStackString
 - osclutil, [83](#)
- ~OSCL_wString
 - OSCL_wString, [303](#)
- ~OscIAcceptMethod
 - OscIAcceptMethod, [306](#)
- ~OscIActiveObject
 - OscIActiveObject, [309](#)
- ~OscIAllocDestructDealloc
 - OscIAllocDestructDealloc, [312](#)
- ~OscIAsyncFile
 - OscIAsyncFile, [315](#)
- ~OscIAsyncFileBuffer
 - OscIAsyncFileBuffer, [318](#)
- ~OscIBinIStream
 - OscIBinIStream, [322](#)
- ~OscIBinOStream
 - OscIBinOStream, [329](#)
- ~OscIBindMethod
 - OscIBindMethod, [320](#)
- ~OscIComponentRegistry
 - OscIComponentRegistry, [342](#)
- ~OscIComponentRegistryElement
 - OscIComponentRegistryElement, [344](#)
- ~OscIConnectMethod
 - OscIConnectMethod, [346](#)
- ~OscIDNS
 - OscIDNS, [349](#)
- ~OscIDNSI
 - OscIDNSI, [351](#)
- ~OscIDNSIBase
 - OscIDNSIBase, [354](#)
- ~OscIDNSObserver
 - OscIDNSObserver, [359](#)
- ~OscIDNSRequest
 - OscIDNSRequest, [360](#)
- ~OscIExclusiveArrayPtr
 - OscIExclusiveArrayPtr, [379](#)
- ~OscIExclusivePtr
 - OscIExclusivePtr, [382](#)
- ~OscIExclusivePtrA
 - OscIExclusivePtrA, [385](#)
- ~OscIExecSchedulerCommonBase
 - OscIExecSchedulerCommonBase, [393](#)
- ~OscIFileCache
 - OscIFileCache, [400](#)

- ~OscIGetHostByNameMethod
 - OscIGetHostByNameMethod, [411](#)
- ~OscIIPSocketI
 - OscIIPSocketI, [417](#)
- ~OscIJump
 - OscIJump, [419](#)
- ~OscIListenMethod
 - OscIListenMethod, [420](#)
- ~OscILockBase
 - OscILockBase, [422](#)
- ~OscIMemAudit
 - OscIMemAudit, [427](#)
- ~OscIMemPoolAllocator
 - OscIMemPoolAllocator, [441](#)
- ~OscIMemPoolFixedChunkAllocator
 - OscIMemPoolFixedChunkAllocator, [443](#)
- ~OscIMemPoolFixedChunkAllocatorObserver
 - OscIMemPoolFixedChunkAllocator-Observer, [446](#)
- ~OscIMemPoolResizableAllocator
 - OscIMemPoolResizableAllocator, [448](#)
- ~OscIMemPoolResizableAllocatorMemoryObserver
 - OscIMemPoolResizableAllocatorMemory-Observer, [455](#)
- ~OscIMemPoolResizableAllocatorObserver
 - OscIMemPoolResizableAllocatorObserver, [456](#)
- ~OscIMemStatsNode
 - OscIMemStatsNode, [457](#)
- ~OscIMutex
 - OscIMutex, [458](#)
- ~OscINativeFile
 - OscINativeFile, [462](#)
- ~OscIOrNullLock
 - OscIOrNullLock, [466](#)
- ~OscIPriorityQueue
 - OscIPriorityQueue, [470](#)
- ~OscIPriorityQueueBase
 - OscIPriorityQueueBase, [473](#)
- ~OscIRecvFromMethod
 - OscIRecvFromMethod, [485](#)
- ~OscIRecvMethod
 - OscIRecvMethod, [489](#)
- ~OscIRefCounter
 - OscIRefCounter, [491](#)
- ~OscIRefCounterDA
 - OscIRefCounterDA, [493](#)
- ~OscIRefCounterMTDA
 - OscIRefCounterMTDA, [497](#)
- ~OscIRefCounterMTSA
 - OscIRefCounterMTSA, [499](#)
- ~OscIRefCounterMemFrag
 - OscIRefCounterMemFrag, [495](#)
- ~OscIRefCounterSA
 - OscIRefCounterSA, [501](#)
- ~OscIRegistryAccessClient
 - OscIRegistryAccessClient, [503](#)
- ~OscIRegistryClient
 - OscIRegistryClient, [508](#)
- ~OscIRegistryServTlsImpl
 - OscIRegistryServTlsImpl, [514](#)
- ~OscISchedulerObserver
 - OscISchedulerObserver, [516](#)
- ~OscIScopedLock
 - OscIScopedLock, [517](#)
- ~OscISemaphore
 - OscISemaphore, [520](#)
- ~OscISendMethod
 - OscISendMethod, [522](#)
- ~OscISendToMethod
 - OscISendToMethod, [524](#)
- ~OscISharedPtr
 - OscISharedPtr, [527](#)
- ~OscIShutdownMethod
 - OscIShutdownMethod, [529](#)
- ~OscISingleton
 - OscISingleton, [531](#)
- ~OscISocketI
 - OscISocketI, [535](#)
- ~OscISocketIBase
 - OscISocketIBase, [540](#)
- ~OscISocketMethod
 - OscISocketMethod, [545](#)
- ~OscISocketObserver
 - OscISocketObserver, [547](#)
- ~OscISocketRequestAO
 - OscISocketRequestAO, [550](#)
- ~OscISocketServ
 - OscISocketServ, [553](#)
- ~OscISocketServIBase
 - OscISocketServIBase, [558](#)
- ~OscITCPSocket
 - OscITCPSocket, [565](#)
- ~OscITCPSocketI
 - OscITCPSocketI, [572](#)
- ~OscITLS
 - OscITLS, [591](#)
- ~OscITLSEx
 - OscITLSEx, [593](#)
- ~OscIThread
 - OscIThread, [574](#)
- ~OscIThreadLock
 - OscIThreadLock, [578](#)
- ~OscITimer
 - OscITimer, [582](#)
- ~OscITimerObject
 - OscITimerObject, [586](#)
- ~OscITimerObserver

- OscTimerObserver, 589
- ~OscUDPSocket
 - OscUDPSocket, 601
- ~OscUDPSocketI
 - OscUDPSocketI, 607
- ~Osc_File
 - Osc_File, 180
- ~Osc_FileFind
 - Osc_FileFind, 189
- ~Osc_FileServer
 - Osc_FileServer, 192
- ~Osc_Linked_List
 - Osc_Linked_List, 205
- ~Osc_Linked_List_Base
 - Osc_Linked_List_Base, 211
- ~Osc_MTLinked_List
 - Osc_MTLinked_List, 224
- ~Osc_Queue
 - Osc_Queue, 235
- ~Osc_Queue_Base
 - Osc_Queue_Base, 237
- ~Osc_Rb_Tree
 - Osc_Rb_Tree, 242
- ~Osc_TAlloc
 - Osc_TAlloc, 280
- ~Osc_Tag
 - Osc_Tag, 263
- ~Osc_TagTree
 - Osc_TagTree, 268
- ~Osc_Vector
 - Osc_Vector, 284
- ~Osc_Vector_Base
 - Osc_Vector_Base, 289
- ~PVActiveBase
 - PVActiveBase, 612
- ~PVLogger
 - PVLogger, 617
- ~PVLoggerAppender
 - PVLoggerAppender, 622
- ~PVLoggerFilter
 - PVLoggerFilter, 624
- ~PVLoggerLayout
 - PVLoggerLayout, 625
- ~PVLoggerRegistry
 - PVLoggerRegistry, 627
- ~PVSchedulerStopper
 - PVSchedulerStopper, 630
- ~PVThreadContext
 - PVThreadContext, 633
- ~SendToParam
 - SendToParam, 639
- ~_OscBasicAllocator
 - _OscBasicAllocator, 108
- ~_OscHeapBase
 - _OscHeapBase, 110
- _FILE_OFFSET_BITS
 - osclconfig_io.h, 816
- _OSCL_Abort
 - osclbase, 35
- _OSCL_CLEANUP_BASE_CLASS
 - osclmemory, 49
- _OSCL_TRAP_NEW
 - osclmemory, 49
- _OscBasicAllocator, 107
- _OscBasicAllocator
 - ~_OscBasicAllocator, 108
 - allocate, 108
 - deallocate, 108
- _OscHeapBase, 109
- _OscHeapBase, 110
- _OscHeapBase
 - ~_OscHeapBase, 110
 - _OscHeapBase, 110
 - PVCleanupStack, 110
- _OscInteger64Transport
 - oscl_int64_utils.h, 704
- _Ownership
 - OSCLMemAutoPtr, 436
- _PVLOGGER_LOGBIN
 - pvlogger.h, 852
- _PVLOGGER_LOGBIN_V
 - pvlogger.h, 852
- _PVLOGGER_LOGMSG
 - pvlogger.h, 852
- _PVLOGGER_LOGMSG_V
 - pvlogger.h, 852
- _PV_TRAP
 - oscl_error_imp_fatalerror.h, 684
 - oscl_error_imp_jumps.h, 685
 - osclerror, 87
- _PV_TRAP_NO_TLS
 - oscl_error_imp_fatalerror.h, 684
 - oscl_error_imp_jumps.h, 685
 - osclerror, 87
- _Ptr
 - OscExclusiveArrayPtr, 380
 - OscExclusivePtr, 383
 - OscExclusivePtrA, 386
 - OscSingleton, 532
 - OscTLS, 592
 - OscTLSEx, 594
- _STRLIT
 - osclconfig_unix_android.h, 843
 - osclconfig_unix_common.h, 847
- _STRLIT_CHAR
 - osclconfig_unix_android.h, 843
 - osclconfig_unix_common.h, 847
- _STRLIT_WCHAR

- osclconfig_unix_android.h, 843
 - osclconfig_unix_common.h, 847
 - __TFS__
 - osclconfig.h, 804
 - __Validate__BasicTimeDateStruct__
 - osclconfig_time_check.h, 839
 - __Validate__BasicTimeStruct__
 - osclconfig_time_check.h, 839
 - __int16__check__
 - osclconfig, 24
 - __int32__check__
 - osclconfig, 24
 - __int8__check__
 - osclconfig, 24
 - __uint16__check__
 - osclconfig, 24
 - __uint32__check__
 - osclconfig, 24
 - __uint8__check__
 - osclconfig, 24
 - __verify__TOscIConditionObject__defined__
 - osclconfig_proc_check.h, 832
 - __verify__TOscIFileOffset__defined__
 - osclconfig_io_check.h, 823
 - __verify__TOscIMutexObject__defined__
 - osclconfig_proc_check.h, 832
 - __verify__TOscISemaphoreObject__defined__
 - osclconfig_proc_check.h, 832
 - __verify__TOscIThreadFuncArg__defined__
 - osclconfig_proc_check.h, 832
 - __verify__TOscIThreadFuncRet__defined__
 - osclconfig_proc_check.h, 832
 - __verify__TOscIThreadId__defined__
 - osclconfig_proc_check.h, 832
 - __verify__TOscIThreadObject__defined__
 - osclconfig_proc_check.h, 832
 - _fixedCaches
 - OscIFileCache, 400
 - _movableCache
 - OscIFileCache, 400
 - _oscl_audit_malloc
 - osclmemory, 58
 - _oscl_audit_free
 - osclmemory, 58
 - _oscl_audit_malloc
 - osclmemory, 58
 - _oscl_audit_new
 - osclmemory, 58
 - _oscl_audit_realloc
 - osclmemory, 59
 - _oscl_malloc
 - osclmemory, 59
 - _oscl_default_audit_malloc
 - osclmemory, 59
 - _oscl_default_audit_new
 - osclmemory, 59
 - _oscl_default_audit_realloc
 - osclmemory, 59
 - _oscl_free
 - osclmemory, 59
 - _oscl_malloc
 - osclmemory, 59
 - _oscl_realloc
 - osclmemory, 59
- a
- internalLeave, 138
- Abort
- OscIDNSMethod, 357
 - OscIDNSRequestAO, 362
 - OscISocketMethod, 545
 - OscISocketRequestAO, 550
- AbortAll
- OscIDNSMethod, 357
 - OscISocketMethod, 545
- Accept
- OscIAcceptMethod, 306
 - OscIAcceptRequest, 307
 - OscISocketI, 535
 - OscISocketIBase, 540
 - OscITCPSocket, 565
 - OscITCPSocketI, 572
- AcceptParam, 111
- AcceptParam, 111
- AcceptParam
- AcceptParam, 111
 - iBlankSocket, 111
- AcceptRequest
- OscIAcceptMethod, 306
- Activate
- OscIDNSRequest, 360
 - OscISocketRequest, 548
 - PVActiveBase, 612
- Add
- OscISocketServRequestList, 559
 - OscITimerQ, 590
- add_element
- Osc_Linked_List, 206
 - Osc_Linked_List_Base, 211
 - Osc_MTLlinked_List, 225
- add_ref
- CHeapRep, 129
- add_to_front
- Osc_Linked_List, 206
 - Osc_Linked_List_Base, 211
 - Osc_MTLlinked_List, 225

- addAllocNode
 - MM_Audit_Imp, 153
- AddAppender
 - PVLogger, 617
- AddFilter
 - PVLogger, 617
- AddFixedCache
 - OscI_File, 180
 - OscIFileCache, 400
- AddFragment
 - BufFragGroup, 120
- AddLocalFragment
 - MediaData, 142
- addnewmempoolbuffer
 - OscIMemPoolResizableAllocator, 448
- addRef
 - OscI_DefAllocWithRefCounter, 172
 - OscIMemPoolFixedChunkAllocator, 443
 - OscIMemPoolResizableAllocator, 448
 - OscIRefCounter, 491
 - OscIRefCounterDA, 494
 - OscIRefCounterMTDA, 498
 - OscIRefCounterMTSA, 500
 - OscIRefCounterSA, 502
- address
 - OscI_TAlloc, 280
- addressListCapacity
 - GetHostByNameParam, 134
- AddToExecTimerQ
 - OscIExecSchedulerCommonBase, 393
- AddToScheduler
 - OscIActiveObject, 309
 - OscITimerObject, 586
 - PVActiveBase, 612
- After
 - OscITimerObject, 586
- Alloc
 - OscIIPSocketI, 417
 - OscISocketMethod, 545
 - OscISocketRequestAO, 550
- ALLOC_AND_CONSTRUCT
 - osclbase, 32
- alloc_and_construct
 - OscI_TAlloc, 280
- alloc_and_construct_fl
 - OscI_TAlloc, 280
- ALLOC_NODE_FLAG
 - osclmemory, 61
- alloc_type
 - PVLogger, 617
 - PVLoggerRegistry, 627
- ALLOCATE
 - osclbase, 32
- allocate
 - _OscIBasicAllocator, 108
 - MemAllocator, 145
 - OscI_Alloc, 169
 - OscI_DefAlloc, 171
 - OscI_Opaque_Type_Alloc, 228
 - OscI_Opaque_Type_Alloc_LL, 229
 - OscI_TAlloc, 280
 - OscIErrorAllocator, 372
 - OscIMemAllocator, 424
 - OscIMemAllocDestructDealloc, 425
 - OSCLMemAutoPtr, 435
 - OscIMemBasicAllocator, 437
 - OscIMemBasicAllocDestructDealloc, 438
 - OscIMemPoolFixedChunkAllocator, 443
 - OscIMemPoolResizableAllocator, 449
 - OscIReadyAlloc, 481
- allocate_fl
 - OscI_Alloc, 169
 - OscI_DefAlloc, 171
 - OscI_TAlloc, 280
 - OscIMemAllocator, 424
 - OscIMemAllocDestructDealloc, 425
 - OscIReadyAlloc, 481
- allocateblock
 - OscIMemPoolResizableAllocator, 449
- allocator, 112
- allocNum
 - MM_AllocInfo, 149
 - MM_AllocQueryInfo, 151
- AllPassFilter, 113
 - AllPassFilter, 114
- AllPassFilter
 - ~AllPassFilter, 114
 - AllPassFilter, 114
 - filter_status_type, 113
 - FilterOpaqueMessge, 114
 - FilterString, 114
 - log_level_type, 113
 - message_id_type, 113
- ALREADY_SUSPENDED_ERROR
 - OscIProcStatus, 474
- Append
 - OscIPtr, 476
- append
 - CFastRep, 127
 - CHeapRep, 129
 - CStackRep, 131
- APPEND_MEDIA_AT_END
 - osclutil, 83
- append_rep
 - CHeapRep, 129
 - OSCL_String, 259
 - OSCL_wString, 303
- AppendBuffers

- PVLoggerAppender, [622](#)
- AppendNext
 - BufFragGroup, [120](#)
- AppendString
 - PVLoggerAppender, [622](#)
- assign
 - CHeapRep, [129](#)
- assign_vector
 - OscI_Vector_Base, [289](#)
- asyncreadcancel_test
 - OscI_File, [185](#)
- asyncreadwrite_test
 - OscI_File, [185](#)
- Attach
 - OscIBinStream, [335](#)
- audit_type
 - OscIMemGlobalAuditObject, [439](#)
- available_localbuf
 - MediaData, [143](#)
- back
 - OscI_Queue, [235](#)
 - OscI_Vector, [285](#)
- BAD_THREADID_ADDR_ERROR
 - OscIProcStatus, [474](#)
- base_link_type
 - OscI_Rb_Tree_Base, [244](#)
 - OscI_Rb_Tree_Const_Iterator, [246](#)
 - OscI_Rb_Tree_Iterator, [249](#)
 - OscI_Rb_Tree_Node_Base, [252](#)
- begin
 - OscI_Map, [218](#)
 - OscI_Rb_Tree, [242](#)
 - OscI_TagTree, [268](#)
 - OscI_Vector, [285](#)
- BeginScheduling
 - OscIExecSchedulerCommonBase, [393](#)
- BeginStats
 - OscIExecSchedulerCommonBase, [393](#)
- BFG_SUCCESS
 - BufFragStatusClass, [122](#)
- big_endian_to_host
 - osclbase, [35](#)
- Bind
 - osclbase, [35](#)
 - OscIBindMethod, [320](#)
 - OscIBindRequest, [321](#)
 - OscIIPSocketI, [417](#)
 - OscISocketI, [535](#)
 - OscISocketIBase, [540](#)
 - OscITCPSocket, [565](#)
 - OscIUDPSocket, [601](#)
- bind
 - BufferState, [118](#)
- BindAsync
 - OscISocketIBase, [540](#)
 - OscITCPSocket, [565](#)
 - OscITCPSocketI, [572](#)
 - OscIUDPSocket, [601](#)
 - OscIUDPSocketI, [607](#)
- BindParam, [115](#)
 - BindParam, [115](#)
- BindParam
 - BindParam, [115](#)
 - iAddr, [115](#)
- BindRequest
 - OscIBindMethod, [320](#)
- black
 - OscI_Rb_Tree_Node_Base, [252](#)
- BlockingLoopL
 - OscIExecSchedulerCommonBase, [393](#)
- bSetFailure
 - MM_AllocInfo, [149](#)
- Buffer
 - OscIAsyncFileBuffer, [318](#)
- buffer
 - CFastRep, [127](#)
 - CHeapRep, [129](#)
 - CStackRep, [131](#)
- buffer_states
 - BufFragGroup, [121](#)
- BufferFragment, [116](#)
- BufferFreeFuncPtr
 - osclutil, [68](#)
- BufferMgr, [117](#)
- BufferMgr
 - ~BufferMgr, [117](#)
 - BufferReleased, [117](#)
- BufferReleased
 - BufferMgr, [117](#)
- BufferState, [118](#)
 - BufferState, [118](#)
- BufferState
 - bind, [118](#)
 - BufferState, [118](#)
 - decrement_refcnt, [118](#)
 - get_buf_mgr, [118](#)
 - get_free_function, [118](#)
 - get_ptr, [118](#)
 - get_refcount, [118](#)
 - increment_refcnt, [118](#)
 - reset, [118](#)
- BufFragGroup, [119](#)
 - BufFragGroup, [120](#)
- BufFragGroup
 - ~BufFragGroup, [120](#)
 - AddFragment, [120](#)
 - AppendNext, [120](#)

- buffer_states, [121](#)
- BufFragGroup, [120](#)
- Clear, [120](#)
- fragments, [121](#)
- GetLength, [120](#)
- GetMaxFrag, [121](#)
- GetNext, [121](#)
- GetNumFrag, [121](#)
- length, [121](#)
- next, [121](#)
- num_fragments, [121](#)
- BufFragStatusClass, [122](#)
 - BFG_SUCCESS, [122](#)
 - EMPTY_FRAGMENT, [122](#)
 - FIXED_FRAG_LOC_FULL, [122](#)
 - INTERNAL_ERROR, [122](#)
 - INVALID_ID, [122](#)
 - NOT_ENOUGH_SPACE, [122](#)
 - NULL_INPUT, [122](#)
 - TOO_MANY_FRAGS, [122](#)
- BufFragStatusClass
 - status_t, [122](#)
- bufsize
 - OscI_Queue_Base, [239](#)
 - OscI_Vector_Base, [291](#)
- BYTES_IN_UUID_ARRAY
 - oscl_uuid.h, [801](#)
- c
 - OscIPriorityQueue, [472](#)
- c_bool
 - osclbase, [34](#)
- c_str
 - StrPtrLen, [647](#)
 - WStrPtrLen, [658](#)
- Callback
 - OscIReadyQ, [484](#)
- callback_timer_type
 - OscITimer, [582](#)
- CallbackTimer, [123](#)
 - CallbackTimer, [123](#)
- CallbackTimer
 - ~CallbackTimer, [123](#)
 - CallbackTimer, [123](#)
 - Run, [123](#)
- CallbackTimer< Alloc >
 - OscITimer, [583](#)
- CallbackTimerObserver, [125](#)
- CallbackTimerObserver
 - ~CallbackTimerObserver, [125](#)
 - TimerBaseElapsed, [125](#)
- CallRunExec
 - OscIExecSchedulerCommonBase, [393](#)
- Cancel
 - OscIActiveObject, [309](#)
 - OscITimer, [582](#)
 - OscITimerObject, [586](#)
 - PVActiveBase, [612](#)
- CancelAccept
 - OscISocketIBase, [541](#)
 - OscITCPSocket, [566](#)
 - OscITCPSocketI, [572](#)
- CancelBind
 - OscISocketIBase, [541](#)
 - OscITCPSocket, [566](#)
 - OscITCPSocketI, [572](#)
 - OscIUDPSocket, [601](#)
 - OscIUDPSocketI, [607](#)
- CancelConnect
 - OscISocketIBase, [541](#)
 - OscITCPSocket, [566](#)
 - OscITCPSocketI, [572](#)
- CancelFreeChunkAvailableCallback
 - OscIMemPoolFixedChunkAllocator, [443](#)
 - OscIMemPoolResizableAllocator, [449](#)
- CancelFreeMemoryAvailableCallback
 - OscIMemPoolResizableAllocator, [449](#)
- CancelFxn
 - OscIDNSIBase, [354](#)
 - OscISocketIBase, [541](#)
- CancelGetHostByName
 - OscIDNS, [349](#)
 - OscIDNSIBase, [354](#)
- Cancelled
 - OscIDNSRequestAO, [362](#)
- CancelListen
 - OscISocketIBase, [541](#)
 - OscITCPSocket, [566](#)
 - OscITCPSocketI, [572](#)
- CancelMethod
 - OscIDNSMethod, [357](#)
 - OscISocketMethod, [545](#)
- CancelRecv
 - OscISocketIBase, [541](#)
 - OscITCPSocket, [566](#)
 - OscITCPSocketI, [572](#)
- CancelRecvFrom
 - OscISocketIBase, [541](#)
 - OscIUDPSocket, [601](#)
 - OscIUDPSocketI, [607](#)
- CancelRequest
 - OscIDNSRequest, [360](#)
 - OscISocketRequest, [548](#)
- CancelSend
 - OscISocketIBase, [541](#)
 - OscITCPSocket, [566](#)
 - OscITCPSocketI, [572](#)
- CancelSendTo

- OscSocketIBase, [541](#)
- OscUDPSocket, [601](#)
- OscUDPSocketI, [607](#)
- CancelShutdown
 - OscSocketIBase, [541](#)
 - OscTCPSocket, [566](#)
 - OscTCPSocketI, [572](#)
- canPersistMoreHostAddresses
 - GetHostByNameParam, [135](#)
- capacity
 - Osc_Queue_Base, [238](#)
 - Osc_Vector_Base, [289](#)
 - OscFileCacheBuffer, [402](#)
- CFastRep, [126](#)
 - CFastRep, [127](#)
- CFastRep
 - append, [127](#)
 - buffer, [127](#)
 - CFastRep, [127](#)
 - maxsize, [127](#)
 - overwrite, [127](#)
 - set_r, [127](#)
 - set_w, [127](#)
 - size, [127](#)
 - writable, [127](#)
- chartype
 - OSCL_FastString, [175](#)
 - OSCL_HeapString, [196](#)
 - OSCL_HeapStringA, [198](#)
 - OSCL_StackString, [256](#)
 - OSCL_String, [259](#)
 - OSCL_wFastString, [292](#)
 - OSCL_wHeapString, [296](#)
 - OSCL_wHeapStringA, [298](#)
 - OSCL_wStackString, [301](#)
 - OSCL_wString, [303](#)
- CHeapRep, [128](#)
 - CHeapRep, [129](#)
- CHeapRep
 - add_ref, [129](#)
 - append, [129](#)
 - append_rep, [129](#)
 - assign, [129](#)
 - buffer, [129](#)
 - CHeapRep, [129](#)
 - maxsize, [129](#)
 - refcount, [129](#)
 - remove_ref, [129](#)
 - set, [129](#)
 - set_rep, [129](#)
 - size, [129](#)
- check_fence
 - MM_AllocBlockFence, [146](#)
- check_list
 - Osc_Linked_List, [206](#)
 - Osc_Linked_List_Base, [211](#)
- checkSum
 - StrCSumPtrLen, [644](#)
- ChecksumType
 - StrCSumPtrLen, [644](#)
- children
 - Osc_TagTree::Node, [278](#)
- children_type
 - Osc_TagTree, [268](#)
 - Osc_TagTree::Node, [278](#)
- ChooseCurCache
 - Osc_File::OscCacheObserver, [186](#)
- CleanInUse
 - OscAsyncFileBuffer, [318](#)
- Cleanup
 - OscErrorTrap, [374](#)
 - OscInit, [413](#)
 - OscMem, [423](#)
 - OscScheduler, [515](#)
 - PVLogger, [618](#)
- CleanupExecQ
 - OscExecSchedulerCommonBase, [393](#)
- CleanupParam
 - OscSocketRequestAO, [550](#)
- CleanupStatQ
 - OscExecSchedulerCommonBase, [393](#)
- Clear
 - BufFragGroup, [120](#)
 - MediaData, [142](#)
 - OscTimer, [582](#)
- clear
 - Osc_Linked_List, [206](#)
 - Osc_Map, [218](#)
 - Osc_Queue, [235](#)
 - Osc_Queue_Base, [238](#)
 - Osc_Rb_Tree, [242](#)
 - Osc_TagTree, [269](#)
 - Osc_Vector, [285](#)
- ClearTOS
 - OscSocketTOS, [563](#)
- Close
 - Osc_File, [180](#)
 - Osc_FileFind, [189](#)
 - Osc_FileServer, [192](#)
 - OscAsyncFile, [315](#)
 - OscDNSI, [351](#)
 - OscDNSIBase, [354](#)
 - OscFileCache, [400](#)
 - OscIPSocketI, [417](#)
 - OscMutex, [458](#)
 - OscNativeFile, [462](#)
 - OscRegistryAccessClient, [503](#)
 - OscRegistryClient, [508](#)

- OscRegistryClientImpl, 511
- OscRegistryServTlsImpl, 514
- OscSemaphore, 520
- OscSocketI, 535
- OscSocketIBase, 541
- OscSocketServ, 553
- OscSocketServI, 555
- OscSocketServIBase, 558
- OscSocketServRequestList, 559
- OscTCPSocket, 567
- OscTCPSocketI, 572
- OscUDPSocket, 602
- OscUDPSocketI, 607
- CloseSession
 - OscComponentRegistry, 342
- color
 - OscRb_Tree_Node_Base, 253
- color_type
 - OscRb_Tree_Node_Base, 252
- comp
 - Osc_Map::value_compare, 222
 - OscPriorityQueue, 472
- compare
 - OscCompareLess, 340
 - OscReadyCompare, 482
 - OscTimerCompare, 584
- compare_data
 - Osc_Opaque_Type_Alloc_LL, 229
- compare_EQ
 - Osc_Opaque_Type_Compare, 231
 - OscPriorityQueue, 470
- compare_LT
 - Osc_Opaque_Type_Compare, 231
 - OscPriorityQueue, 470
- CompareId
 - OscThread, 574
- Complete
 - OscDNSRequest, 360
 - OscSocketRequest, 548
- COMPUTE_MEM_ALIGN_SIZE
 - osclmemory, 50
- Connect
 - Osc_FileServer, 192
 - OscConnectMethod, 346
 - OscConnectRequest, 347
 - OscRegistryAccessClient, 503
 - OscRegistryClient, 508
 - OscRegistryClientImpl, 511
 - OscRegistryServTlsImpl, 514
 - OscSocketI, 535
 - OscSocketIBase, 541
 - OscSocketServ, 553
 - OscSocketServI, 555
 - OscSocketServIBase, 558
 - OscTCPSocket, 567
 - OscTCPSocketI, 572
- ConnectParam, 130
 - ConnectParam, 130
- ConnectParam
 - ConnectParam, 130
 - iAddr, 130
- ConnectRequest
 - OscConnectMethod, 346
- const_iterator
 - Osc_Map, 217
 - OscRb_Tree, 242
 - OscRb_Tree_Const_Iterator, 246
 - Osc_TagTree::const_iterator, 272
 - Osc_Vector, 284
- const_pointer
 - OscRb_Tree, 242
 - Osc_TAlloc, 280
- const_reference
 - Osc_Map, 217
 - Osc_Queue, 235
 - OscRb_Tree, 242
 - Osc_TAlloc, 280
 - Osc_Vector, 284
 - OscPriorityQueue, 470
- Construct
 - OscReadyQ, 484
 - OscTimerQ, 590
- construct
 - Osc_Linked_List_Base, 211
 - Osc_Opaque_Type_Alloc, 228
 - Osc_Opaque_Type_Alloc_LL, 229
 - Osc_Queue_Base, 238
 - Osc_TAlloc, 280
 - Osc_Vector_Base, 289
 - OscPriorityQueueBase, 473
- ConstructL
 - OscDNSMethod, 357
 - OscDNSRequestAO, 362
 - OscExecSchedulerCommonBase, 393
 - OscIPSocketI, 417
 - OscSocketMethod, 545
 - OscSocketRequestAO, 550
- ConstructStatQ
 - OscExecSchedulerCommonBase, 393
- container_type
 - OscPriorityQueue, 470
- Contains
 - Osc_File::OscFixedCacheParam, 187
 - OscFileCacheBuffer, 402
- count
 - Osc_Map, 218
 - OscRb_Tree, 242
 - Osc_TagTree, 269

- CPVInterfaceProxy
 - OscLErrorTrapImp, [376](#)
- Create
 - GetHostByNameParam, [135](#)
 - OscLMutex, [458](#)
 - OscLSemaphore, [520](#)
 - OscLThread, [575](#)
- CreateMemPool
 - OscLMemPoolAllocator, [441](#)
- createmempool
 - OscLMemPoolFixedChunkAllocator, [443](#)
- CreatePVLogger
 - PVLoggerRegistry, [628](#)
- createStatsNode
 - MM_Audit_Imp, [153](#)
- CStackRep, [131](#)
 - CStackRep, [131](#)
- CStackRep
 - append, [131](#)
 - buffer, [131](#)
 - CStackRep, [131](#)
 - maxsize, [131](#)
 - set, [131](#)
 - size, [131](#)
- CTIME_BUFFER_SIZE
 - osclbase, [45](#)
- CtimeStrBuf
 - osclbase, [34](#)
- Current
 - OscLExecScheduler, [387](#)
- currentPos
 - OscLFileCacheBuffer, [402](#)
- data
 - LinkedListElement, [139](#)
- data1
 - OscLUuid, [610](#)
- data2
 - OscLUuid, [610](#)
- data3
 - OscLUuid, [610](#)
- data4
 - OscLUuid, [610](#)
- deallocate
 - _OscLBasicAllocator, [108](#)
 - MemAllocator, [145](#)
 - OscL_Dealloc, [170](#)
 - OscL_DefAlloc, [171](#)
 - OscL_Opaque_Type_Alloc, [228](#)
 - OscL_Opaque_Type_Alloc_LL, [229](#)
 - OscL_TAlloc, [280](#)
 - OscLErrorAllocator, [372](#)
 - OscLMemAllocator, [424](#)
 - OscLMemAllocDestructDealloc, [425](#)
 - OSCLMemAutoPtr, [435](#)
 - OscLMemBasicAllocator, [437](#)
 - OscLMemBasicAllocDestructDealloc, [438](#)
 - OscLMemPoolFixedChunkAllocator, [444](#)
 - OscLMemPoolResizableAllocator, [449](#)
 - OscLReadyAlloc, [481](#)
- deallocateblock
 - OscLMemPoolResizableAllocator, [449](#)
- decrement_refcnt
 - BufferState, [118](#)
- DEFAULT_MM_AUDIT_MODE
 - osclmemory, [51](#)
- DEFAULT_POSTFILL_PATTERN
 - osclmemory, [51](#)
- DEFAULT_PREFILL_PATTERN
 - osclmemory, [51](#)
- Delete
 - OscL_DefAllocWithRefCount, [172](#)
 - OscLAsyncFile, [315](#)
 - OscLBuf, [339](#)
- Depth
 - OscLReadyQ, [484](#)
- depth
 - OscL_TagTree::Node, [278](#)
- dequeue_element
 - OscL_Linked_List, [206](#)
 - OscL_MTLinked_List, [225](#)
- Des
 - OscLBuf, [339](#)
- DesC
 - OscLBuf, [339](#)
- Destroy
 - DNSRequestParam, [132](#)
 - GetHostByNameParam, [135](#)
 - PVActiveBase, [612](#)
- destroy
 - OscL_Linked_List_Base, [211](#)
 - OscL_Opaque_Type_Alloc, [228](#)
 - OscL_Opaque_Type_Alloc_LL, [229](#)
 - OscL_Queue_Base, [238](#)
 - OscL_TAlloc, [280](#)
 - OscL_Vector, [285](#)
 - OscL_Vector_Base, [289](#)
- destroyallmempoolbuffers
 - OscLMemPoolResizableAllocator, [449](#)
- DestroyMemPool
 - OscLMemPoolAllocator, [441](#)
- destroymempool
 - OscLMemPoolFixedChunkAllocator, [444](#)
- destruct_and_dealloc
 - OscL_TAlloc, [280](#)
 - OscLDestructDealloc, [348](#)
 - OscLMemAllocDestructDealloc, [425](#)
 - OscLMemBasicAllocDestructDealloc, [438](#)

- difference_type
 - OscI_Rb_Tree, 242
- DIR_TYPE
 - OscI_FileFind, 188
- DisableAppenderInheritance
 - PVLogger, 618
- DiscardAcceptedSocket
 - OscIAcceptMethod, 306
- DNSRequestParam, 132
 - DNSRequestParam, 132
 - OscIDNSI, 352
- DNSRequestParam
 - ~DNSRequestParam, 132
 - Destroy, 132
 - DNSRequestParam, 132
 - iDNSRequest, 133
 - iFxn, 133
 - InThread, 132
 - iRefCount, 133
 - RemoveRef, 133
- DoCancel
 - OscIActiveObject, 310
 - OscIDNSRequestAO, 362
 - OscISocketRequestAO, 550
 - OscITimerObject, 586
 - PVActiveBase, 612
- E_BUFFER_TOO_SMALL
 - OscI_FileFind, 189
- E_INVALID_ARG
 - OscI_FileFind, 188
- E_INVALID_STATE
 - OscI_FileFind, 188
- E_MEMORY_ERROR
 - OscI_FileFind, 189
- E_NO_MATCH
 - OscI_FileFind, 189
- E_NOT_IMPLEMENTED
 - OscI_FileFind, 189
- E_OK
 - OscI_FileFind, 188
- E_OTHER
 - OscI_FileFind, 189
- E_PATH_NOT_FOUND
 - OscI_FileFind, 188
- E_PATH_TOO_LONG
 - OscI_FileFind, 188
- element_type
 - OscI_FileFind, 188
- elems
 - OscI_Queue_Base, 239
 - OscI_Vector_Base, 291
- empty
 - OscI_Map, 218
 - OscI_Queue_Base, 238
 - OscI_Rb_Tree, 242
 - OscI_TagTree, 269
 - OscI_Vector_Base, 289
 - OscIPriorityQueue, 471
- EMPTY_FRAGMENT
 - BufFragStatusClass, 122
- EMPTY_UUID
 - oscl_uuid.h, 801
- EnableKill
 - OscIThread, 575
- enablenullpointerreturn
 - OscIMemPoolFixedChunkAllocator, 444
 - OscIMemPoolResizableAllocator, 449
- End
 - OscIFileStats, 409
- end
 - OscI_Map, 218
 - OscI_Rb_Tree, 242
 - OscI_TagTree, 269
 - OscI_Vector, 285
- EndOfFile
 - OscI_File, 180
 - OscIAsyncFile, 315
 - OscIFileCache, 400
 - OscINativeFile, 462
- endPos
 - OscIFileCacheBuffer, 402
- EndScheduling
 - OscIExecSchedulerCommonBase, 393
- EndStats
 - OscIExecSchedulerCommonBase, 393
- EnterThreadContext
 - PVThreadContext, 633
- eof
 - OscIBinStream, 335
- EOF_STATE
 - OscIBinStream, 335
- EOSCL_StringOp_CompressASCII
 - osclutil, 69
- EOSCL_StringOp_UTF16ToUTF8
 - osclutil, 69
- EOSCL_wStringOp_ExpandASCII
 - osclutil, 69
- EOSCL_wStringOp_UTF8ToUTF16
 - osclutil, 69
- EOscIFileOp_Close
 - osclio, 97
- EOscIFileOp_EndOfFile
 - osclio, 97
- EOscIFileOp_Flush
 - osclio, 97
- EOscIFileOp_Last
 - osclio, 98

- EOsclFileOp_NativeClose
osclio, [97](#)
- EOsclFileOp_NativeEndOfFile
osclio, [98](#)
- EOsclFileOp_NativeFlush
osclio, [98](#)
- EOsclFileOp_NativeOpen
osclio, [97](#)
- EOsclFileOp_NativeRead
osclio, [97](#)
- EOsclFileOp_NativeSeek
osclio, [98](#)
- EOsclFileOp_NativeSetSize
osclio, [98](#)
- EOsclFileOp_NativeSize
osclio, [98](#)
- EOsclFileOp_NativeTell
osclio, [98](#)
- EOsclFileOp_NativeWrite
osclio, [98](#)
- EOsclFileOp_Open
osclio, [97](#)
- EOsclFileOp_Read
osclio, [97](#)
- EOsclFileOp_Seek
osclio, [97](#)
- EOsclFileOp_SetSize
osclio, [97](#)
- EOsclFileOp_Size
osclio, [97](#)
- EOsclFileOp_Tell
osclio, [97](#)
- EOsclFileOp_Write
osclio, [97](#)
- eOsclProcError
OsclProcStatus, [474](#)
- EOsclSocket_DataRecv
oscl_socket_stats.h, [772](#)
- EOsclSocket_DataSent
oscl_socket_stats.h, [772](#)
- EOsclSocket_Except
oscl_socket_stats.h, [771](#)
- EOsclSocket_OS
oscl_socket_stats.h, [771](#)
- EOsclSocket_Readable
oscl_socket_stats.h, [771](#)
- EOsclSocket_RequestAO_Canceled
oscl_socket_stats.h, [771](#)
- EOsclSocket_RequestAO_Error
oscl_socket_stats.h, [771](#)
- EOsclSocket_RequestAO_Success
oscl_socket_stats.h, [771](#)
- EOsclSocket_RequestAO_Timeout
oscl_socket_stats.h, [771](#)
- EOsclSocket_ServPoll
oscl_socket_stats.h, [771](#)
- EOsclSocket_ServRequestCancelIssued
oscl_socket_stats.h, [772](#)
- EOsclSocket_ServRequestComplete
oscl_socket_stats.h, [772](#)
- EOsclSocket_ServRequestIssued
oscl_socket_stats.h, [771](#)
- EOsclSocket_Writable
oscl_socket_stats.h, [771](#)
- EOsclSocketServ_LastEvent
oscl_socket_stats.h, [771](#)
- EOsclSocketServ_LoopsockError
oscl_socket_stats.h, [772](#)
- EOsclSocketServ_LoopsockOk
oscl_socket_stats.h, [772](#)
- EOsclSocketServ_SelectActivity
oscl_socket_stats.h, [771](#)
- EOsclSocketServ_SelectNoActivity
oscl_socket_stats.h, [771](#)
- EOsclSocketServ_SelectRescheduleAsap
oscl_socket_stats.h, [771](#)
- EOsclSocketServ_SelectReschedulePoll
oscl_socket_stats.h, [771](#)
- EOtherExecStats_Last
OsclExecSchedulerCommonBase, [392](#)
- EOtherExecStats_NativeOS
OsclExecSchedulerCommonBase, [392](#)
- EOtherExecStats_QueueTime
OsclExecSchedulerCommonBase, [392](#)
- EOtherExecStats_ReleaseTime
OsclExecSchedulerCommonBase, [392](#)
- EOtherExecStats_WaitTime
OsclExecSchedulerCommonBase, [392](#)
- EPriorityHigh
OsclActiveObject, [309](#)
- EPriorityHighest
OsclActiveObject, [309](#)
- EPriorityIdle
OsclActiveObject, [309](#)
- EPriorityLow
OsclActiveObject, [309](#)
- EPriorityNominal
OsclActiveObject, [309](#)
- EPVCritic_Ecp
OsclSocketTOS, [562](#)
- EPVDNSCancel
osclio, [98](#)
- EPVDNSFailure
osclio, [98](#)
- EPVDNSGetHostByName
osclio, [98](#)
- EPVDNSPending
osclio, [98](#)

- EPVDNSSuccess
 - osclio, [98](#)
- EPVDNSTimeout
 - osclio, [98](#)
- EPVFlash
 - OscSocketTOS, [562](#)
- EPVHiRel
 - OscSocketTOS, [562](#)
- EPVHiThrpt
 - OscSocketTOS, [562](#)
- EPVImmediate
 - OscSocketTOS, [562](#)
- EPVInetControl
 - OscSocketTOS, [562](#)
- EPVIPAddMembership
 - oscl_socket_types.h, [776](#)
- EPVIPMulticastTTL
 - oscl_socket_types.h, [776](#)
- EPVIPProtoIP
 - oscl_socket_types.h, [776](#)
- EPVIPProtoTCP
 - oscl_socket_types.h, [776](#)
- EPVIPTOS
 - oscl_socket_types.h, [776](#)
- EPVLDelay
 - OscSocketTOS, [562](#)
- EPVNetControl
 - OscSocketTOS, [562](#)
- EPVNoTOS
 - OscSocketTOS, [562](#)
- EPVOverrideFlash
 - OscSocketTOS, [562](#)
- EPVPriority
 - OscSocketTOS, [562](#)
- EPVRoutine
 - OscSocketTOS, [562](#)
- EPVSocket
 - oscl_socket_types.h, [776](#)
- EPVSocket_Last
 - oscl_socket_types.h, [776](#)
- EPVSocketAccept
 - oscl_socket_types.h, [776](#)
- EPVSocketBind
 - oscl_socket_types.h, [776](#)
- EPVSocketBothShutdown
 - oscl_socket_types.h, [776](#)
- EPVSocketCancel
 - oscl_socket_types.h, [775](#)
- EPVSocketConnect
 - oscl_socket_types.h, [776](#)
- EPVSocketFailure
 - oscl_socket_types.h, [775](#)
- EPVSocketListen
 - oscl_socket_types.h, [776](#)
- EPVSocketNotImplemented
 - oscl_socket_types.h, [776](#)
- EPVSocketPending
 - oscl_socket_types.h, [775](#)
- EPVSocketRecv
 - oscl_socket_types.h, [776](#)
- EPVSocketRecvFrom
 - oscl_socket_types.h, [776](#)
- EPVSocketRecvShutdown
 - oscl_socket_types.h, [776](#)
- EPVSocketSend
 - oscl_socket_types.h, [776](#)
- EPVSocketSendShutdown
 - oscl_socket_types.h, [776](#)
- EPVSocketSendTo
 - oscl_socket_types.h, [776](#)
- EPVSocketShutdown
 - oscl_socket_types.h, [776](#)
- EPVSocketSuccess
 - oscl_socket_types.h, [775](#)
- EPVSocketTimeout
 - oscl_socket_types.h, [775](#)
- EPVSockReuseAddr
 - oscl_socket_types.h, [776](#)
- EPVThreadContext_InThread
 - osclproc, [105](#)
- EPVThreadContext_NonOscThread
 - osclproc, [105](#)
- EPVThreadContext_OscThread
 - osclproc, [105](#)
- EPVThreadContext_Undetermined
 - osclproc, [105](#)
- equal_range
 - Osc_Map, [218](#)
 - Osc_Rb_Tree, [242](#)
- erase
 - Osc_Map, [219](#)
 - Osc_Rb_Tree, [242](#)
 - Osc_TagTree, [269](#)
 - Osc_Vector, [285](#)
 - Osc_Vector_Base, [289, 290](#)
- Error
 - OscExecSchedulerCommonBase, [393](#)
- error_type
 - Osc_FileFind, [188](#)
- ESocketServ_Connected
 - OscSocketServIBase, [557](#)
- ESocketServ_Error
 - OscSocketServIBase, [558](#)
- ESocketServ_Idle
 - OscSocketServIBase, [557](#)
- ESymbianAccessMode_Rfile
 - Osc_File, [179](#)
- ESymbianAccessMode_RfileBuf

- OscI_File, 179
- EXCEED_MAX_COUNT_VARIABLE_ERROR
 - OscIProcStatus, 475
- EXCEED_MAX_SEM_COUNT_ERROR
 - OscIProcStatus, 475
- Exit
 - OscIThread, 575
- ExitThreadContext
 - PVThreadContext, 633
- extract_string
 - osclutil, 69
- fail
 - OscIBinStream, 336
- FAIL_STATE
 - OscIBinStream, 335
- Failure
 - OscIDNSRequestAO, 362
- FENCE_PATTERN
 - osclmemory, 51
- FILE_TYPE
 - OscI_FileFind, 188
- fileName
 - MM_AllocQueryInfo, 151
- filePosition
 - OscIFileCacheBuffer, 402
- FileSize
 - OscIFileCache, 400
- fill_fence
 - MM_AllocBlockFence, 146
- FillFromFile
 - OscIFileCacheBuffer, 402
- filter_status_type
 - AllPassFilter, 113
 - PVLogger, 617
 - PVLoggerFilter, 623
- FilterOpaqueMessage
 - AllPassFilter, 114
 - PVLoggerFilter, 624
- FilterString
 - AllPassFilter, 114
 - PVLoggerFilter, 624
- Find
 - OscIComponentRegistryData, 343
- find
 - OscI_Map, 219
 - OscI_Rb_Tree, 242
 - OscI_TagTree, 269
- find_heap
 - OscIPriorityQueue, 471
 - OscIPriorityQueueBase, 473
- FindExact
 - OscIComponentRegistry, 342
- FindFirst
 - OscI_FileFind, 189
- findfreeblock
 - OscIMemPoolResizableAllocator, 450
- FindHierarchical
 - OscIComponentRegistry, 342
- FindNext
 - OscI_FileFind, 190
- FindPVBase
 - OscIExecSchedulerCommonBase, 393
- first
 - OscI_Pair, 233
- firstFragPtr
 - OscIBinStream, 337
- FIXED_FRAG_LOC_FULL
 - BufFragStatusClass, 122
- Flush
 - OscI_File, 181
 - OscIAsyncFile, 315
 - OscIFileCache, 400
 - OscINativeFile, 462
- FormatOpaqueMessage
 - PVLoggerLayout, 625
- FormatString
 - PVLoggerLayout, 625
- fragments
 - BufFragGroup, 121
- fragsLeft
 - OscIBinStream, 337
- freeblockavailable
 - OscIMemPoolResizableAllocatorObserver, 456
- freebytes
 - oscl_fsstat, 194
- freechunkavailable
 - OscIMemPoolFixedChunkAllocatorObserver, 446
- freememoryavailable
 - OscIMemPoolResizableAllocatorMemoryObserver, 455
- front
 - OscI_Queue, 236
 - OscI_Vector, 286
- Fxn
 - OscISocketRequest, 548
- get
 - OscIBinIStream, 322
 - OscIExclusiveArrayPtr, 379
 - OscIExclusivePtr, 382
 - OscIExclusivePtrA, 385
 - OSCLMemAutoPtr, 435
- get_buf_mgr
 - BufferState, 118

- get_count
 - OscISharedPtr, 527
- get_cstr
 - OSCL_FastString, 176
 - OSCL_HeapStringA, 199
 - OSCL_String, 259
 - OSCL_wFastString, 293
 - OSCL_wHeapStringA, 298
 - OSCL_wString, 303
 - osclutil, 69
- get_data
 - OscI_Opaque_Type_Alloc_LL, 230
- get_element
 - OscI_Linked_List, 206
 - OscI_Linked_List_Base, 211
 - OscI_MTLinked_List, 225
- get_first
 - OscI_Linked_List, 207
 - OscI_Linked_List_Base, 212
- get_free_function
 - BufferState, 118
- get_index
 - OscI_Linked_List, 207
 - OscI_Linked_List_Base, 212
 - OscI_MTLinked_List, 225
- get_int64_lower32
 - OscI_Int64_Utils, 203
- get_int64_middle32
 - OscI_Int64_Utils, 203
- get_int64_upper32
 - OscI_Int64_Utils, 203
- get_ISO8601_str_time
 - TimeValue, 651
- get_local_time
 - TimeValue, 651
- get_lower32
 - NTPTIME, 167
- get_maxsize
 - OSCL_FastString, 176
 - OSCL_HeapStringA, 199
 - OSCL_String, 259
 - OSCL_wFastString, 293
 - OSCL_wHeapStringA, 298
 - OSCL_wString, 303
 - osclutil, 70
- get_middle32
 - NTPTIME, 167
- get_next
 - OscI_Linked_List, 207
 - OscI_Linked_List_Base, 212
 - OscI_Opaque_Type_Alloc_LL, 230
- get_num_elements
 - OscI_Linked_List, 207
- get_ptr
 - BufferState, 118
- get_pv8601_str_time
 - TimeValue, 651
- get_refcount
 - BufferState, 118
- get_registry
 - TLSSStorageOps, 655
- get_rfc822_gmtime_str
 - TimeValue, 651
- get_sec
 - TimeValue, 652
- get_size
 - OSCL_FastString, 176
 - OSCL_HeapStringA, 200
 - OSCL_String, 260
 - OSCL_wFastString, 293
 - OSCL_wHeapStringA, 298
 - OSCL_wString, 303
 - osclutil, 70
- get_str
 - OSCL_FastString, 176
 - OSCL_HeapStringA, 200
 - OSCL_String, 260
 - OSCL_wFastString, 293
 - OSCL_wHeapStringA, 299
 - OSCL_wString, 303
 - osclutil, 71
- get_str_ctime
 - TimeValue, 652
- get_timeval_ptr
 - TimeValue, 652
- get_timevalue_in_usec
 - TimeValue, 652
- get_uint64_lower32
 - OscI_Int64_Utils, 203
- get_uint64_middle32
 - OscI_Int64_Utils, 203
- get_uint64_upper32
 - OscI_Int64_Utils, 203
- get_upper32
 - NTPTIME, 167
- get_usec
 - TimeValue, 652
- get_value
 - NTPTIME, 167
- GetAcceptedSocket
 - OscI_AcceptMethod, 306
- GetAcceptedSocketL
 - OscITCPSocket, 567
 - OscITCPSocketI, 572
- getAllocatedSize
 - OscI_MemPoolResizableAllocator, 450
- getAuditRoot
 - MM_Audit_Imp, 153

- GetAvailableBufferSize
 - MediaData, [142](#)
- getAvailableSize
 - OsciMemPoolResizableAllocator, [450](#)
- getBufferSize
 - OsciMemPoolResizableAllocator, [450](#)
- GetBufferState
 - osclutil, [71](#)
- getCapacity
 - OsciRefCountMemFrag, [496](#)
- getChecksum
 - StrCSumPtrLen, [644](#)
- getCount
 - Osci_DefAllocWithRefCount, [172](#)
 - OsciRefCount, [491](#)
 - OsciRefCountDA, [494](#)
 - OsciRefCountMemFrag, [496](#)
 - OsciRefCountMTDA, [498](#)
 - OsciRefCountMTSA, [500](#)
 - OsciRefCountSA, [502](#)
- GetElementType
 - Osci_FileFind, [190](#)
- GetError
 - Osci_File, [181](#)
 - OsciNativeFile, [462](#)
- GetErrorTrapImp
 - OsciErrorTrap, [374](#)
- GetFactories
 - OsciRegistryAccessClient, [503](#)
 - OsciRegistryClientImpl, [511](#)
 - OsciRegistryServTlsImpl, [514](#)
- GetFactory
 - OsciRegistryAccessClient, [503](#)
 - OsciRegistryClientImpl, [511](#)
 - OsciRegistryServTlsImpl, [514](#)
- GetFragment
 - osclutil, [71](#)
- getGlobalMemAuditObject
 - OsciMemGlobalAuditObject, [439](#)
- getHead
 - OsciDoubleListBase, [367](#)
- GetHostByName
 - OsciDNS, [350](#)
 - OsciDNSI, [351](#)
 - OsciDNSIBase, [354](#)
 - OsciGetHostByNameMethod, [411](#)
- GetHostByNameParam, [134](#)
 - addressListCapacity, [134](#)
 - OsciDNSRequestAO, [363](#)
- GetHostByNameParam
 - ~GetHostByNameParam, [135](#)
 - canPersistMoreHostAddresses, [135](#)
 - Create, [135](#)
 - Destroy, [135](#)
 - iAddr, [135](#)
 - iAddressList, [135](#)
 - iName, [135](#)
 - PersistHostAddress, [135](#)
- GetHostByNameResponseContainsAliasInfo
 - OsciDNSI, [352](#)
 - OsciDNSIBase, [354](#)
- GetHostByNameSuccess
 - OsciDNSI, [352](#)
 - OsciDNSIBase, [354](#)
- GetId
 - OsciExecSchedulerCommonBase, [393](#)
 - OsciThread, [575](#)
- getInstance
 - OsciSingletonRegistry, [533](#)
 - OsciTLSRegistry, [595](#)
 - OsciTLSRegistryEx, [596](#)
- getLargestContiguousFreeBlockSize
 - OsciMemPoolResizableAllocator, [450](#)
- GetLastError
 - Osci_FileFind, [190](#)
- getLeaveCode
 - OsciException, [377](#)
- GetLength
 - BufFragGroup, [120](#)
- GetLocalBufsize
 - MediaData, [143](#)
- GetLocalFragment
 - MediaData, [143](#)
- GetLock
 - OsciMemAudit, [428](#)
- GetLoggerObject
 - PVLogger, [618](#)
- GetLogLevel
 - PVLogger, [618](#)
- GetMaxFrag
 - BufFragGroup, [121](#)
- GetMediaFragment
 - MediaData, [143](#)
- GetMediaSize
 - MediaData, [143](#)
- getMemFrag
 - OsciRefCountMemFrag, [496](#)
- getMemFragPtr
 - OsciRefCountMemFrag, [496](#)
- getMemFragSize
 - OsciRefCountMemFrag, [496](#)
- getMemPoolBufferAllocatedSize
 - OsciMemPoolResizableAllocator, [450](#)
- getMemPoolBufferSize
 - OsciMemPoolResizableAllocator, [450](#)
- GetName
 - OsciExecSchedulerCommonBase, [393](#)
- GetNext

- BufFragGroup, [121](#)
- GetNextHost
 - OscIDNSI, [352](#)
 - OscIDNSIBase, [354](#)
- GetNextHostSuccess
 - OscIDNSI, [352](#)
 - OscIDNSIBase, [354](#)
- GetNumAppenders
 - PVLogger, [618](#)
- GetNumFrag
 - BufFragGroup, [121](#)
- GetNumMediaFrag
 - MediaData, [143](#)
- getOffset
 - OscIDoubleListBase, [367](#)
- GetParent
 - PVLogger, [619](#)
- GetPeerName
 - OscIPSocketI, [417](#)
 - OscISocketI, [535](#)
 - OscITCPSocket, [567](#)
 - OscIUDPSocket, [602](#)
- GetPriority
 - OscThread, [576](#)
- GetPVLoggerObject
 - PVLoggerRegistry, [628](#)
- GetPVLoggerRegistry
 - PVLoggerRegistry, [628](#)
- GetReadAsyncNumElements
 - OscNativeFile, [462](#)
- GetRecvData
 - OscIPSocketI, [417](#)
 - OscRecvFromMethod, [485](#)
 - OscRecvFromRequest, [487](#)
 - OscRecvMethod, [489](#)
 - OscRecvRequest, [490](#)
 - OscITCPSocket, [568](#)
 - OscITCPSocketI, [572](#)
 - OscIUDPSocket, [602](#)
 - OscIUDPSocketI, [607](#)
- GetRefCount
 - OscSharedPtr, [527](#)
- getRefCount
 - OscRefCountMemFrag, [496](#)
- GetRep
 - OscSharedPtr, [527](#)
- GetScheduler
 - OscExecSchedulerCommonBase, [393](#)
- GetSendData
 - OscIPSocketI, [417](#)
 - OscSendMethod, [522](#)
 - OscSendRequest, [523](#)
 - OscSendToMethod, [524](#)
 - OscSendToRequest, [525](#)
- OscITCPSocket, [568](#)
- OscITCPSocketI, [572](#)
- OscIUDPSocket, [602](#)
- OscIUDPSocketI, [607](#)
- GetShutdown
 - OscSocketIBase, [541](#)
- getSize
 - MM_Audit_Imp, [153](#)
- GetSocketError
 - OscDNSRequestAO, [362](#)
 - OscSocketRequestAO, [550](#)
- getTagActualSize
 - MM_Audit_Imp, [153](#)
- GetTimestamp
 - MediaData, [143](#)
- GetTOS
 - OscSocketTOS, [563](#)
- good
 - OscBinStream, [336](#)
- GOOD_STATE
 - OscBinStream, [335](#)
- Handle
 - Osc_File, [181](#)
 - OscFileHandle, [403](#)
- HandleDNSEvent
 - OscDNSObserver, [359](#)
- HandleSocketEvent
 - OscSocketObserver, [547](#)
- HasAsyncBind
 - OscSocketIBase, [541](#)
- HasAsyncListen
 - OscSocketIBase, [541](#)
- HasAsyncRead
 - OscNativeFile, [462](#)
- hash
 - OSCL_String, [260](#)
 - OSCL_wString, [303](#)
- HasThisOffset
 - OscAsyncFileBuffer, [318](#)
- HaveRoomInCurrentBlock
 - OscBinStream, [336](#)
- Head
 - OscIDoubleList, [365](#)
 - OscPriorityList, [468](#)
- head
 - Osc_Linked_List_Base, [214](#)
- HeapBase, [136](#)
- HeapBase, [137](#)
- HeapBase
 - ~HeapBase, [137](#)
 - HeapBase, [137](#)
- host_to_big_endian
 - osclbase, [35](#)

- host_to_little_endian
 - osclbase, [36](#)
- iActive
 - OscIDNSRequest, [360](#)
- iAddedNum
 - PVActiveBase, [614](#)
- iAddr
 - BindParam, [115](#)
 - ConnectParam, [130](#)
 - GetHostByNameParam, [135](#)
 - RecvFromParam, [635](#)
 - SendToParam, [639](#)
- iAddress
 - OscIPSocketI, [418](#)
- iAddressList
 - GetHostByNameParam, [135](#)
- iAlloc
 - OscIDNSIBase, [355](#)
 - OscIDNSMethod, [358](#)
 - OscExecSchedulerCommonBase, [397](#)
 - OscIPSocketI, [418](#)
 - OscSocketIBase, [543](#)
 - OscSocketServIBase, [558](#)
- iAllocatedSz
 - OscMemPoolResizableAllocator::MemPoolBufferInfo, [454](#)
- iAOPriority
 - TReadyQueueLink, [656](#)
- iAsyncReadBufferSize
 - OscNativeFileParams, [464](#)
- iBlankSocket
 - AcceptParam, [111](#)
- iBlockBuffer
 - OscMemPoolResizableAllocator::MemPoolBlockInfo, [453](#)
- iBlockInfoAlignedSize
 - OscMemPoolResizableAllocator, [452](#)
- iBlockingMode
 - OscExecSchedulerCommonBase, [397](#)
- iBlockPostFence
 - OscMemPoolResizableAllocator::MemPoolBlockInfo, [453](#)
- iBlockPreFence
 - OscMemPoolResizableAllocator::MemPoolBlockInfo, [453](#)
- iBlockSize
 - OscMemPoolResizableAllocator::MemPoolBlockInfo, [453](#)
- iBuffer
 - OscBuf, [339](#)
- iBufferInfoAlignedSize
 - OscMemPoolResizableAllocator, [452](#)
- iBufferPostFence
 - OscMemPoolResizableAllocator::MemPoolBufferInfo, [454](#)
- iBufferPreFence
 - OscMemPoolResizableAllocator::MemPoolBufferInfo, [454](#)
- iBufferSize
 - OscMemPoolResizableAllocator::MemPoolBufferInfo, [454](#)
- iBufRecv
 - RecvFromParam, [635](#)
 - RecvParam, [637](#)
- iBufSend
 - SendParam, [638](#)
 - SendToParam, [639](#)
- iBusy
 - PVActiveBase, [614](#)
- iCancel
 - OscSocketServRequestQElem, [561](#)
- iCBase
 - OscTrapStackItem, [599](#)
- iCheckFreeMemoryAvailable
 - OscMemPoolResizableAllocator, [452](#)
- iCheckNextAvailable
 - OscMemPoolResizableAllocator, [452](#)
- iCheckNextAvailableFreeChunk
 - OscMemPoolFixedChunkAllocator, [445](#)
- iChunkAlignment
 - OscMemPoolFixedChunkAllocator, [445](#)
- iChunkSize
 - OscMemPoolFixedChunkAllocator, [445](#)
- iChunkSizeMemAligned
 - OscMemPoolFixedChunkAllocator, [445](#)
- iComponentId
 - OscComponentRegistryElement, [344](#)
- iComponentIdCounter
 - OscComponentRegistry, [342](#)
- iContainer
 - OscFileCacheBuffer, [402](#)
 - OscSocketMethod, [546](#)
 - OscSocketRequestAO, [552](#)
- Id
 - OscAsyncFileBuffer, [318](#)
 - OscSocketRequestAO, [551](#)
 - PVThreadContext, [633](#)
- iData
 - OscComponentRegistry, [342](#)
- iDebugLogger
 - OscExecSchedulerCommonBase, [397](#)
- iDefAlloc
 - OscExecSchedulerCommonBase, [397](#)
- iDelta
 - OscExecSchedulerCommonBase, [397](#)
- iDNSFxn
 - OscIDNSMethod, [358](#)

- iDNSI
 - OscIDNSRequestAO, [363](#)
- iDNSMethod
 - OscIDNSRequestAO, [363](#)
- iDNSObserver
 - OscIDNSMethod, [358](#)
- iDNSRequest
 - DNSRequestParam, [133](#)
- iDNSRequestAO
 - OscIDNSMethod, [358](#)
 - OscIDNSRequest, [360](#)
- iDNSRequestParam
 - OscIDNSRequest, [360](#)
- iDoStop
 - OscIExecSchedulerCommonBase, [397](#)
- iDoSuspend
 - OscIExecSchedulerCommonBase, [397](#)
- iEnableNullPtrReturn
 - OscMemPoolFixedChunkAllocator, [445](#)
 - OscMemPoolResizableAllocator, [452](#)
- iEndAddr
 - OscMemPoolResizableAllocator::MemPoolBufferInfo, [454](#)
- iErrAlloc
 - OscISelect, [519](#)
- iErrorTrapImp
 - OscIExecSchedulerCommonBase, [397](#)
- iExecTimerQ
 - OscIExecSchedulerCommonBase, [397](#)
- iExpectedNumBlocksPerBuffer
 - OscMemPoolResizableAllocator, [452](#)
- iFactory
 - OscComponentRegistryElement, [344](#)
 - OscRegistryAccessElement, [507](#)
- iFilePosition
 - Osc_File::OscFixedCacheParam, [187](#)
- iFlags
 - RecvFromParam, [635](#)
 - RecvParam, [637](#)
 - SendParam, [638](#)
 - SendToParam, [639](#)
- iFreeMemChunkList
 - OscMemPoolFixedChunkAllocator, [445](#)
- iFreeMemContextData
 - OscMemPoolResizableAllocator, [452](#)
- iFreeMemPoolObserver
 - OscMemPoolResizableAllocator, [452](#)
- ifront
 - Osc_Queue_Base, [239](#)
- iFxn
 - DNSRequestParam, [133](#)
 - SocketRequestParam, [642](#)
- iGrandTotalTicks
 - OscIExecSchedulerCommonBase, [397](#)
- iHead
 - OscIDoubleListBase, [367](#)
 - OscIDoubleRunner, [368](#)
- iHeapCheck
 - OscISelect, [519](#)
- iHigh
 - OscInteger64Transport, [414](#)
- iHow
 - ShutdownParam, [640](#)
- iId
 - OscComponentRegistryElement, [344](#)
 - OscIDNSMethod, [358](#)
 - OscIPSocketI, [418](#)
- iIsIn
 - TReadyQueLink, [656](#)
- iJumpData
 - OscIErrorTrapImp, [376](#)
- iLeave
 - OscIErrorTrapImp, [376](#)
- iLen
 - PVsockBufRecv, [631](#)
 - PVsockBufSend, [632](#)
- iLength
 - OscIBuf, [339](#)
- iLogger
 - OscIDNSMethod, [358](#)
 - OscIDNSRequestAO, [363](#)
 - OscIExecSchedulerCommonBase, [397](#)
 - OscIPSocketI, [418](#)
 - OscISocketServIBase, [558](#)
- iLogPerfIndentStr
 - OscIExecSchedulerCommonBase, [397](#)
- iLogPerfIndentStrLen
 - OscIExecSchedulerCommonBase, [397](#)
- iLogPerfTotal
 - OscIExecSchedulerCommonBase, [397](#)
- iLow
 - OscInteger64Transport, [414](#)
- iMaxLen
 - PVsockBufRecv, [631](#)
- iMaxLength
 - OscIBuf, [339](#)
- iMaxNewMemPoolBufferSz
 - OscMemPoolResizableAllocator, [452](#)
- iMemPool
 - OscMemPoolFixedChunkAllocator, [445](#)
- iMemPoolAligned
 - OscMemPoolFixedChunkAllocator, [445](#)
- iMemPoolAllocator
 - OscMemPoolFixedChunkAllocator, [445](#)
- iMemPoolBufferAllocator
 - OscMemPoolResizableAllocator, [452](#)
- iMemPoolBufferList
 - OscMemPoolResizableAllocator, [452](#)

- iMemPoolBufferNumLimit
 - OscMemPoolResizableAllocator, [452](#)
- iMemPoolBufferSize
 - OscMemPoolResizableAllocator, [452](#)
- iMimeString
 - OscRegistryAccessElement, [507](#)
- iMultiMaxLen
 - RecvFromParam, [635](#)
- iMutex
 - OscComponentRegistry, [342](#)
- iName
 - GetHostByNameParam, [135](#)
 - OscExecSchedulerCommonBase, [397](#)
 - PVActiveBase, [614](#)
- iNativeAccessMode
 - OscNativeFileParams, [464](#)
- iNativeBufferSize
 - OscNativeFileParams, [464](#)
- iNativeMode
 - OscExecSchedulerCommonBase, [397](#)
- IncLogPerf
 - OscExecSchedulerCommonBase, [394](#)
- increment_refcnt
 - BufferState, [118](#)
- iNext
 - OscDoubleLink, [364](#)
 - OscDoubleRunner, [368](#)
 - OscTrapStackItem, [599](#)
- iNextAvailableContextData
 - OscMemPoolFixedChunkAllocator, [445](#)
 - OscMemPoolResizableAllocator, [452](#)
- iNextFreeBlock
 - OscMemPoolResizableAllocator::Mem-PoolBlockInfo, [453](#)
 - OscMemPoolResizableAllocator::Mem-PoolBufferInfo, [454](#)
- Init
 - OscErrorTrap, [374](#)
 - OscInit, [413](#)
 - OscMem, [423](#)
 - OscScheduler, [515](#)
 - PVLogger, [619](#)
- InitExecQ
 - OscExecSchedulerCommonBase, [394](#)
- Insert
 - OscDoubleListBase, [367](#)
 - OscPriorityList, [468](#)
- insert
 - Osc_Map, [219](#)
 - Osc_TagTree, [270](#)
 - Osc_Vector, [286](#)
 - Osc_Vector_Base, [290](#)
- insert_element
 - Osc_Linked_List, [207](#)
 - Osc_Linked_List_Base, [212](#)
- insert_unique
 - Osc_Rb_Tree, [242](#)
- InsertAfter
 - OscDoubleLink, [364](#)
- InsertBefore
 - OscDoubleLink, [364](#)
- InsertHead
 - OscDoubleList, [365](#)
 - OscDoubleListBase, [367](#)
- InsertTail
 - OscDoubleList, [365](#)
 - OscDoubleListBase, [367](#)
- InstallScheduler
 - OscExecSchedulerCommonBase, [394](#)
- INT64
 - osclconfig_unix_android.h, [843](#)
 - osclconfig_unix_common.h, [847](#)
- int64
 - osclbase, [34](#)
- INT64_HILO
 - osclconfig_unix_android.h, [843](#)
 - osclconfig_unix_common.h, [847](#)
- interfaceAddr
 - OscIpMReq, [415](#)
- INTERNAL_ERROR
 - BufFragStatusClass, [122](#)
- internalLeave, [138](#)
 - osclerror, [87](#)
- internalLeave
 - a, [138](#)
- InThread
 - DNSRequestParam, [132](#)
- iNumAOAdded
 - OscExecSchedulerCommonBase, [397](#)
- iNumChunk
 - OscMemPoolFixedChunkAllocator, [445](#)
- iNumOfRun
 - OscAsyncFile, [316](#)
- iNumOfRunErr
 - OscAsyncFile, [316](#)
- iNumOutstanding
 - OscMemPoolResizableAllocator::Mem-PoolBufferInfo, [454](#)
- iNumSessions
 - OscComponentRegistry, [342](#)
- INVALID_ACCESS_ERROR
 - OscProcStatus, [475](#)
- INVALID_ARGUMENT_ERROR
 - OscProcStatus, [475](#)
- INVALID_FUNCTION_ERROR
 - OscProcStatus, [475](#)
- INVALID_HANDLE_ERROR
 - OscProcStatus, [475](#)

INVALID_ID
 BufFragStatusClass, 122
 INVALID_OPERATION_ERROR
 OsciProcStatus, 475
 INVALID_PARAM_ERROR
 OsciProcStatus, 474
 INVALID_POINTER_ERROR
 OsciProcStatus, 475
 INVALID_PRIORITY_ERROR
 OsciProcStatus, 474
 INVALID_THREAD_ERROR
 OsciProcStatus, 474
 INVALID_THREAD_ID_ERROR
 OsciProcStatus, 474
 INVALID_TYPE
 Osci_FileFind, 188
 iObserver
 OsciIPSocketI, 418
 OsciMemPoolFixedChunkAllocator, 445
 OsciMemPoolResizableAllocator, 452
 iOffset
 OsciDoubleListBase, 367
 OsciDoubleRunner, 368
 iOpCount
 OsciFileStatsItem, 410
 iOsciBase
 OsciSelect, 519
 iOsciErrorTrap
 OsciSelect, 519
 iOsciLogger
 OsciSelect, 519
 iOsciMemory
 OsciSelect, 519
 iOsciScheduler
 OsciSelect, 519
 iOtherExecStats
 OsciExecSchedulerCommonBase, 397
 iOutputFile
 OsciSelect, 519
 iPacketLen
 RecvFromParam, 635
 iPacketSource
 RecvFromParam, 635
 ipAddr
 OsciNetworkAddress, 465
 iParam
 OsciFileStatsItem, 410
 OsciSocketRequest, 548
 OsciSocketRequestAO, 552
 iParam2
 OsciFileStatsItem, 410
 iParamSize
 OsciSocketRequestAO, 552
 iParentBuffer
 OsciMemPoolResizableAllocator::Mem-
 PoolBlockInfo, 453
 iPrev
 OsciDoubleLink, 364
 iPrevFreeBlock
 OsciMemPoolResizableAllocator::Mem-
 PoolBlockInfo, 453
 iPriority
 OsciPriorityLink, 467
 iPtr
 PVSockBufRecv, 631
 PVSockBufSend, 632
 iPVActiveStats
 PVActiveBase, 614
 iPVReadyQLink
 PVActiveBase, 614
 iPVStatQ
 OsciExecSchedulerCommonBase, 397
 iPVStats
 OsciExecSchedulerCommonBase, 397
 iQSize
 ListenParam, 140
 iReadyQ
 OsciExecSchedulerCommonBase, 397
 irear
 Osci_Queue_Base, 239
 iRefCount
 DNSRequestParam, 133
 OsciMemPoolFixedChunkAllocator, 445
 OsciMemPoolResizableAllocator, 452
 iRequestedAvailableFreeMemSize
 OsciMemPoolResizableAllocator, 452
 iRequestedNextAvailableSize
 OsciMemPoolResizableAllocator, 452
 iResumeSem
 OsciExecSchedulerCommonBase, 397
 is_writable
 OSCL_String, 260
 OSCL_wString, 304
 is_zero
 TimeValue, 653
 is_zulu
 TimeValue, 653
 IsActive
 PVLogger, 619
 IsAdded
 PVActiveBase, 612
 isAllocNodePtr
 MM_AllocBlockHdr, 147
 IsBusy
 OsciActiveObject, 310
 OsciTimerObject, 587
 iSchedulerAlloc
 OsciSelect, 519

- iSchedulerName
 - OscSelect, 519
- iSchedulerReserve
 - OscSelect, 519
- isCIEquivalentTo
 - StrCSumPtrLen, 644
 - StrPtrLen, 647
 - WStrPtrLen, 658
- isCIPrefixOf
 - StrPtrLen, 647
- iSelect
 - OscSocketServRequestQElem, 561
- IsEmpty
 - OscDoubleListBase, 367
- iSeqNum
 - TReadyQueLink, 656
- iServerError
 - OscSocketServIBase, 558
- iServState
 - OscSocketServIBase, 558
- isFixed
 - OscFileCacheBuffer, 402
- IsHead
 - OscDoubleList, 365
 - OscPriorityList, 468
- IsIn
 - OscReadyQ, 484
 - OscTimerQ, 590
- IsInAnyQ
 - PVActiveBase, 613
- IsInstalled
 - OscExecSchedulerCommonBase, 394
- IsInUse
 - OscAsyncFileBuffer, 318
- iSize
 - Osc_File::OscFixedCacheParam, 187
- isLetter
 - StrPtrLen, 647
- IsLocalData
 - MediaData, 143
- ISO8601TIME_BUFFER_SIZE
 - osclbase, 45
- ISO8601timeStrBuf
 - osclbase, 34
- ISO8601ToRFC822
 - osclbase, 36
- iSocket
 - OscIPSocketI, 418
- iSocketError
 - OscDNSRequestAO, 363
 - OscSocketRequestAO, 552
- iSocketFxn
 - OscSocketMethod, 546
- iSocketI
 - OscSocketRequest, 548
- iSocketRequest
 - OscSocketServRequestQElem, 561
- iSocketRequestAO
 - OscSocketMethod, 546
 - OscSocketRequest, 548
- iSocketServ
 - OscDNSIBase, 355
 - OscIPSocketI, 418
 - OscSocketIBase, 543
- IsOpen
 - OscSocketIBase, 541
- IsReady
 - OscDNSIBase, 354
- IsSameThreadContext
 - PVThreadContext, 633
- IsServConnected
 - OscSocketServIBase, 558
- IsServerThread
 - OscSocketServI, 556
- isSetFailure
 - MM_Audit_Imp, 154
- IsStarted
 - OscExecSchedulerCommonBase, 394
- IsTail
 - OscDoubleList, 365
 - OscPriorityList, 468
- iStartAddr
 - OscMemPoolResizableAllocator::MemPoolBufferInfo, 454
- iStartTick
 - OscFileStatsItem, 410
- iStatus
 - PVActiveBase, 614
- iStopper
 - OscExecSchedulerCommonBase, 397
- iStopperCrit
 - OscExecSchedulerCommonBase, 397
- IsUpdated
 - OscFileCacheBuffer, 402
- iSuspended
 - OscExecSchedulerCommonBase, 397
- IsValid
 - OscAsyncFileBuffer, 318
- iTAny
 - OscTrapStackItem, 599
- iterator
 - Osc_Linked_List_Base, 214
 - Osc_Map, 217
 - Osc_Rb_Tree, 242
 - Osc_Rb_Tree_Iterator, 249
 - Osc_TagTree::iterator, 275
 - Osc_Vector, 284
 - OscPriorityQueue, 470

- iThreadContext
 - OsciExecSchedulerCommonBase, [397](#)
 - PVActiveBase, [614](#)
- iTime
 - OsciExecSchedulerCommonBase, [397](#)
- iTimeCompareThreshold
 - OsciExecSchedulerCommonBase, [397](#)
- iTimeQueuedTicks
 - TReadyQueLink, [656](#)
- iTimeToRunTicks
 - TReadyQueLink, [656](#)
- iTotalPercent
 - OsciExecSchedulerCommonBase, [397](#)
- iTotalTicks
 - OsciFileStatsItem, [410](#)
- iTotalTicksTemp
 - OsciExecSchedulerCommonBase, [397](#)
- iTrapOperation
 - OsciTrapStackItem, [599](#)
- iTrapStack
 - OsciErrorTrapImp, [376](#)
- iVec
 - OsciComponentRegistryData, [343](#)
- iXferLen
 - SendParam, [638](#)
 - SendToParam, [639](#)
- Join
 - OsciIPSocketI, [417](#)
 - OsciSocketI, [536](#)
 - OsciSocketIBase, [541](#)
 - OsciUDPSocket, [602](#)
- JoinMulticastGroup
 - OsciUDPSocket, [603](#)
 - OsciUDPSocketI, [607](#)
- Jump
 - OsciJump, [419](#)
- key_comp
 - Osci_Map, [220](#)
- key_compare
 - Osci_Map, [217](#)
- key_type
 - Osci_Map, [217](#)
 - Osci_Rb_Tree, [242](#)
- largeasyncfilereadwrite_test
 - Osci_File, [185](#)
- Leave
 - OsciError, [370](#)
- LeaveIfError
 - OsciError, [370](#)
- LeaveIfNull
 - OsciError, [370](#)
- Left
 - OsciPtrC, [479](#)
- left
 - Osci_Rb_Tree_Node_Base, [253](#)
- len
 - OsciMemoryFragment, [440](#)
 - StrPtrLen, [647](#)
 - WStrPtrLen, [658](#)
- Length
 - OsciAsyncFileBuffer, [318](#)
 - OsciBuf, [339](#)
 - OsciPtr, [476](#)
 - OsciPtrC, [479](#)
- length
 - BufFragGroup, [121](#)
 - OsciBinStream, [337](#)
 - StrPtrLen, [647](#)
 - WStrPtrLen, [658](#)
- lineNo
 - MM_AllocInfo, [149](#)
 - MM_AllocQueryInfo, [151](#)
- link_type
 - Osci_Rb_Tree, [242](#)
 - Osci_Rb_Tree_Const_Iterator, [246](#)
 - Osci_Rb_Tree_Iterator, [249](#)
 - Osci_Rb_Tree_Node, [251](#)
- LinkedListElement, [139](#)
 - LinkedListElement, [139](#)
- LinkedListElement
 - data, [139](#)
 - LinkedListElement, [139](#)
 - next, [139](#)
- Listen
 - OsciListenMethod, [420](#)
 - OsciListenRequest, [421](#)
 - OsciSocketI, [536](#)
 - OsciSocketIBase, [541](#)
 - OsciTCPSocket, [568](#)
 - OsciTCPSocketI, [572](#)
- ListenAsync
 - OsciSocketIBase, [541](#)
 - OsciTCPSocket, [568](#)
 - OsciTCPSocketI, [573](#)
- ListenParam, [140](#)
 - ListenParam, [140](#)
- ListenParam
 - iQSize, [140](#)
 - ListenParam, [140](#)
- ListenRequest
 - OsciListenMethod, [420](#)
- little_endian_to_host
 - osclbase, [36](#)
- localbuf
 - MediaData, [143](#)

- Lock
 - OscILockBase, [422](#)
 - OscIMutex, [459](#)
 - OscIOrNullLock, [466](#)
 - OscIThreadLock, [578](#)
- lockAndGetInstance
 - OscISingletonRegistry, [533](#)
- Log
 - OscIFileStats, [409](#)
- log_level_type
 - AllPassFilter, [113](#)
 - PVLogger, [617](#)
 - PVLoggerFilter, [623](#)
 - PVLoggerRegistry, [627](#)
- LogAll
 - OscIFileStats, [409](#)
- Logger
 - OscISocketI, [536](#)
- LogMsgBuffers
 - PVLogger, [619](#)
- LogMsgBuffersV
 - PVLogger, [619](#)
- LogMsgString
 - PVLogger, [620](#)
- LogMsgStringV
 - PVLogger, [620](#)
- LoopbackSocket
 - OscISocketServI, [556](#)
- lower_bound
 - OscI_Map, [220](#)
 - OscI_Rb_Tree, [242](#)
- MakeAddr
 - OscISocketI, [536](#)
- MakeMulticastGroupInformation
 - OscISocketI, [536](#)
- makeValidTag
 - MM_Audit_Imp, [154](#)
- map_type
 - OscI_TagTree, [268](#)
- mapit
 - OscI_TagTree::const_iterator, [272](#)
 - OscI_TagTree::iterator, [275](#)
- mapiter
 - OscI_TagTree::const_iterator, [272](#)
 - OscI_TagTree::iterator, [275](#)
- Match
 - OscIComponentRegistryElement, [344](#)
- MAX_NUMBER_OF_BYTE_PER_UTF8
 - osclutil, [68](#)
- max_size
 - OscI_Map, [220](#)
 - OscI_Rb_Tree, [242](#)
- MAX_THRDS_REACHED_ERROR
 - OscIProcStatus, [474](#)
- maximum
 - OscI_Rb_Tree_Node_Base, [253](#)
- MaxLen
 - OscINameString, [460](#)
- maxsize
 - CFastRep, [127](#)
 - CHeapRep, [129](#)
 - CStackRep, [131](#)
- mbchar
 - osclbase, [34](#)
- MediaData, [141](#)
 - MediaData, [142](#)
- MediaData
 - ~MediaData, [142](#)
 - AddLocalFragment, [142](#)
 - available_localbuf, [143](#)
 - Clear, [142](#)
 - GetAvailableBufferSize, [142](#)
 - GetLocalBufsize, [143](#)
 - GetLocalFragment, [143](#)
 - GetMediaFragment, [143](#)
 - GetMediaSize, [143](#)
 - GetNumMediaFragments, [143](#)
 - GetTimestamp, [143](#)
 - IsLocalData, [143](#)
 - localbuf, [143](#)
 - MediaData, [142](#)
 - num_reserved_fragments, [143](#)
 - SetTimestamp, [143](#)
 - timestamp, [143](#)
- MediaStatusClass, [144](#)
- MediaTimestamp
 - osclutil, [68](#)
- MEM_ALIGN_SIZE
 - osclmemory, [51](#)
- MemAllocator, [145](#)
- MemAllocator
 - ~MemAllocator, [145](#)
 - allocate, [145](#)
 - deallocate, [145](#)
 - pointer, [145](#)
- memoryPoolBufferMgmtOverhead
 - OscIMemPoolResizableAllocator, [450](#)
- message_id_type
 - AllPassFilter, [113](#)
 - PVLogger, [617](#)
 - PVLoggerAppender, [622](#)
 - PVLoggerFilter, [623](#)
 - PVLoggerLayout, [625](#)
- MethodDone
 - OscIDNSMethod, [357](#)
 - OscISocketMethod, [545](#)
- MICROSECONDS

- osclbase, [35](#)
- MILLISECONDS
 - osclbase, [35](#)
- MIN_FENCE_SIZE
 - osclmemory, [51](#)
- minimum
 - OscI_Rb_Tree_Node_Base, [253](#)
- MM_AddTag
 - MM_Audit_Imp, [154](#)
 - OscIMemAudit, [428](#)
- MM_ALLOC_MAX_QUERY_FILENAME_LEN
 - osclmemory, [51](#)
- MM_ALLOC_MAX_QUERY_TAG_LEN
 - osclmemory, [51](#)
- MM_allocate
 - MM_Audit_Imp, [154](#)
 - OscIMemAudit, [428](#)
- MM_AllocBlockFence, [146](#)
 - MM_AllocBlockFence, [146](#)
- MM_AllocBlockFence
 - check_fence, [146](#)
 - fill_fence, [146](#)
 - MM_AllocBlockFence, [146](#)
 - pad, [146](#)
- MM_AllocBlockHdr, [147](#)
 - MM_AllocBlockHdr, [147](#)
- MM_AllocBlockHdr
 - isAllocNodePtr, [147](#)
 - MM_AllocBlockHdr, [147](#)
 - pad, [147](#)
 - pNode, [147](#)
 - pRootNode, [147](#)
 - setAllocNodeFlag, [147](#)
 - size, [147](#)
- MM_AllocInfo, [148](#)
 - MM_AllocInfo, [149](#)
- MM_AllocInfo
 - ~MM_AllocInfo, [149](#)
 - allocNum, [149](#)
 - bSetFailure, [149](#)
 - lineNo, [149](#)
 - MM_AllocInfo, [149](#)
 - operator delete, [149](#)
 - operator new, [149](#)
 - pFileName, [149](#)
 - pMemBlock, [149](#)
 - pStatsNode, [149](#)
 - size, [149](#)
- MM_AllocNode, [150](#)
 - MM_AllocNode, [150](#)
- MM_AllocNode
 - ~MM_AllocNode, [150](#)
 - MM_AllocNode, [150](#)
- operator delete, [150](#)
 - operator new, [150](#)
 - pAllocInfo, [150](#)
 - pNext, [150](#)
 - pPrev, [150](#)
- MM_AllocNodeAutoPtr
 - osclmemory, [58](#)
- MM_AllocQueryInfo, [151](#)
 - MM_AllocQueryInfo
 - allocNum, [151](#)
 - fileName, [151](#)
 - lineNo, [151](#)
 - pMemBlock, [151](#)
 - size, [151](#)
 - tag, [151](#)
- MM_AUDIT_ALLOC_NODE_ENABLE_FLAG
 - osclmemory, [51](#)
- MM_AUDIT_ALLOC_NODE_SUPPORT
 - osclmemory, [51](#)
- MM_AUDIT_FAILURE_SIMULATION_SUPPORT
 - osclmemory, [51](#)
- MM_AUDIT_FENCE_SUPPORT
 - osclmemory, [51](#)
- MM_AUDIT_FILL_SUPPORT
 - osclmemory, [51](#)
- MM_Audit_Imp, [152](#)
 - ~MM_Audit_Imp, [153](#)
 - addAllocNode, [153](#)
 - createStatsNode, [153](#)
 - getAuditRoot, [153](#)
 - getSize, [153](#)
 - getTagActualSize, [153](#)
 - isSetFailure, [154](#)
 - makeValidTag, [154](#)
 - MM_AddTag, [154](#)
 - MM_allocate, [154](#)
 - MM_Audit_Imp, [153](#)
 - MM_CreateAllocNodeInfo, [154](#)
 - MM_deallocate, [154](#)
 - MM_GetAllocNo, [154](#)
 - MM_GetAllocNodeInfo, [155](#)
 - MM_GetExistingTag, [155](#)
 - MM_GetMode, [155](#)
 - MM_GetNumAllocNodes, [155](#)
 - MM_GetOverheadStats, [155](#)
 - MM_GetPostfillPattern, [155](#)
 - MM_GetPrefillPattern, [155](#)
 - MM_GetRootNode, [156](#)
 - MM_GetStats, [156](#)
 - MM_GetStatsInDepth, [156](#)
 - MM_GetTagNode, [156](#)
 - MM_GetTreeNodees, [156](#)

- MM_ReleaseAllocNodeInfo, 156
- MM_SetFailurePoint, 157
- MM_SetMode, 157
- MM_SetPostfillPattern, 157
- MM_SetPrefillPattern, 157
- MM_SetTagLevel, 157
- MM_UnsetFailurePoint, 157
- MM_Validate, 157
- pruneSubtree, 158
- removeALLAllocNodes, 158
- removeAllocNode, 158
- retrieveParentTag, 158
- retrieveParentTagLength, 158
- updateStatsNode, 158
- updateStatsNodeInFailure, 158
- validate, 158
- validate_all_heap, 158
- MM_AUDIT_INCLUDE_ALL_HEAP_-
VALIDATION
 - osclmemory, 51
- MM_AUDIT_POSTFILL_FLAG
 - osclmemory, 51
- MM_AUDIT_PREFILL_FLAG
 - osclmemory, 51
- MM_AUDIT_SUPPRESS_FILENAME_FLAG
 - osclmemory, 51
- MM_AUDIT_VALIDATE_ALL_HEAP_FLAG
 - osclmemory, 51
- MM_AUDIT_VALIDATE_BLOCK
 - osclmemory, 51
- MM_AUDIT_VALIDATE_ON_FREE_FLAG
 - osclmemory, 51
- MM_AuditOverheadStats, 160
- MM_AuditOverheadStats
 - per_allocation_overhead, 160
 - stats_overhead, 160
- MM_CreateAllocNodeInfo
 - MM_Audit_Imp, 154
 - OscMemAudit, 428
- MM_deallocate
 - MM_Audit_Imp, 154
 - OscMemAudit, 428
- MM_FailInsertParam, 161
 - MM_FailInsertParam, 161
- MM_FailInsertParam
 - MM_FailInsertParam, 161
 - nAllocNum, 161
 - operator delete, 161
 - operator new, 161
 - reset, 161
 - xsubi, 161
- MM_GetAllocNo
 - MM_Audit_Imp, 154
 - OscMemAudit, 428
- MM_GetAllocNodeInfo
 - MM_Audit_Imp, 155
 - OscMemAudit, 428
- MM_GetExistingTag
 - MM_Audit_Imp, 155
 - OscMemAudit, 429
- MM_GetMode
 - MM_Audit_Imp, 155
 - OscMemAudit, 429
- MM_GetNumAllocNodes
 - MM_Audit_Imp, 155
 - OscMemAudit, 429
- MM_GetOverheadStats
 - MM_Audit_Imp, 155
 - OscMemAudit, 429
- MM_GetPostfillPattern
 - MM_Audit_Imp, 155
 - OscMemAudit, 429
- MM_GetPrefillPattern
 - MM_Audit_Imp, 155
 - OscMemAudit, 429
- MM_GetRefCount
 - OscMemAudit, 429
- MM_GetRootNode
 - MM_Audit_Imp, 156
 - OscMemAudit, 430
- MM_GetStats
 - MM_Audit_Imp, 156
 - OscMemAudit, 430
- MM_GetStatsInDepth
 - MM_Audit_Imp, 156
 - OscMemAudit, 430
- MM_GetTagNode
 - MM_Audit_Imp, 156
 - OscMemAudit, 430
- MM_GetTreeNodees
 - MM_Audit_Imp, 156
 - OscMemAudit, 430
- MM_ReleaseAllocNodeInfo
 - MM_Audit_Imp, 156
 - OscMemAudit, 430
- MM_SetFailurePoint
 - MM_Audit_Imp, 157
 - OscMemAudit, 430
- MM_SetMode
 - MM_Audit_Imp, 157
 - OscMemAudit, 431
- MM_SetPostfillPattern
 - MM_Audit_Imp, 157
 - OscMemAudit, 431
- MM_SetPrefillPattern
 - MM_Audit_Imp, 157
 - OscMemAudit, 431
- MM_SetTagLevel

- MM_Audit_Imp, 157
- OscMemAudit, 431
- MM_Stats_CB, 162
 - MM_Stats_CB, 162
 - num_child_nodes, 162
 - operator delete, 162
 - operator new, 162
 - pStats, 162
 - tag, 162
- MM_Stats_t, 163
 - MM_Stats_t, 164
 - numAllocFails, 164
 - numAllocs, 164
 - numBytes, 164
 - operator delete, 164
 - operator new, 164
 - peakNumAllocs, 164
 - peakNumBytes, 164
 - reset, 164
 - totalNumAllocs, 164
 - totalNumBytes, 164
 - update, 164
- MM_StatsNodeTagTreeType
 - osclmemory, 58
- MM_UnsetFailurePoint
 - MM_Audit_Imp, 157
 - OscMemAudit, 431
- MM_Validate
 - MM_Audit_Imp, 157
 - OscMemAudit, 431
- MMAuditCharAutoPtr
 - osclmemory, 58
- MMAuditUInt8AutoPtr
 - osclmemory, 58
- Mode
 - OscNativeFile, 462
- mode
 - oscl_stat_buf, 257
- MODE_APPEND
 - Osc_File, 179
- MODE_BINARY
 - Osc_File, 179
- MODE_READ
 - Osc_File, 179
- MODE_READ_PLUS
 - Osc_File, 179
- MODE_READWRITE
 - Osc_File, 179
- MODE_TEXT
 - Osc_File, 179
- mode_type
 - Osc_File, 179
- move_to_end
 - Osc_Linked_List, 208
- Osc_Linked_List_Base, 212
- Osc_MTLinked_List, 225
- move_to_front
 - Osc_Linked_List, 208
 - Osc_Linked_List_Base, 213
 - Osc_MTLinked_List, 226
- MSEC_PER_SEC
 - osclbase, 45
- MSEC_TO_MICROSEC
 - oscl_socket_method.h, 760
- MsecToTicks
 - OscTickCount, 579
- multicastAddr
 - OscIpMReq, 415
- MUTEX_LOCKED_ERROR
 - OscProcStatus, 475
- nAllocNum
 - MM_FailInsertParam, 161
- New
 - Osc_DefAllocWithRefCount, 173
- NewL
 - OscAcceptMethod, 306
 - OscAsyncFile, 315
 - OscAsyncFileBuffer, 318
 - OscBindMethod, 320
 - OscBuf, 339
 - OscConnectMethod, 346
 - OscDNS, 350
 - OscDNSI, 352
 - OscGetHostByNameMethod, 411
 - OscListenMethod, 420
 - OscRecvFromMethod, 485
 - OscRecvMethod, 489
 - OscSendMethod, 522
 - OscSendToMethod, 524
 - OscShutdownMethod, 529
 - OscSocketI, 536
 - OscSocketServ, 554
 - OscSocketServI, 556
 - OscTCPSocket, 568
 - OscTCPSocketI, 573
 - OscUDPSocket, 603
 - OscUDPSocketI, 607
- NewRequest
 - OscDNSRequestAO, 362
 - OscSocketRequestAO, 551
- next
 - BufFragGroup, 121
 - LinkedListElement, 139
- nextFragPtr
 - OscBinStream, 337
- NO_PERMISSION_ERROR
 - OscProcStatus, 474

- Node
 - OscI_TagTree::Node, 278
- node
 - OscI_Rb_Tree_Const_Iterator, 246
 - OscI_Rb_Tree_Iterator, 249
- node_ptr
 - OscI_TagTree, 268
- node_type
 - OscI_TagTree, 268
- NOT_ENOUGH_MEMORY_ERROR
 - OscIProcStatus, 474
- NOT_ENOUGH_RESOURCES_ERROR
 - OscIProcStatus, 474
- NOT_ENOUGH_SPACE
 - BufFragStatusClass, 122
- NOT_IMPLEMENTED
 - OscIProcStatus, 475
- NOT_SUSPENDED_ERROR
 - OscIProcStatus, 474
- notifyfreeblockavailable
 - OscIMemPoolResizableAllocator, 450
- notifyfreechunkavailable
 - OscIMemPoolFixedChunkAllocator, 444
- notifyfreememoryavailable
 - OscIMemPoolResizableAllocator, 450
- NTPTIME, 165
 - get_lower32, 167
 - get_middle32, 167
 - get_upper32, 167
 - get_value, 167
 - NTPTIME, 166, 167
 - operator+=, 167
 - operator-, 167
 - operator=, 167, 168
 - set_from_system_time, 168
 - set_to_current_time, 168
 - TimeValue, 654
 - to_system_time, 168
- NULL
 - osclbase, 32
- NULL_INPUT
 - BufFragStatusClass, 122
- NULL_TERM_CHAR
 - osclbase, 32
- num_child_nodes
 - MM_Stats_CB, 162
- num_elements
 - OscI_Linked_List_Base, 214
- num_fragments
 - BufFragGroup, 121
- num_reserved_fragments
 - MediaData, 143
- numAllocFails
 - MM_Stats_t, 164
- numAllocs
 - MM_Stats_t, 164
- numBytes
 - MM_Stats_t, 164
- numelems
 - OscI_Queue_Base, 239
 - OscI_Vector_Base, 291
- numFrag
 - OscIBinStream, 337
- octet
 - osclbase, 34
- Offset
 - OscIAsyncFileBuffer, 318
- Open
 - OscI_File, 181
 - OscIAsyncFile, 315, 316
 - OscIDNSI, 352
 - OscIDNSIBase, 354
 - OscIFileCache, 400
 - OscINativeFile, 462
 - OscISocketI, 536
 - OscISocketIBase, 542
 - OscISocketServRequestList, 559
- OpenSession
 - OscIComponentRegistry, 342
- operator *
 - OscI_Rb_Tree_Const_Iterator, 246
 - OscI_Rb_Tree_Iterator, 249
 - OscI_TagTree::const_iterator, 272
 - OscI_TagTree::iterator, 275
 - OscIExclusiveArrayPtr, 379
 - OscIExclusivePtr, 382
 - OscIExclusivePtrA, 385
 - OSCLMemAutoPtr, 435
 - OscISharedPtr, 527
 - OscISingleton, 531
 - OscITLS, 591
 - OscITLSEx, 593
- operator *=
 - TimeValue, 653
- operator delete
 - MM_AllocInfo, 149
 - MM_AllocNode, 150
 - MM_FailInsertParam, 161
 - MM_Stats_CB, 162
 - MM_Stats_t, 164
 - oscl_mem.h, 714
 - OscIErrorAllocator, 373
 - osclmemory, 59
 - OscIMemStatsNode, 457
- operator delete[]
 - osclmemory, 59
- operator new

- MM_AllocInfo, 149
- MM_AllocNode, 150
- MM_FailInsertParam, 161
- MM_Stats_CB, 162
- MM_Stats_t, 164
- oscl_mem.h, 714
- osclconfig_global_placement_new.h, 811
- OscLErrorAllocator, 373
- osclmemory, 59
- OscMemStatsNode, 457
- operator new[]
 - osclmemory, 59
- operator T *
 - OscDoubleRunner, 368
- operator TheClass *
 - OscSharedPtr, 528
- operator!=
 - OscRb_Tree_Const_Iterator, 246
 - OscRb_Tree_Iterator, 249
 - OSCL_String, 260
 - OscTagTree::const_iterator, 272
 - OscTagTree::iterator, 275
 - OSCL_wString, 304
 - OscAOSStatus, 313
 - OscUuid, 610
 - StrCSumPtrLen, 644
 - StrPtrLen, 647
 - TimeValue, 654
 - WStrPtrLen, 658
- operator()
 - Osc_Less, 204
 - Osc_Map::value_compare, 222
 - Osc_Select1st, 254
 - Osc_Tag_Base, 266
- operator+
 - osclbase, 36, 37
- operator++
 - OscRb_Tree_Const_Iterator, 246
 - OscRb_Tree_Iterator, 249
 - OscTagTree::const_iterator, 272
 - OscTagTree::iterator, 275
 - OscDoubleRunner, 368
- operator+=
 - NTPTIME, 167
 - OSCL_String, 260
 - OSCL_wString, 304
 - TimeValue, 653
- operator-
 - NTPTIME, 167
 - osclbase, 37
- operator-
 - OscRb_Tree_Const_Iterator, 246
 - OscRb_Tree_Iterator, 249
 - OscTagTree::const_iterator, 272
- OscTagTree::iterator, 275
- OscDoubleRunner, 368
- operator-=
 - TimeValue, 653
- operator->
 - OscRb_Tree_Const_Iterator, 246
 - OscRb_Tree_Iterator, 249
 - OscTagTree::const_iterator, 272
 - OscTagTree::iterator, 275
 - OscExclusiveArrayPtr, 379
 - OscExclusivePtr, 382
 - OscExclusivePtrA, 385
 - OSCLMemAutoPtr, 435
 - OscSharedPtr, 528
 - OscSingleton, 531
 - OscTLS, 591
 - OscTLSEx, 593
- operator<
 - OSCL_String, 260
 - Osc_Tag, 263
 - OSCL_wString, 304
 - OscAOSStatus, 313
 - TimeValue, 654
- operator<<
 - OscBinOStreamBigEndian, 331
 - OscBinOStreamLittleEndian, 333
- operator<=
 - OSCL_String, 261
 - OSCL_wString, 304
 - OscAOSStatus, 313
 - TimeValue, 654
- operator=
 - NTPTIME, 167, 168
 - OSCL_FastString, 176
 - OSCL_HeapStringA, 200
 - Osc_Map, 220
 - OscRb_Tree, 242
 - OSCL_String, 261
 - OscTagTree, 270
 - Osc_Vector, 286
 - OSCL_wFastString, 293
 - OSCL_wHeapStringA, 299
 - OSCL_wString, 304
 - OscAOSStatus, 313
 - OscComponentRegistryElement, 344
 - OscExclusiveArrayPtr, 379
 - OscExclusivePtr, 382
 - OscExclusivePtrA, 385
 - OSCLMemAutoPtr, 435
 - OscRefCountMemFrag, 496
 - OscSharedPtr, 528
 - osclutil, 71–73
 - OscUuid, 610
 - StrCSumPtrLen, 644

- StrPtrLen, [647](#)
- TimeValue, [653](#)
- WStrPtrLen, [658](#)
- operator==
 - OscI_Rb_Tree_Const_Iterator, [246](#)
 - OscI_Rb_Tree_Iterator, [249](#)
 - OSCL_String, [261](#)
 - OscI_TagTree::const_iterator, [272](#)
 - OscI_TagTree::iterator, [275](#)
 - OSCL_wString, [304](#)
 - OscIAOStatus, [313](#)
 - osclbase, [37](#)
 - OscINetworkAddress, [465](#)
 - OscIUuid, [610](#)
 - StrCSumPtrLen, [644](#)
 - StrPtrLen, [647](#)
 - TimeValue, [654](#)
 - WStrPtrLen, [658](#)
- operator>
 - OSCL_String, [261](#)
 - OSCL_wString, [304](#)
 - OscIAOStatus, [313](#)
 - TimeValue, [654](#)
- operator>=
 - OSCL_String, [261](#)
 - OSCL_wString, [304](#)
 - OscIAOStatus, [313](#)
 - TimeValue, [654](#)
- operator>>
 - OscIBinIStreamBigEndian, [325](#)
 - OscIBinIStreamLittleEndian, [328](#)
- operator[]
 - OscI_Map, [220](#)
 - OSCL_String, [261](#)
 - OscI_TagTree, [270](#)
 - OscI_Vector, [286](#)
 - OSCL_wString, [304](#)
- optype
 - OSCL_FastString, [175](#)
 - OSCL_HeapString, [196](#)
 - OSCL_HeapStringA, [198](#)
 - OSCL_StackString, [256](#)
 - OSCL_wFastString, [292](#)
 - OSCL_wHeapString, [296](#)
 - OSCL_wHeapStringA, [298](#)
 - OSCL_wStackString, [301](#)
- OSCL Base, [25](#)
- OSCL config, [21](#)
- OSCL Error, [84](#)
- OSCL Init, [106](#)
- OSCL IO, [94](#)
- OSCL Memory, [46](#)
- OSCL Proc, [102](#)
- OSCL Util, [62](#)
- OSCL_ABS
 - osclbase, [32](#)
- oscl_abs
 - osclutil, [73](#)
- OSCL_AF_INET
 - osclconfig_io.h, [816](#)
- OscI_Alloc, [169](#)
 - allocate, [169](#)
 - allocate_fl, [169](#)
- OSCL_ALLOC_DELETE
 - osclmemory, [51](#)
- OSCL_ALLOC_NEW
 - osclmemory, [52](#)
- oscl_aostatus.h, [659](#)
- OSCL_ARRAY_DELETE
 - osclmemory, [52](#)
- OSCL_ARRAY_NEW
 - osclmemory, [52](#)
- OSCL_ASCII_CASE_MAGIC_BIT
 - osclutil, [83](#)
- oscl_asin
 - osclutil, [73](#)
- OSCL_ASSERT
 - osclbase, [32](#)
- OSCL_Assert
 - osclbase, [37](#)
- oscl_assert.h, [660](#)
- OSCL_ASSERT_ALWAYS
 - osclconfig, [22](#)
- oscl_atan
 - osclutil, [73](#)
- OSCL_AUDIT_ARRAY_NEW
 - osclmemory, [52](#)
- OSCL_AUDIT_CALLOC
 - osclmemory, [53](#)
- OSCL_AUDIT_MALLOC
 - osclmemory, [53](#)
- OSCL_AUDIT_NEW
 - osclmemory, [53](#)
- OSCL_AUDIT_REALLOC
 - osclmemory, [54](#)
- OSCL_BAD_ALLOC_EXCEPTION_CODE
 - osclerror, [87](#)
- oscl_base.h, [661](#)
- oscl_base_alloc.h, [662](#)
- oscl_base_macros.h, [663](#)
- oscl_bin_stream.h, [664](#)
- OSCL_BYPASS_MEMMGT
 - osclconfig_memory.h, [828](#)
- oscl_byte_order.h, [665](#)
- OSCL_BYTE_ORDER_BIG_ENDIAN
 - osclconfig, [22](#)
- OSCL_BYTE_ORDER_LITTLE_ENDIAN
 - osclconfig, [22](#)

- OSCL_CALLOC
 - osclmemory, [54](#)
- oscl_calloc
 - osclmemory, [54](#)
- OSCL_CATCH
 - osclerror, [87](#)
- OSCL_CATCH_ANY
 - osclerror, [88](#)
- OSCL_CHAR_IS_SIGNED
 - osclconfig_limits_typedefs.h, [827](#)
- OSCL_CHAR_IS_UNSIGNED
 - osclconfig_limits_typedefs.h, [827](#)
- oscl_chdir
 - osclio, [98](#)
- oscl_CIstrcmp
 - osclbase, [37](#)
- oscl_CIstrncmp
 - osclbase, [38](#)
- OSCL_CLEANUP_BASE_CLASS
 - osclmemory, [54](#)
- OSCL_CLOCK_HAS_DRIFT_CORRECTION
 - osclconfig_util.h, [848](#)
- OSCL_COND_EXPORT_REF
 - osclbase, [32](#)
- OSCL_COND_IMPORT_REF
 - osclbase, [32](#)
- OSCL_CONST_CAST
 - osclbase, [32](#)
- oscl_cos
 - osclutil, [73](#)
- OscL_Dealloc, [170](#)
 - deallocate, [170](#)
- OscL_DefAlloc, [171](#)
 - allocate, [171](#)
 - allocate_fl, [171](#)
 - deallocate, [171](#)
- oscl_defalloc.h, [666](#)
- OscL_DefAllocWithRefCounter, [172](#)
 - addRef, [172](#)
 - Delete, [172](#)
 - getCount, [172](#)
 - New, [173](#)
 - removeRef, [173](#)
- OSCL_DEFAULT_FREE
 - osclmemory, [55](#)
- OSCL_DEFAULT_MALLOC
 - osclmemory, [55](#)
- OSCL_DELETE
 - osclmemory, [55](#)
- OscL_DeleteFile
 - OscL_FileServer, [192](#), [193](#)
- OSCL_DISABLE_INLINES
 - osclconfig_unix_android.h, [843](#)
 - osclconfig_unix_common.h, [847](#)
- OSCL_DISABLE_WARNING_RETURN_-
TYPE_NOT_UDT
 - osclbase, [32](#)
 - osclmemory, [55](#)
- OSCL_DISABLE_WARNING_TRUNCATE_-
DEBUG_MESSAGE
 - oscl_map.h, [708](#)
 - oscl_mem.h, [714](#)
 - oscl_mem_audit.h, [717](#)
 - oscl_mem_audit_internals.h, [718](#)
 - oscl_mem_auto_ptr.h, [719](#)
 - oscl_tagtree.h, [787](#)
 - oscl_tree.h, [796](#)
 - osclbase, [32](#)
 - osclmemory, [55](#)
- oscl_dll.h, [667](#)
- OSCL_DLL_ENTRY_POINT
 - osclbase, [32](#)
- OSCL_DLL_ENTRY_POINT_DEFAULT
 - osclbase, [33](#)
- oscl_dns.h, [668](#)
- oscl_dns_gethostbyname.h, [669](#)
- oscl_dns_imp.h, [670](#)
- oscl_dns_imp_base.h, [671](#)
- oscl_dns_imp_pv.h, [672](#)
- oscl_dns_method.h, [673](#)
- oscl_dns_param.h, [674](#)
 - TDNSRequestParamAllocator, [674](#)
- oscl_dns_request.h, [675](#)
- oscl_dns_tuneables.h, [676](#)
 - PV_DNS_IS_THREAD, [676](#)
 - PV_DNS_SERVER, [676](#)
- oscl_double_list.h, [677](#)
- OSCL_DYNAMIC_CAST
 - osclbase, [33](#)
- OSCL_ERR_NONE
 - osclerror, [88](#)
- oscl_errno.h, [678](#)
- oscl_error.h, [679](#)
 - oscl_error_allocator.h, [680](#)
 - oscl_error_codes.h, [681](#)
 - oscl_error_imp.h, [682](#)
 - oscl_error_imp_cppexceptions.h, [683](#)
 - oscl_error_imp_fatalerror.h, [684](#)
 - _PV_TRAP, [684](#)
 - _PV_TRAP_NO_TLS, [684](#)
 - PVError_DoLeave, [684](#)
 - oscl_error_imp_jumps.h, [685](#)
 - _PV_TRAP, [685](#)
 - _PV_TRAP_NO_TLS, [685](#)
 - PVError_DoLeave, [686](#)
 - oscl_error_trapcleanup.h, [687](#)

- oscl_exception.h, 688
- OSCL_EXCEPTSET_FLAG
 - oscl_socket_serv_imp_pv.h, 768
- oscl_exclusive_ptr.h, 689
- oscl_exp
 - osclutil, 73
- OSCL_EXPORT_REF
 - osclconfig.h, 804
- OSCL_FastString, 174
 - OSCL_FastString, 175
- OSCL_FastString
 - ~OSCL_FastString, 175
 - chartype, 175
 - get_cstr, 176
 - get_maxsize, 176
 - get_size, 176
 - get_str, 176
 - operator=, 176
 - optype, 175
 - OSCL_FastString, 175
 - OSCL_String, 177
 - other_chartype, 175
 - set, 176, 177
 - set_length, 177
- OscL_File
 - ESymbianAccessMode_Rfile, 179
 - ESymbianAccessMode_RfileBuf, 179
 - MODE_APPEND, 179
 - MODE_BINARY, 179
 - MODE_READ, 179
 - MODE_READ_PLUS, 179
 - MODE_READWRITE, 179
 - MODE_TEXT, 179
 - SEEKCUR, 179
 - SEEKEND, 179
 - SEEKSET, 179
- OscL_File, 178
 - ~OscL_File, 180
 - AddFixedCache, 180
 - asynclilereadcancel_test, 185
 - asynclilereadwrite_test, 185
 - Close, 180
 - EndOfFile, 180
 - Flush, 181
 - GetError, 181
 - Handle, 181
 - largeasynclilereadwrite_test, 185
 - mode_type, 179
 - Open, 181
 - OscL_File, 180
 - OscL_FileServer, 193
 - OscL_FileCache, 185
 - OscL_FileCacheBuffer, 185
 - OscL_FileHandle, 403
 - Read, 182
 - RemoveFixedCache, 182
 - Seek, 182
 - seek_type, 179
 - SetAsyncReadBufferSize, 182
 - SetCacheObserver, 183
 - SetFileHandle, 183
 - SetLoggingEnable, 183
 - SetNativeAccessMode, 183
 - SetNativeBufferSize, 184
 - SetPVCacheSize, 184
 - SetSize, 184
 - SetSummaryStatsLoggingEnable, 184
 - Size, 184
 - Tell, 184
 - TSymbianAccessMode, 179
 - Write, 185
- OscL_File::OscL_FileCacheObserver, 186
- OscL_File::OscL_FileCacheObserver
 - ChooseCurCache, 186
- OscL_File::OscL_FileFixedCacheParam, 187
- OscL_File::OscL_FileFixedCacheParam
 - Contains, 187
 - iFilePosition, 187
 - iSize, 187
- oscl_file_async_read.h, 690
- OSCL_FILE_ATTRIBUTE_ARCHIVE
 - OscL_FileManager, 404
- OSCL_FILE_ATTRIBUTE_DIRECTORY
 - OscL_FileManager, 404
- OSCL_FILE_ATTRIBUTE_HIDDEN
 - OscL_FileManager, 404
- OSCL_FILE_ATTRIBUTE_NORMAL
 - OscL_FileManager, 404
- OSCL_FILE_ATTRIBUTE_READONLY
 - OscL_FileManager, 404
- OSCL_FILE_ATTRIBUTE_SYSTEM
 - OscL_FileManager, 404
- OSCL_FILE_ATTRIBUTE_TYPE
 - OscL_FileManager, 404
- OSCL_FILE_BUFFER_MAX_SIZE
 - osclconfig_io.h, 816
- oscl_file_cache.h, 691
- OSCL_FILE_CHAR_PATH_DELIMITER
 - osclio, 96
- oscl_file_dir_utils.h, 692
- oscl_file_find.h, 694
- oscl_file_handle.h, 695
- oscl_file_io.h, 696
- oscl_file_manager.h, 697
- oscl_file_native.h, 698
- oscl_file_server.h, 699
- oscl_file_stats.h, 700
- OSCL_FILE_STATS_LOGGER_NODE

- osclio, 96
- oscl_file_types.h, 701
- OSCL_FILE_WCHAR_PATH_DELIMITER
 - osclio, 96
- OscL_FileFind
 - DIR_TYPE, 188
 - E_BUFFER_TOO_SMALL, 189
 - E_INVALID_ARG, 188
 - E_INVALID_STATE, 188
 - E_MEMORY_ERROR, 189
 - E_NO_MATCH, 189
 - E_NOT_IMPLEMENTED, 189
 - E_OK, 188
 - E_OTHER, 189
 - E_PATH_NOT_FOUND, 188
 - E_PATH_TOO_LONG, 188
 - FILE_TYPE, 188
 - INVALID_TYPE, 188
- OscL_FileFind, 188
 - OscL_FileFind, 189
- OscL_FileFind
 - ~OscL_FileFind, 189
 - Close, 189
 - element_type, 188
 - error_type, 188
 - FindFirst, 189
 - FindNext, 190
 - GetElementType, 190
 - GetLastError, 190
 - OscL_FileFind, 189
- OSCL_FILEMGMT_E_ALREADY_EXISTS
 - osclio, 97
- OSCL_FILEMGMT_E_NO_MATCH
 - osclio, 97
- OSCL_FILEMGMT_E_NOT_EMPTY
 - osclio, 97
- OSCL_FILEMGMT_E_NOT_IMPLEMENTED
 - osclio, 97
- OSCL_FILEMGMT_E_OK
 - osclio, 97
- OSCL_FILEMGMT_E_PATH_NOT_FOUND
 - osclio, 97
- OSCL_FILEMGMT_E_PATH_TOO_LONG
 - osclio, 97
- OSCL_FILEMGMT_E_PERMISSION_DENIED
 - osclio, 97
- OSCL_FILEMGMT_E_SYS_SPECIFIC
 - osclio, 97
- OSCL_FILEMGMT_E_UNKNOWN
 - osclio, 97
- OSCL_FILEMGMT_ERR_TYPE
 - osclio, 97
- OSCL_FILEMGMT_MODE_DIR
 - osclio, 97
- OSCL_FILEMGMT_MODES
 - osclio, 97
- OSCL_FILEMGMT_PERMS
 - osclio, 97
- OSCL_FILEMGMT_PERMS_EXECUTE
 - osclio, 97
- OSCL_FILEMGMT_PERMS_READ
 - osclio, 97
- OSCL_FILEMGMT_PERMS_WRITE
 - osclio, 97
- OscL_FileServer, 192
 - OscL_FileServer, 192
- OscL_FileServer
 - ~OscL_FileServer, 192
 - Close, 192
 - Connect, 192
 - OscL_DeleteFile, 192, 193
 - OscL_File, 193
 - OscL_FileServer, 192
 - OscL_NativeFile, 193
- OSCL_FIRST_CATCH
 - osclerror, 88
- OSCL_FIRST_CATCH_ANY
 - osclerror, 88
- oscl_floor
 - osclutil, 73
- OSCL_FREE
 - osclmemory, 55
- oscl_free
 - osclmemory, 55
- OSCL_FSSTAT
 - osclio, 96
- oscl_fsstat, 194
 - freebytes, 194
 - totalbytes, 194
- OSCL_FUNCTION_PTR
 - osclconfig_compiler_warnings.h, 807
- oscl_getcwd
 - osclio, 98, 99
- OSCL_GetLastError
 - osclerror, 92
- OSCL_HAS_ANDROID_FILE_IO_SUPPORT
 - osclconfig.h, 804
- OSCL_HAS_ANDROID_SUPPORT
 - osclconfig.h, 804
- OSCL_HAS_ANSI_64BIT_FILE_IO_SUPPORT
 - osclconfig_io.h, 816
- OSCL_HAS_ANSI_FILE_IO_SUPPORT
 - osclconfig_io.h, 816
- OSCL_HAS_ANSI_MATH_SUPPORT
 - osclconfig_unix_android.h, 843

- osclconfig_unix_common.h, [847](#)
- OSCL_HAS_ANSI_MEMORY_FUNCS
 - osclconfig_ansi_memory.h, [805](#)
- OSCL_HAS_ANSI_STDIO_SUPPORT
 - osclconfig_unix_android.h, [843](#)
 - osclconfig_unix_common.h, [847](#)
- OSCL_HAS_ANSI_STDLIB_SUPPORT
 - osclconfig_unix_android.h, [843](#)
 - osclconfig_unix_common.h, [847](#)
- OSCL_HAS_ANSI_STRING_SUPPORT
 - osclconfig_unix_android.h, [843](#)
 - osclconfig_unix_common.h, [847](#)
- OSCL_HAS_ANSI_WIDE_STRING_SUPPORT
 - osclconfig_unix_android.h, [843](#)
 - osclconfig_unix_common.h, [847](#)
- OSCL_HAS_BASIC_LOCK
 - osclconfig_unix_android.h, [843](#)
 - osclconfig_unix_common.h, [847](#)
- OSCL_HAS_BERKELEY_SOCKETS
 - osclconfig, [22](#)
 - osclconfig_io.h, [816](#)
- OSCL_HAS_ERRNO_H
 - osclconfig_error.h, [808](#)
- OSCL_HAS_EXCEPTIONS
 - osclconfig_error.h, [808](#)
- OSCL_HAS_GLOB
 - osclconfig_io.h, [816](#)
- OSCL_HAS_GLOBAL_NEW_DELETE
 - osclconfig_memory.h, [828](#)
 - osclmemory, [55](#)
- OSCL_HAS_GLOBAL_VARIABLE_SUPPORT
 - osclconfig_unix_android.h, [843](#)
 - osclconfig_unix_common.h, [847](#)
- OSCL_HAS_HEAP_BASE_SUPPORT
 - osclconfig_memory.h, [828](#)
- OSCL_HAS_LARGE_FILE_SUPPORT
 - osclconfig_io.h, [816](#)
- OSCL_HAS_MSWIN_FILE_IO_SUPPORT
 - osclconfig_io.h, [816](#)
- OSCL_HAS_MSWIN_PARTIAL_SUPPORT
 - osclconfig, [22](#)
 - osclconfig_unix_android.h, [843](#)
 - osclconfig_unix_common.h, [847](#)
- OSCL_HAS_MSWIN_SUPPORT
 - osclconfig, [22](#)
 - osclconfig_unix_android.h, [843](#)
 - osclconfig_unix_common.h, [847](#)
- OSCL_HAS_MSWIN_TIME_SUPPORT
 - osclconfig, [22](#)
 - osclconfig_unix_android.h, [843](#)
 - osclconfig_unix_common.h, [847](#)
- OSCL_HAS_NATIVE_FILE_CACHE_ENABLE
 - osclconfig_io.h, [816](#)
- OSCL_HAS_NATIVE_INT64_TYPE
 - osclconfig_unix_android.h, [843](#)
 - osclconfig_unix_common.h, [847](#)
- OSCL_HAS_NATIVE_UINT64_TYPE
 - osclconfig_unix_android.h, [843](#)
 - osclconfig_unix_common.h, [847](#)
- OSCL_HAS_NON_PREEMPTIVE_THREAD_SUPPORT
 - osclconfig_proc_unix_android.h, [835](#)
 - osclconfig_proc_unix_common.h, [837](#)
- OSCL_HAS_PACKED_STRUCT
 - osclconfig.h, [804](#)
- OSCL_HAS_PRAGMA_PACK
 - osclconfig, [22](#)
- OSCL_HAS_PTHREAD_SUPPORT
 - osclconfig, [22](#)
 - osclconfig_proc_unix_android.h, [835](#)
 - osclconfig_proc_unix_common.h, [837](#)
- OSCL_HAS_PV_C_OS_API_MEMORY_FUNCS
 - osclconfig, [23](#)
- OSCL_HAS_PV_C_OS_SUPPORT
 - osclconfig, [23](#)
- OSCL_HAS_PV_C_OS_TIME_FUNCS
 - osclconfig, [23](#)
- OSCL_HAS_PV_FILE_CACHE
 - osclconfig_io.h, [816](#)
- OSCL_HAS_RUNTIME_LIB_LOADING_SUPPORT
 - osclconfig_lib.h, [825](#)
- OSCL_HAS_SAVAJE_IO_SUPPORT
 - osclconfig, [23](#)
- OSCL_HAS_SAVAJE_SUPPORT
 - osclconfig, [23](#)
- OSCL_HAS_SEM_TIMEDWAIT_SUPPORT
 - osclconfig, [23](#)
 - osclconfig_proc_unix_android.h, [835](#)
 - osclconfig_proc_unix_common.h, [837](#)
- OSCL_HAS_SETJMP_H
 - osclconfig_error.h, [808](#)
- OSCL_HAS_SINGLETON_SUPPORT
 - osclbase, [33](#)
- OSCL_HAS_SOCKET_SUPPORT
 - osclconfig_io.h, [816](#)
- OSCL_HAS_SYMBIAN_COMPATIBLE_IO_FUNCTION
 - osclconfig, [23](#)
 - osclconfig_io.h, [816](#)
- OSCL_HAS_SYMBIAN_DNS_SERVER
 - osclconfig, [23](#)
 - osclconfig_io.h, [816](#)

- OSCL_HAS_SYMBIAN_ERRORTRAP
 - osclconfig, 23
 - osclconfig_error.h, 808
- OSCL_HAS_SYMBIAN_MATH
 - osclconfig, 23
 - osclconfig_util.h, 848
- OSCL_HAS_SYMBIAN_MEMORY_FUNCS
 - osclconfig, 23
 - osclconfig_memory.h, 828
- OSCL_HAS_SYMBIAN_SCHEDULER
 - osclconfig, 23
 - osclconfig_proc_unix_android.h, 835
 - osclconfig_proc_unix_common.h, 837
- OSCL_HAS_SYMBIAN_SOCKET_SERVER
 - osclconfig, 23
 - osclconfig_io.h, 816
- OSCL_HAS_SYMBIAN_SUPPORT
 - osclconfig, 23
 - osclconfig_unix_android.h, 843
 - osclconfig_unix_common.h, 847
- OSCL_HAS_SYMBIAN_TIMERS
 - osclconfig, 23
 - osclconfig_util.h, 848
- OSCL_HAS_THREAD_SUPPORT
 - osclconfig_proc_unix_android.h, 835
 - osclconfig_proc_unix_common.h, 837
- OSCL_HAS_TLS_SUPPORT
 - osclconfig_unix_android.h, 843
 - osclconfig_unix_common.h, 847
- OSCL_HAS_UNICODE_SUPPORT
 - osclconfig_unix_android.h, 843
 - osclconfig_unix_common.h, 847
- OSCL_HAS_UNIX_SUPPORT
 - osclconfig, 23
 - osclconfig_unix_android.h, 843
 - osclconfig_unix_common.h, 847
- OSCL_HAS_UNIX_TIME_FUNCS
 - osclconfig, 23
 - osclconfig_time.h, 838
- oscl_heapbase.h, 702
- OSCL_HeapString, 195
 - osclutil, 73, 74
- OSCL_HeapString
 - chartype, 196
 - optype, 196
 - OSCL_String, 196
 - other_chartype, 196
- OSCL_HeapStringA, 197
 - OSCL_HeapStringA, 198, 199
- OSCL_HeapStringA
 - ~OSCL_HeapStringA, 199
 - chartype, 198
 - get_cstr, 199
 - get_maxsize, 199
 - get_size, 200
 - get_str, 200
 - operator=, 200
 - optype, 198
 - OSCL_HeapStringA, 198, 199
 - OSCL_String, 201
 - other_chartype, 198
 - set, 200, 201
- OSCL_IMPORT_REF
 - osclconfig.h, 804
- oscl_init.h, 703
- OSCL_INLINE
 - osclbase, 33
- OscL_Int64_Utils, 202
 - get_int64_lower32, 203
 - get_int64_middle32, 203
 - get_int64_upper32, 203
 - get_uint64_lower32, 203
 - get_uint64_middle32, 203
 - get_uint64_upper32, 203
 - set_int64, 203
 - set_uint64, 203
- oscl_int64_utils.h, 704
 - _OscInteger64Transport, 704
- OSCL_INTEGERS_WORD_ALIGNED
 - osclconfig, 23
- OSCL_IO_EXTENSION_MAXLEN
 - osclio, 96
- OSCL_IO_FILENAME_MAXLEN
 - osclio, 96
- oscl_ip_socket.h, 705
- OSCL_IPPROTO_IP
 - osclconfig_io.h, 816
- OSCL_IPPROTO_TCP
 - osclconfig_io.h, 816
- OSCL_IPPROTO_UDP
 - osclconfig_io.h, 816
- oscl_isdigit
 - osclutil, 68
- OSCL_IsErrnoSupported
 - osclerror, 92
- oscl_isLetter
 - osclbase, 38
- OSCL_JUMP_MAX_JUMP_MARKS
 - osclerror, 88
- OSCL_LAST_CATCH
 - osclerror, 88
- OSCL_LEAVE
 - osclerror, 89
- OscL_Less, 204
 - operator(), 204
- OSCL_LIB_READ_DEBUG_LIBS
 - osclconfig_lib.h, 825
- OscL_Linked_List, 205

- ~OscL_Linked_List, 205
- add_element, 206
- add_to_front, 206
- check_list, 206
- clear, 206
- dequeue_element, 206
- get_element, 206
- get_first, 207
- get_index, 207
- get_next, 207
- get_num_elements, 207
- insert_element, 207
- move_to_end, 208
- move_to_front, 208
- OscL_Linked_List, 205
- remove_element, 208
- oscl_linked_list.h, 706
- OscL_Linked_List_Base, 210
 - ~OscL_Linked_List_Base, 211
 - add_element, 211
 - add_to_front, 211
 - check_list, 211
 - construct, 211
 - destroy, 211
 - get_element, 211
 - get_first, 212
 - get_index, 212
 - get_next, 212
 - head, 214
 - insert_element, 212
 - iterator, 214
 - move_to_end, 212
 - move_to_front, 213
 - num_elements, 214
 - remove_element, 213
 - sizeof_T, 214
 - tail, 214
- oscl_lock_base.h, 707
- oscl_log
 - osclutil, 74
- oscl_log10
 - osclutil, 74
- OSCL_MALLOC
 - osclmemory, 56
- oscl_malloc
 - osclmemory, 56
- OscL_Map, 215
 - begin, 218
 - clear, 218
 - const_iterator, 217
 - const_reference, 217
 - count, 218
 - empty, 218
 - end, 218
 - equal_range, 218
 - erase, 219
 - find, 219
 - insert, 219
 - iterator, 217
 - key_comp, 220
 - key_compare, 217
 - key_type, 217
 - lower_bound, 220
 - max_size, 220
 - operator=, 220
 - operator[], 220
 - OscL_Map, 217
 - pair_citerator_citerator, 217
 - pair_iterator_bool, 217
 - pair_iterator_iterator, 217
 - pointer, 217
 - reference, 217
 - self, 217
 - size, 220
 - size_type, 217
 - upper_bound, 220, 221
 - value_comp, 221
 - value_type, 217
- oscl_map.h, 708
 - OSCL_DISABLE_WARNING_-
TRUNCATE_DEBUG_MESSAGE,
708
- OscL_Map::value_compare, 222
 - comp, 222
 - operator(), 222
 - OscL_Map< Key, T, Alloc, Compare >, 222
 - value_compare, 222
- OscL_Map< Key, T, Alloc, Compare >
 - OscL_Map::value_compare, 222
- oscl_math.h, 709
- OSCL_MAX
 - osclbase, 33
- OSCL_MAX_TRAP_LEVELS
 - osclerror, 89
- oscl_media_data.h, 710
- oscl_media_status.h, 711
- oscl_mem.h, 712
 - operator delete, 714
 - operator new, 714
 - OSCL_DISABLE_WARNING_-
TRUNCATE_DEBUG_MESSAGE,
714
- oscl_mem_align.h, 715
- oscl_mem_aligned_size
 - osclmemory, 59
 - OscLMemPoolAllocator, 441
- oscl_mem_audit.h, 716

- OSCL_DISABLE_WARNING_-
TRUNCATE_DEBUG_MESSAGE,
717
- oscl_mem_audit_internals.h, 718
 - OSCL_DISABLE_WARNING_-
TRUNCATE_DEBUG_MESSAGE,
718
- oscl_mem_auto_ptr.h, 719
 - OSCL_DISABLE_WARNING_-
TRUNCATE_DEBUG_MESSAGE,
719
- oscl_mem_basic_functions.h, 720
- oscl_mem_inst.h, 721
- oscl_mem_mempool.h, 722
- oscl_memcmp
 - osclmemory, 60
- oscl_memcpy
 - osclmemory, 60
- OSCL_MEMFRAG_PTR_BEFORE_LEN
 - osclconfig_unix_android.h, 843
 - osclconfig_unix_common.h, 847
- oscl_memmove
 - osclmemory, 60
- oscl_memmove32
 - osclmemory, 60
- oscl_mempool_allocator.h, 723
- oscl_memset
 - osclmemory, 61
- oscl_memsize_t
 - osclconfig_ansi_memory.h, 805
- OSCL_MIN
 - osclbase, 33
- oscl_mkdir
 - osclio, 99
- OscL_MTLlinked_List, 224
 - ~OscL_MTLlinked_List, 224
 - add_element, 225
 - add_to_front, 225
 - dequeue_element, 225
 - get_element, 225
 - get_index, 225
 - move_to_end, 225
 - move_to_front, 226
 - OscL_MTLlinked_List, 224
 - remove_element, 226
 - the_list, 226
- oscl_mutex.h, 724
 - OscLNoYieldMutex, 724
- oscl_namestring.h, 725
- OSCL_NATIVE_INT64_TYPE
 - osclconfig_unix_android.h, 843
 - osclconfig_unix_common.h, 847
- OSCL_NATIVE_UINT64_TYPE
 - osclconfig.h, 804
- osclconfig_unix_android.h, 843
- osclconfig_unix_common.h, 847
- OSCL_NATIVE_WCHAR_TYPE
 - osclconfig_unix_android.h, 843
 - osclconfig_unix_common.h, 847
- OSCL_NEW
 - osclmemory, 56
- oscl_opaque_type.h, 726
- OscL_Opaque_Type_Alloc, 228
 - allocate, 228
 - construct, 228
 - deallocate, 228
 - destroy, 228
- OscL_Opaque_Type_Alloc_LL, 229
 - allocate, 229
 - compare_data, 229
 - construct, 229
 - deallocate, 229
 - destroy, 229
 - get_data, 230
 - get_next, 230
 - set_next, 230
- OscL_Opaque_Type_Compare, 231
 - compare_EQ, 231
 - compare_LT, 231
 - swap, 231
- OSCL_PACKED_STRUCT_BEGIN
 - osclconfig.h, 804
- OSCL_PACKED_STRUCT_END
 - osclconfig.h, 804
- OSCL_PACKED_VAR
 - osclbase, 33
 - osclconfig.h, 804
- OscL_Pair, 233
 - first, 233
 - OscL_Pair, 233
 - second, 233
- OSCL_PERF_SUMMARY_LOGGING
 - osclproc, 104
- OSCL_PLACEMENT_NEW
 - osclmemory, 56
- oscl_pow
 - osclutil, 74
- oscl_priqueue.h, 727
- oscl_priqueue_test
 - OscLPriorityQueue, 472
- oscl_procstatus.h, 728
- OscL_Queue, 234
 - ~OscL_Queue, 235
 - back, 235
 - clear, 235
 - const_reference, 235
 - front, 236
 - OscL_Queue, 235

- pointer, 235
- pop, 236
- push, 236
- reference, 235
- size_type, 235
- value_type, 235
- oscl_queue.h, 729
- OscL_Queue_Base, 237
 - ~OscL_Queue_Base, 237
 - bufsize, 239
 - capacity, 238
 - clear, 238
 - construct, 238
 - destroy, 238
 - elems, 239
 - empty, 238
 - ifront, 239
 - irear, 239
 - numelems, 239
 - pop, 238
 - push, 238
 - reserve, 238
 - size, 238
 - sizeof_T, 239
- oscl_rand.h, 730
- OSCL_RAND_MAX
 - osclconfig_util.h, 848
- OscL_Rb_Tree, 240
 - ~OscL_Rb_Tree, 242
 - begin, 242
 - clear, 242
 - const_iterator, 242
 - const_pointer, 242
 - const_reference, 242
 - count, 242
 - difference_type, 242
 - empty, 242
 - end, 242
 - equal_range, 242
 - erase, 242
 - find, 242
 - insert_unique, 242
 - iterator, 242
 - key_type, 242
 - link_type, 242
 - lower_bound, 242
 - max_size, 242
 - operator=, 242
 - OscL_Rb_Tree, 242
 - pointer, 242
 - reference, 242
 - size, 242
 - size_type, 242
 - upper_bound, 242
 - value_type, 242
- OscL_Rb_Tree_Base, 244
 - base_link_type, 244
 - rebalance, 244
 - rebalance_for_erase, 244
 - rotate_left, 244
 - rotate_right, 244
- OscL_Rb_Tree_Const_Iterator, 245
 - base_link_type, 246
 - const_iterator, 246
 - link_type, 246
 - node, 246
 - operator *, 246
 - operator!=, 246
 - operator++, 246
 - operator-, 246
 - operator->, 246
 - operator==, 246
 - OscL_Rb_Tree_Const_Iterator, 246
 - pointer, 246
 - reference, 246
 - self, 246
 - value_type, 246
- OscL_Rb_Tree_Iterator, 248
 - base_link_type, 249
 - iterator, 249
 - link_type, 249
 - node, 249
 - operator *, 249
 - operator!=, 249
 - operator++, 249
 - operator-, 249
 - operator->, 249
 - operator==, 249
 - OscL_Rb_Tree_Iterator, 249
 - pointer, 249
 - reference, 249
 - self, 249
 - value_type, 249
- OscL_Rb_Tree_Node, 251
 - link_type, 251
 - value, 251
 - value_type, 251
- OscL_Rb_Tree_Node_Base
 - black, 252
 - red, 252
- OscL_Rb_Tree_Node_Base, 252
 - base_link_type, 252
 - color, 253
 - color_type, 252
 - left, 253
 - maximum, 253
 - minimum, 253
 - parent, 253

- RedBl, 252
- right, 253
- OSCL_READSET_FLAG
 - oscl_socket_serv_imp_pv.h, 768
- OSCL_REALLOC
 - osclmemory, 56
- oscl_realloc
 - osclmemory, 56
- oscl_refcounter.h, 731
- oscl_refcounter_memfrag.h, 732
- oscl_registry_access_client.h, 733
- oscl_registry_client.h, 734
- oscl_registry_client_impl.h, 735
- oscl_registry_serv_impl.h, 736
- oscl_registry_serv_impl_global.h, 737
- oscl_registry_serv_impl_tls.h, 738
- oscl_registry_types.h, 739
- OSCL_REINTERPRET_CAST
 - osclbase, 33
- OSCL_RELEASE_BUILD
 - osclconfig.h, 804
- oscl_rename
 - osclio, 99, 100
- OSCL_REQUEST_ERR_CANCEL
 - osclproc, 105
- OSCL_REQUEST_ERR_GENERAL
 - osclproc, 105
- OSCL_REQUEST_ERR_NONE
 - osclproc, 105
- OSCL_REQUEST_PENDING
 - osclproc, 105
- oscl_rmdir
 - osclio, 100
- oscl_scheduler.h, 740
- oscl_scheduler_ao.h, 741
- oscl_scheduler_aobase.h, 742
- oscl_scheduler_readyq.h, 743
- oscl_scheduler_threadcontext.h, 744
- oscl_scheduler_tuneables.h, 745
- oscl_scheduler_types.h, 746
- OSCL_SD_BOTH
 - osclconfig_io.h, 816
- OSCL_SD_RECEIVE
 - osclconfig_io.h, 816
- OSCL_SD_SEND
 - osclconfig_io.h, 816
- Osc_Select1st, 254
 - operator(), 254
- oscl_semaphore.h, 747
- OSCL_SetLastError
 - osclerror, 92
- oscl_shared_ptr.h, 748
- oscl_sin
 - osclutil, 75
- oscl_singleton.h, 749
 - OSCL_SINGLETON_ID_CPM_PLUGIN, 750
 - OSCL_SINGLETON_ID_LAST, 750
 - OSCL_SINGLETON_ID_OMX, 750
 - OSCL_SINGLETON_ID_-OMXMASTERCORE, 750
 - OSCL_SINGLETON_ID_OSCLMEM, 750
 - OSCL_SINGLETON_ID_-OSCLREGISTRY, 750
 - OSCL_SINGLETON_ID_-PAYLOADPARSER, 750
 - OSCL_SINGLETON_ID_-PERRORTRAP, 750
 - OSCL_SINGLETON_ID_PVLOGGER, 750
 - OSCL_SINGLETON_ID_-PVMFRECOGNIZER, 750
 - OSCL_SINGLETON_ID_-PVSCHEDULER, 750
 - OSCL_SINGLETON_ID_-SDPMEDIAPARSER, 750
 - OSCL_SINGLETON_ID_TEST, 750
 - OSCL_SINGLETON_ID_TICKCOUNT, 750
 - OSCL_SINGLETON_ID_-WMDRMLOCK, 750
 - OSCL_SINGLETON_ID_CPM_PLUGIN
 - oscl_singleton.h, 750
 - OSCL_SINGLETON_ID_LAST
 - oscl_singleton.h, 750
 - OSCL_SINGLETON_ID_OMX
 - oscl_singleton.h, 750
 - OSCL_SINGLETON_ID_-OMXMASTERCORE
 - oscl_singleton.h, 750
 - OSCL_SINGLETON_ID_OSCLMEM
 - oscl_singleton.h, 750
 - OSCL_SINGLETON_ID_OSCLREGISTRY
 - oscl_singleton.h, 750
 - OSCL_SINGLETON_ID_PAYLOADPARSER
 - oscl_singleton.h, 750
 - OSCL_SINGLETON_ID_PERRORTRAP
 - oscl_singleton.h, 750
 - OSCL_SINGLETON_ID_PVLOGGER
 - oscl_singleton.h, 750
 - OSCL_SINGLETON_ID_-PVMFRECOGNIZER
 - oscl_singleton.h, 750
 - OSCL_SINGLETON_ID_PVSCHEDULER
 - oscl_singleton.h, 750
 - OSCL_SINGLETON_ID_-SDPMEDIAPARSER

- oscl_singleton.h, 750
- OSCL_SINGLETON_ID_TEST
 - oscl_singleton.h, 750
- OSCL_SINGLETON_ID_TICKCOUNT
 - oscl_singleton.h, 750
- OSCL_SINGLETON_ID_WMDRMLOCK
 - oscl_singleton.h, 750
- oscl_snprintf
 - osclutil, 75
- oscl_snprintf.h, 751
- OSCL_SOCKET_DATAGRAM
 - osclconfig_io.h, 816
- OSCL_SOCKET_STREAM
 - osclconfig_io.h, 816
- oscl_socket.h, 752
- oscl_socket_accept.h, 753
- oscl_socket_bind.h, 754
- oscl_socket_connect.h, 755
- oscl_socket_imp.h, 756
- oscl_socket_imp_base.h, 757
- oscl_socket_imp_pv.h, 758
 - PVSOCK_ERR_BAD_PARAM, 758
 - PVSOCK_ERR_NOT_IMPLEMENTED, 758
 - PVSOCK_ERR_NOT_SUPPORTED, 758
 - PVSOCK_ERR_SERV_NOT_CONNECTED, 758
 - PVSOCK_ERR_SOCKET_NO_SERV, 758
 - PVSOCK_ERR_SOCKET_NOT_CONNECTED, 758
 - PVSOCK_ERR_SOCKET_NOT_OPEN, 758
- oscl_socket_listen.h, 759
 - OSCL_SOCKET_LISTEN_H_INCLUDEDd, 759
- OSCL_SOCKET_LISTEN_H_INCLUDEDd
 - oscl_socket_listen.h, 759
- oscl_socket_method.h, 760
 - MSEC_TO_MICROSEC, 760
- oscl_socket_recv.h, 761
- oscl_socket_recv_from.h, 762
- oscl_socket_request.h, 763
- oscl_socket_send.h, 764
- oscl_socket_send_to.h, 765
- oscl_socket_serv_imp.h, 766
- oscl_socket_serv_imp_base.h, 767
- oscl_socket_serv_imp_pv.h, 768
 - OSCL_EXCEPTSET_FLAG, 768
 - OSCL_READSET_FLAG, 768
 - OSCL_WRITESET_FLAG, 768
- oscl_socket_serv_imp_reqlist.h, 769
- oscl_socket_shutdown.h, 770
- oscl_socket_stats.h
 - EOsclSocket_DataRecv, 772
 - EOsclSocket_DataSent, 772
- EOsclSocket_Except, 771
- EOsclSocket_OS, 771
- EOsclSocket_Readable, 771
- EOsclSocket_RequestAO_Canceled, 771
- EOsclSocket_RequestAO_Error, 771
- EOsclSocket_RequestAO_Success, 771
- EOsclSocket_RequestAO_Timeout, 771
- EOsclSocket_ServPoll, 771
- EOsclSocket_ServRequestCancelIssued, 772
- EOsclSocket_ServRequestComplete, 772
- EOsclSocket_ServRequestIssued, 771
- EOsclSocket_Writable, 771
- EOsclSocketServ_LastEvent, 771
- EOsclSocketServ_LoopsockError, 772
- EOsclSocketServ_LoopsockOk, 772
- EOsclSocketServ_SelectActivity, 771
- EOsclSocketServ_SelectNoActivity, 771
- EOsclSocketServ_SelectRescheduleAsap, 771
- EOsclSocketServ_SelectReschedulePoll, 771
- oscl_socket_stats.h, 771
 - TOsclSocketServStatEvent, 771
 - TOsclSocketStatEvent, 771
- oscl_socket_tuneables.h, 773
 - PV_OSCL_SOCKET_1MB_RECV_BUF, 773
 - PV_OSCL_SOCKET_SERVER_LOGGER_OUTPUT, 773
 - PV_OSCL_SOCKET_STATS_LOGGING, 773
 - PV_SOCKET_REQUEST_AO_PRIORITY, 773
 - PV_SOCKET_SERVER, 773
 - PV_SOCKET_SERVER_AO_INTERVAL_MSEC, 774
 - PV_SOCKET_SERVER_AO_PRIORITY, 774
 - PV_SOCKET_SERVER_IS_THREAD, 774
 - PV_SOCKET_SERVER_SELECT, 774
 - PV_SOCKET_SERVER_SELECT_LOOPBACK_SOCKET, 774
 - PV_SOCKET_SERVER_SELECT_TIMEOUT_MSEC, 774
 - PV_SOCKET_SERVER_THREAD_PRIORITY, 774
 - PV_SOCKET_SERVI_STATS, 774
- oscl_socket_types.h
 - EPVIPAddMembership, 776
 - EPVIPMulticastTTL, 776
 - EPVIPProtoIP, 776
 - EPVIPProtoTCP, 776

- EPVIPTOS, [776](#)
- EPVSocket, [776](#)
- EPVSocket_Last, [776](#)
- EPVSocketAccept, [776](#)
- EPVSocketBind, [776](#)
- EPVSocketBothShutdown, [776](#)
- EPVSocketCancel, [775](#)
- EPVSocketConnect, [776](#)
- EPVSocketFailure, [775](#)
- EPVSocketListen, [776](#)
- EPVSocketNotImplemented, [776](#)
- EPVSocketPending, [775](#)
- EPVSocketRecv, [776](#)
- EPVSocketRecvFrom, [776](#)
- EPVSocketRecvShutdown, [776](#)
- EPVSocketSend, [776](#)
- EPVSocketSendShutdown, [776](#)
- EPVSocketSendTo, [776](#)
- EPVSocketShutdown, [776](#)
- EPVSocketSuccess, [775](#)
- EPVSocketTimeout, [775](#)
- EPVSocketReuseAddr, [776](#)
- oscl_socket_types.h, [775](#)
 - PVNETWORKADDRESS_LEN, [775](#)
 - TPVSocketEvent, [775](#)
 - TPVSocketFxn, [776](#)
 - TPVSocketOptionLevel, [776](#)
 - TPVSocketOptionName, [776](#)
 - TPVSocketShutdown, [776](#)
- OSCL_SOCKET_IP_ADDMEMBERSHIP
 - osclconfig_io.h, [816](#)
- OSCL_SOCKET_IP_MULTICAST_TTL
 - osclconfig_io.h, [816](#)
- OSCL_SOCKET_IP_TOS
 - osclconfig_io.h, [816](#)
- OSCL_SOCKET_SOL_REUSEADDR
 - osclconfig_io.h, [816](#)
- OSCL_SOL_IP
 - osclconfig_io.h, [816](#)
- OSCL_SOL_SOCKET
 - osclconfig_io.h, [816](#)
- OSCL_SOL_TCP
 - osclconfig_io.h, [816](#)
- OSCL_SOL_UDP
 - osclconfig_io.h, [816](#)
- oscl_sqrt
 - osclutil, [75](#)
- OSCL_StackString, [255](#)
 - osclutil, [75, 76](#)
- OSCL_StackString
 - chartype, [256](#)
 - optype, [256](#)
 - OSCL_String, [256](#)
 - other_chartype, [256](#)
- oscl_stat
 - osclio, [100, 101](#)
- OSCL_STAT_BUF
 - osclio, [96](#)
- oscl_stat_buf, [257](#)
 - mode, [257](#)
 - perms, [257](#)
- oscl_statfs
 - osclio, [101](#)
- OSCL_STATIC_CAST
 - osclbase, [33](#)
- oscl_stdstring.h, [777](#)
- oscl_str_escape_xml
 - osclutil, [76](#)
- oscl_str_is_valid_utf8
 - osclutil, [76](#)
- oscl_str_need_escape_xml
 - osclutil, [77](#)
- oscl_str_ptr_len.h, [779](#)
- oscl_str_truncate_utf8
 - osclutil, [77](#)
- oscl_str_unescape_uri
 - osclutil, [77, 78](#)
- oscl_strcat
 - osclbase, [38, 39](#)
- oscl_strchr
 - osclbase, [39](#)
- oscl_strcmp
 - osclbase, [40](#)
- OSCL_StrError
 - osclerror, [92](#)
- OSCL_String, [258](#)
 - ~OSCL_String, [259](#)
 - append_rep, [259](#)
 - chartype, [259](#)
 - get_cstr, [259](#)
 - get_maxsize, [259](#)
 - get_size, [260](#)
 - get_str, [260](#)
 - hash, [260](#)
 - is_writable, [260](#)
 - operator!=, [260](#)
 - operator+/, [260](#)
 - operator<, [260](#)
 - operator<=, [261](#)
 - operator=, [261](#)
 - operator==, [261](#)
 - operator>, [261](#)
 - operator>=, [261](#)
 - operator[], [261](#)
 - OSCL_FastString, [177](#)
 - OSCL_HeapString, [196](#)
 - OSCL_HeapStringA, [201](#)
 - OSCL_StackString, [256](#)

- OSCL_String, 259
- read, 261
- set_len, 261
- set_rep, 261, 262
- setrep_to_char, 262
- write, 262
- oscl_string.h, 780
- oscl_string_containers.h, 781
- oscl_string_rep.h, 782
- oscl_string_uri.h, 783
- oscl_string_utf8.h, 784
- oscl_string_utils.h, 785
- oscl_string_xml.h, 786
- oscl_strlen
 - osclbase, 40
- oscl_strncat
 - osclbase, 40, 41
- oscl_strncmp
 - osclbase, 41
- oscl_strncpy
 - osclbase, 42
- oscl_strchr
 - osclbase, 42, 43
- oscl_strset
 - osclbase, 43
- oscl_strstr
 - osclbase, 43, 44
- OscL_Tag, 263
 - ~OscL_Tag, 263
 - operator<, 263
 - OscL_Tag, 263
 - tag, 263
 - tagAllocator, 263
- OscL_Tag_Base, 265
 - operator(), 266
 - size_type, 266
 - tag_ancestor, 266
 - tag_base_type, 266
 - tag_base_unit, 266
 - tag_cmp, 266
 - tag_copy, 266
 - tag_depth, 266
 - tag_len, 266
- OscL_TagTree, 267
 - OscL_TagTree, 268
- OscL_TagTree
 - ~OscL_TagTree, 268
 - begin, 268
 - children_type, 268
 - clear, 269
 - count, 269
 - empty, 269
 - end, 269
 - erase, 269
 - find, 269
 - insert, 270
 - map_type, 268
 - node_ptr, 268
 - node_type, 268
 - operator=, 270
 - operator[], 270
 - OscL_TagTree, 268
 - pair_iterator_bool, 268
 - size, 270
 - size_type, 268
 - tag_base_type, 268
 - tag_type, 268
 - value_type, 268
- oscl_tagtree.h, 787
 - OSCL_DISABLE_WARNING_-
TRUNCATE_DEBUG_MESSAGE,
787
- OscL_TagTree::const_iterator, 271
- OscL_TagTree::const_iterator
 - const_iterator, 272
 - mapit, 272
 - mapiter, 272
 - operator *, 272
 - operator!=, 272
 - operator++, 272
 - operator-, 272
 - operator->, 272
 - operator==, 272
 - pointer, 272
 - reference, 272
 - self, 272
- OscL_TagTree::iterator, 274
- OscL_TagTree::iterator
 - iterator, 275
 - mapit, 275
 - mapiter, 275
 - operator *, 275
 - operator!=, 275
 - operator++, 275
 - operator-, 275
 - operator->, 275
 - operator==, 275
 - pointer, 275
 - reference, 275
 - self, 275
- OscL_TagTree::Node, 277
- OscL_TagTree::Node
 - children, 278
 - children_type, 278
 - depth, 278
 - Node, 278
 - parent, 278
 - sort_children, 278

- tag, 278
- value, 278
- OscI_TAlloc, 279
 - ~OscI_TAlloc, 280
 - address, 280
 - alloc_and_construct, 280
 - alloc_and_construct_fl, 280
 - allocate, 280
 - allocate_fl, 280
 - const_pointer, 280
 - const_reference, 280
 - construct, 280
 - deallocate, 280
 - destroy, 280
 - destruct_and_dealloc, 280
 - pointer, 280
 - reference, 280
 - size_type, 280
 - value_type, 280
- OscI_TAlloc::rebind, 282
 - other, 282
- oscl_tan
 - osclutil, 78
- OSCL_TCHAR
 - osclbase, 34
- oscl_tcp_socket.h, 788
- OSCL_TEMPLATED_DESTRUCTOR_CALL
 - osclbase, 33
 - osclconfig.h, 804
- oscl_thread.h
 - Start_on_creation, 789
 - Suspend_on_creation, 789
 - ThreadPriorityAboveNormal, 790
 - ThreadPriorityBelowNormal, 789
 - ThreadPriorityHighest, 790
 - ThreadPriorityLow, 789
 - ThreadPriorityLowest, 789
 - ThreadPriorityNormal, 789
 - ThreadPriorityTimeCritical, 790
- oscl_thread.h, 789
 - OscIThread_State, 789
 - OscIThreadPriority, 789
 - TOscIThreadFuncPtr, 789
- OSCL_THREAD_DECL
 - osclconfig_proc_unix_android.h, 835
 - osclconfig_proc_unix_common.h, 837
- oscl_tickcount.h, 791
- oscl_time.h, 792
- oscl_timer.h, 794
- oscl_tls.h, 795
- OSCL_TLS_BASE_SLOTS
 - osclbase, 33
- OSCL_TLS_EXTERNAL_SLOTS
 - osclbase, 33
- OSCL_TLS_GET_FUNC
 - osclconfig_unix_android.h, 843
 - osclconfig_unix_common.h, 847
- OSCL_TLS_ID_BASE_LAST
 - osclbase, 45
- OSCL_TLS_ID_ERRORHOOK
 - osclbase, 45
- OSCL_TLS_ID_MAGICNUM
 - osclbase, 45
- OSCL_TLS_ID_OSCLREGISTRY
 - osclbase, 45
- OSCL_TLS_ID_PAYLOADPARSER
 - osclbase, 45
- OSCL_TLS_ID_PVERRORTRAP
 - osclbase, 45
- OSCL_TLS_ID_PVLOGGER
 - osclbase, 45
- OSCL_TLS_ID_PVMFRECognizer
 - osclbase, 45
- OSCL_TLS_ID_PVSCHEDULER
 - osclbase, 45
- OSCL_TLS_ID_SDPMEDIAPARSER
 - osclbase, 45
- OSCL_TLS_ID_SQLITE3
 - osclbase, 45
- OSCL_TLS_ID_TEST
 - osclbase, 45
- OSCL_TLS_ID_WMDRM
 - osclbase, 45
- OSCL_TLS_IS_KEYED
 - osclconfig_unix_android.h, 843
 - osclconfig_unix_common.h, 847
- OSCL_TLS_KEY_CREATE_FUNC
 - osclconfig_unix_android.h, 843
 - osclconfig_unix_common.h, 847
- OSCL_TLS_KEY_DELETE_FUNC
 - osclconfig_unix_android.h, 843
 - osclconfig_unix_common.h, 847
- OSCL_TLS_MAX_SLOTS
 - osclbase, 33
- OSCL_TLS_STORE_FUNC
 - osclconfig_unix_android.h, 843
 - osclconfig_unix_common.h, 847
- oscl_tolower
 - osclbase, 44
- OSCL_TRAP_ALLOC_NEW
 - osclmemory, 56
- OSCL_TRAP_AUDIT_NEW
 - osclmemory, 57
- OSCL_TRAP_NEW
 - osclmemory, 57
- OSCL_TRAPSTACK_POP
 - osclerror, 89
- OSCL_TRAPSTACK_POPDEALLOC

- osclerror, 89
- OSCL_TRAPSTACK_PUSH
 - osclerror, 89
- oscl_tree.h, 796
 - OSCL_DISABLE_WARNING_-
TRUNCATE_DEBUG_MESSAGE,
796
- OSCL_TRY
 - osclerror, 89
- OSCL_TRY_NO_TLS
 - osclerror, 89
- OSCL_TStrPtrLen
 - osclutil, 68
- oscl_types.h, 797
- oscl_udp_socket.h, 798
- oscl_UnicodeToUTF8
 - osclutil, 78
- OSCL_UNSIGNED_CONST
 - osclbase, 33
 - osclconfig.h, 804
- OSCL_UNUSED_ARG
 - osclbase, 33
- OSCL_UNUSED_RETURN
 - osclbase, 34
- oscl_utf8conv.h, 799
- oscl_UTF8ToUnicode
 - osclutil, 79
- oscl_uuid.h, 800
 - BYTES_IN_UUID_ARRAY, 801
 - EMPTY_UUID, 801
 - OscUuid32, 801
 - PV_CHAR_CLOSE_BRACKET, 801
 - PV_CHAR_COMMA, 801
- Osc_Vector, 283
 - ~Osc_Vector, 284
 - back, 285
 - begin, 285
 - clear, 285
 - const_iterator, 284
 - const_reference, 284
 - destroy, 285
 - end, 285
 - erase, 285
 - front, 286
 - insert, 286
 - iterator, 284
 - operator=, 286
 - operator[], 286
 - Osc_Vector, 284
 - pointer, 284
 - pop_back, 286
 - push_back, 287
 - push_front, 287
 - reference, 284
 - value_type, 284
- oscl_vector.h, 802
- Osc_Vector_Base, 288
 - ~Osc_Vector_Base, 289
 - assign_vector, 289
 - bufsize, 291
 - capacity, 289
 - construct, 289
 - destroy, 289
 - elems, 291
 - empty, 289
 - erase, 289, 290
 - insert, 290
 - numelems, 291
 - OscPriorityQueueBase, 291
 - pop_back, 290
 - push_back, 290
 - push_front, 291
 - reserve, 291
 - size, 291
 - sizeof_T, 291
- OSCL_VIRTUAL_BASE
 - osclbase, 34
- oscl_vsnprintf
 - osclutil, 79, 81
- oscl_wchar
 - osclbase, 34
- OSCL_wFastString, 292
 - OSCL_wFastString, 293
- OSCL_wFastString
 - ~OSCL_wFastString, 293
 - chartype, 292
 - get_cstr, 293
 - get_maxsize, 293
 - get_size, 293
 - get_str, 293
 - operator=, 293
 - optype, 292
 - OSCL_wFastString, 293
 - OSCL_wString, 294
 - other_chartype, 293
 - set, 294
 - set_length, 294
- OSCL_wHeapString, 295
 - osclutil, 81
- OSCL_wHeapString
 - chartype, 296
 - optype, 296
 - OSCL_wString, 296
 - other_chartype, 296
- OSCL_wHeapStringA, 297
 - OSCL_wHeapStringA, 298
- OSCL_wHeapStringA
 - ~OSCL_wHeapStringA, 298

- chartype, 298
- get_cstr, 298
- get_maxsize, 298
- get_size, 298
- get_str, 299
- operator=, 299
- optype, 298
- OSCL_wHeapStringA, 298
- OSCL_wString, 299
- other_chartype, 298
- set, 299
- OSCL_WRITESET_FLAG
 - oscl_socket_serv_imp_pv.h, 768
- OSCL_wStackString, 300
 - osclutil, 81
- OSCL_wStackString
 - chartype, 301
 - optype, 301
 - OSCL_wString, 301
 - other_chartype, 301
- OSCL_wString, 302
 - OSCL_wFastString, 294
 - OSCL_wHeapString, 296
 - OSCL_wHeapStringA, 299
 - OSCL_wStackString, 301
 - OSCL_wString, 303
- OSCL_wString
 - ~OSCL_wString, 303
 - append_rep, 303
 - chartype, 303
 - get_cstr, 303
 - get_maxsize, 303
 - get_size, 303
 - get_str, 303
 - hash, 303
 - is_writable, 304
 - operator!=, 304
 - operator+=", 304
 - operator<, 304
 - operator<=, 304
 - operator=, 304
 - operator==, 304
 - operator>, 304
 - operator>=, 304
 - operator[], 304
 - OSCL_wString, 303
 - read, 304
 - set_len, 305
 - set_rep, 305
 - setrep_to_wide_char, 305
 - write, 305
- OSCL_ZEROIZE
 - osclproc, 104
- OsclAccept
 - osclconfig_io.h, 816
- OsclAcceptMethod, 306
- OsclAcceptMethod
 - ~OsclAcceptMethod, 306
 - Accept, 306
 - AcceptRequest, 306
 - DiscardAcceptedSocket, 306
 - GetAcceptedSocket, 306
 - NewL, 306
- OsclAcceptRequest, 307
 - OsclAcceptRequest, 307
 - OsclSocketI, 538
- OsclAcceptRequest
 - Accept, 307
 - OsclAcceptRequest, 307
- OsclActiveObject, 308
 - EPriorityHigh, 309
 - EPriorityHighest, 309
 - EPriorityIdle, 309
 - EPriorityLow, 309
 - EPriorityNominal, 309
 - OsclActiveObject, 309
 - OsclExecSchedulerCommonBase, 395
 - PVActiveBase, 614
 - PVActiveStats, 615
 - PVThreadContext, 634
- OsclActiveObject
 - ~OsclActiveObject, 309
 - AddToScheduler, 309
 - Cancel, 309
 - DoCancel, 310
 - IsBusy, 310
 - OsclActiveObject, 309
 - OsclActivePriority, 309
 - PendComplete, 310
 - PendForExec, 310
 - Priority, 310
 - RemoveFromScheduler, 310
 - RunError, 310
 - RunIfNotReady, 311
 - SetBusy, 311
 - SetStatus, 311
 - Status, 311
 - StatusRef, 311
- OsclActivePriority
 - OsclActiveObject, 309
- OsclAllocDestructDealloc, 312
- OsclAllocDestructDealloc
 - ~OsclAllocDestructDealloc, 312
- OsclAny
 - osclbase, 34
- OsclAOSStatus, 313
 - OsclAOSStatus, 313
- OsclAOSStatus

- operator!=, 313
- operator<, 313
- operator<=, 313
- operator=, 313
- operator==, 313
- operator>, 313
- operator>=, 313
- OscIAOStatus, 313
- Value, 313
- OscIAsyncFile, 314
- OscIAsyncFile
 - ~OscIAsyncFile, 315
 - Close, 315
 - Delete, 315
 - EndOfFile, 315
 - Flush, 315
 - iNumOfRun, 316
 - iNumOfRunErr, 316
 - NewL, 315
 - Open, 315, 316
 - Read, 316
 - Seek, 316
 - Size, 316
 - Tell, 316
 - Write, 316
- OscIAsyncFileBuffer, 317
- OscIAsyncFileBuffer
 - ~OscIAsyncFileBuffer, 318
 - Buffer, 318
 - CleanInUse, 318
 - HasThisOffset, 318
 - Id, 318
 - IsInUse, 318
 - IsValid, 318
 - Length, 318
 - NewL, 318
 - Offset, 318
 - SetInUse, 318
 - SetOffset, 318
 - StartAsyncRead, 318
 - UpdateData, 318
- OscIAuditCB, 319
 - OscIAuditCB, 319
- OscIAuditCB
 - OscIAuditCB, 319
 - pAudit, 319
 - pStatsNode, 319
- OscIBase
 - OscISingletonRegistry, 533
 - OscITLSRegistry, 595
- osclbase
 - _OSCL_Abort, 35
 - ALLOC_AND_CONSTRUCT, 32
 - ALLOCATE, 32
 - big_endian_to_host, 35
 - Bind, 35
 - c_bool, 34
 - CTIME_BUFFER_SIZE, 45
 - CtimeStrBuf, 34
 - host_to_big_endian, 35
 - host_to_little_endian, 36
 - int64, 34
 - ISO8601TIME_BUFFER_SIZE, 45
 - ISO8601timeStrBuf, 34
 - ISO8601ToRFC822, 36
 - little_endian_to_host, 36
 - mbchar, 34
 - MICROSECONDS, 35
 - MILLISECONDS, 35
 - MSEC_PER_SEC, 45
 - NULL, 32
 - NULL_TERM_CHAR, 32
 - octet, 34
 - operator+, 36, 37
 - operator-, 37
 - operator==, 37
 - OSCL_ABS, 32
 - OSCL_ASSERT, 32
 - OSCL_Assert, 37
 - oscl_CIstrcmp, 37
 - oscl_CIstrncmp, 38
 - OSCL_COND_EXPORT_REF, 32
 - OSCL_COND_IMPORT_REF, 32
 - OSCL_CONST_CAST, 32
 - OSCL_DISABLE_WARNING_-
RETURN_TYPE_NOT_UDT, 32
 - OSCL_DISABLE_WARNING_-
TRUNCATE_DEBUG_MESSAGE,
32
 - OSCL_DLL_ENTRY_POINT, 32
 - OSCL_DLL_ENTRY_POINT_DEFAULT,
33
 - OSCL_DYNAMIC_CAST, 33
 - OSCL_HAS_SINGLETON_SUPPORT, 33
 - OSCL_INLINE, 33
 - oscl_isLetter, 38
 - OSCL_MAX, 33
 - OSCL_MIN, 33
 - OSCL_PACKED_VAR, 33
 - OSCL_REINTERPRET_CAST, 33
 - OSCL_STATIC_CAST, 33
 - oscl_strcat, 38, 39
 - oscl_strchr, 39
 - oscl_strcmp, 40
 - oscl_strlen, 40
 - oscl_strncat, 40, 41
 - oscl_strncmp, 41
 - oscl_strncpy, 42

- oscl_strchr, [42, 43](#)
- oscl_strset, [43](#)
- oscl_strstr, [43, 44](#)
- OSCL_TCHAR, [34](#)
- OSCL_TEMPLATED_DESTRUCTOR_CALL, [33](#)
- OSCL_TLS_BASE_SLOTS, [33](#)
- OSCL_TLS_EXTERNAL_SLOTS, [33](#)
- OSCL_TLS_ID_BASE_LAST, [45](#)
- OSCL_TLS_ID_ERRORHOOK, [45](#)
- OSCL_TLS_ID_MAGICNUM, [45](#)
- OSCL_TLS_ID_OSCLREGISTRY, [45](#)
- OSCL_TLS_ID_PAYLOADPARSER, [45](#)
- OSCL_TLS_ID_PVERRORTRAP, [45](#)
- OSCL_TLS_ID_PVLOGGER, [45](#)
- OSCL_TLS_ID_PVMFRECOGNIZER, [45](#)
- OSCL_TLS_ID_PVSCHEDULER, [45](#)
- OSCL_TLS_ID_SDPMEDIAPARSER, [45](#)
- OSCL_TLS_ID_SQLITE3, [45](#)
- OSCL_TLS_ID_TEST, [45](#)
- OSCL_TLS_ID_WMDRM, [45](#)
- OSCL_TLS_MAX_SLOTS, [33](#)
- oscl_tolower, [44](#)
- OSCL_UNSIGNED_CONST, [33](#)
- OSCL_UNUSED_ARG, [33](#)
- OSCL_UNUSED_RETURN, [34](#)
- OSCL_VIRTUAL_BASE, [34](#)
- oscl_wchar, [34](#)
- OscAny, [34](#)
- OscFloat, [34](#)
- PV8601TIME_BUFFER_SIZE, [45](#)
- PV8601timeStrBuf, [34](#)
- PV8601ToRFC822, [44](#)
- PVMEM_INST_LEVEL, [34](#)
- PVOscBase_Cleanup, [44](#)
- PVOscBase_Init, [44](#)
- RFC822ToPV8601, [45](#)
- SECONDS, [35](#)
- TimeUnits, [35](#)
- TOscTlsKey, [35](#)
- uint, [35](#)
- uint64, [35](#)
- unix_ntp_offset, [45](#)
- USEC_PER_SEC, [45](#)
- OscBasicDateTimeStruct
 - osclconfig_time.h, [838](#)
- OscBasicTimeStruct
 - osclconfig_time.h, [838](#)
- OscBind
 - osclconfig_io.h, [817](#)
- OscBindMethod, [320](#)
- OscBindMethod
 - ~OscBindMethod, [320](#)
 - Bind, [320](#)
 - BindRequest, [320](#)
 - NewL, [320](#)
- OscBindRequest, [321](#)
 - OscBindRequest, [321](#)
- OscBindRequest
 - Bind, [321](#)
 - OscBindRequest, [321](#)
- OscBinIStream, [322](#)
 - OscBinIStream, [322](#)
- OscBinIStream
 - ~OscBinIStream, [322](#)
 - get, [322](#)
 - OscBinIStream, [322](#)
 - Read_uint8, [322](#)
- OscBinIStreamBigEndian, [324](#)
 - OscBinIStreamBigEndian, [325](#)
- OscBinIStreamBigEndian
 - operator>>, [325](#)
 - OscBinIStreamBigEndian, [325](#)
 - Read, [325](#)
 - Read_uint16, [325](#)
 - Read_uint32, [325](#)
- OscBinIStreamLittleEndian, [327](#)
 - OscBinIStreamLittleEndian, [328](#)
- OscBinIStreamLittleEndian
 - operator>>, [328](#)
 - OscBinIStreamLittleEndian, [328](#)
 - Read_uint16, [328](#)
 - Read_uint32, [328](#)
- OscBinOStream, [329](#)
 - OscBinOStream, [329](#)
- OscBinOStream
 - ~OscBinOStream, [329](#)
 - OscBinOStream, [329](#)
 - write, [329](#)
- OscBinOStreamBigEndian, [330](#)
 - OscBinOStreamBigEndian, [331](#)
- OscBinOStreamBigEndian
 - operator<<, [331](#)
 - OscBinOStreamBigEndian, [331](#)
 - WriteUnsignedLong, [331](#)
 - WriteUnsignedShort, [331](#)
- OscBinOStreamLittleEndian, [332](#)
 - OscBinOStreamLittleEndian, [333](#)
- OscBinOStreamLittleEndian
 - operator<<, [333](#)
 - OscBinOStreamLittleEndian, [333](#)
 - WriteUnsignedLong, [333](#)
 - WriteUnsignedShort, [333](#)
- OscBinStream, [334](#)
 - EOF_STATE, [335](#)
 - FAIL_STATE, [335](#)
 - GOOD_STATE, [335](#)
 - OscBinStream, [335](#)

- OscBinStream
 - Attach, 335
 - eof, 335
 - fail, 336
 - firstFragPtr, 337
 - fragsLeft, 337
 - good, 336
 - HaveRoomInCurrentBlock, 336
 - length, 337
 - nextFragPtr, 337
 - numFrag, 337
 - OscBinStream, 335
 - pBasePosition, 337
 - PositionInBlock, 336
 - pPosition, 337
 - ReserveSpace, 336
 - Seek, 336
 - seekFromCurrentPosition, 336
 - specialFragBuffer, 337
 - state, 337
 - state_t, 335
 - tellg, 336
- OscBuf, 338
 - OscBuf, 339
- OscBuf
 - Delete, 339
 - Des, 339
 - DesC, 339
 - iBuffer, 339
 - iLength, 339
 - iMaxLength, 339
 - Length, 339
 - NewL, 339
 - OscBuf, 339
- OscCloseSocket
 - osclconfig_io.h, 817
- OscCoeActiveScheduler
 - OscExecSchedulerBase, 389
 - OscExecSchedulerCommonBase, 395
 - PVThreadContext, 634
- OscCoeActiveSchedulerBase
 - PVThreadContext, 634
- OscCompareLess, 340
- OscCompareLess
 - compare, 340
- OscComponentFactory
 - osclutil, 68
- OscComponentRegistry, 341
 - OscComponentRegistry, 342
- OscComponentRegistry
 - ~OscComponentRegistry, 342
 - CloseSession, 342
 - FindExact, 342
 - FindHierarchical, 342
 - iComponentIdCounter, 342
 - iData, 342
 - iMutex, 342
 - iNumSessions, 342
 - OpenSession, 342
 - OscComponentRegistry, 342
 - Register, 342
 - Unregister, 342
- OscComponentRegistryData, 343
- OscComponentRegistryData
 - Find, 343
 - iVec, 343
- OscComponentRegistryElement, 344
 - OscComponentRegistryElement, 344
- OscComponentRegistryElement
 - ~OscComponentRegistryElement, 344
 - iComponentId, 344
 - iFactory, 344
 - iId, 344
 - Match, 344
 - operator=, 344
 - OscComponentRegistryElement, 344
- osclconfig
 - __int16__check__, 24
 - __int32__check__, 24
 - __int8__check__, 24
 - __uint16__check__, 24
 - __uint32__check__, 24
 - __uint8__check__, 24
 - OSCL_ASSERT_ALWAYS, 22
 - OSCL_BYTE_ORDER_BIG_ENDIAN, 22
 - OSCL_BYTE_ORDER_LITTLE_ENDIAN, 22
 - OSCL_HAS_BERKELEY_SOCKETS, 22
 - OSCL_HAS_MSWIN_PARTIAL_SUPPORT, 22
 - OSCL_HAS_MSWIN_SUPPORT, 22
 - OSCL_HAS_MSWIN_TIME_SUPPORT, 22
 - OSCL_HAS_PRAGMA_PACK, 22
 - OSCL_HAS_PTHREAD_SUPPORT, 22
 - OSCL_HAS_PV_C_OS_API_MEMORY_FUNCS, 23
 - OSCL_HAS_PV_C_OS_SUPPORT, 23
 - OSCL_HAS_PV_C_OS_TIME_FUNCS, 23
 - OSCL_HAS_SAVAJE_IO_SUPPORT, 23
 - OSCL_HAS_SAVAJE_SUPPORT, 23
 - OSCL_HAS_SEM_TIMEDWAIT_SUPPORT, 23
 - OSCL_HAS_SYMBIAN_COMPATIBLE_IO_FUNCTION, 23

- OSCL_HAS_SYMBIAN_DNS_SERVER, 23
- OSCL_HAS_SYMBIAN_ERRORTRAP, 23
- OSCL_HAS_SYMBIAN_MATH, 23
- OSCL_HAS_SYMBIAN_MEMORY_FUNCS, 23
- OSCL_HAS_SYMBIAN_SCHEDULER, 23
- OSCL_HAS_SYMBIAN_SOCKET_SERVER, 23
- OSCL_HAS_SYMBIAN_SUPPORT, 23
- OSCL_HAS_SYMBIAN_TIMERS, 23
- OSCL_HAS_UNIX_SUPPORT, 23
- OSCL_HAS_UNIX_TIME_FUNCS, 23
- OSCL_INTEGERS_WORD_ALIGNED, 23
- osclconfig.h, 803
 - __TFS__, 804
 - OSCL_EXPORT_REF, 804
 - OSCL_HAS_ANDROID_FILE_IO_SUPPORT, 804
 - OSCL_HAS_ANDROID_SUPPORT, 804
 - OSCL_HAS_PACKED_STRUCT, 804
 - OSCL_IMPORT_REF, 804
 - OSCL_NATIVE_UINT64_TYPE, 804
 - OSCL_PACKED_STRUCT_BEGIN, 804
 - OSCL_PACKED_STRUCT_END, 804
 - OSCL_PACKED_VAR, 804
 - OSCL_RELEASE_BUILD, 804
 - OSCL_TEMPLATED_DESTRUCTOR_CALL, 804
 - OSCL_UNSIGNED_CONST, 804
 - PVLOGGER_INST_LEVEL, 804
- osclconfig_ansi_memory.h, 805
 - OSCL_HAS_ANSI_MEMORY_FUNCS, 805
 - oscl_memsize_t, 805
- osclconfig_check.h, 806
- osclconfig_compiler_warnings.h, 807
 - OSCL_FUNCTION_PTR, 807
- osclconfig_error.h, 808
 - OSCL_HAS_ERRNO_H, 808
 - OSCL_HAS_EXCEPTIONS, 808
 - OSCL_HAS_SETJMP_H, 808
 - OSCL_HAS_SYMBIAN_ERRORTRAP, 808
- osclconfig_error_check.h, 809
- osclconfig_global_new_delete.h, 810
- osclconfig_global_placement_new.h, 811
 - operator new, 811
- osclconfig_io.h, 812
 - _FILE_OFFSET_BITS, 816
 - OSCL_AF_INET, 816
 - OSCL_FILE_BUFFER_MAX_SIZE, 816
 - OSCL_HAS_ANSI_64BIT_FILE_IO_SUPPORT, 816
 - OSCL_HAS_ANSI_FILE_IO_SUPPORT, 816
 - OSCL_HAS_BERKELEY_SOCKETS, 816
 - OSCL_HAS_GLOB, 816
 - OSCL_HAS_LARGE_FILE_SUPPORT, 816
 - OSCL_HAS_MSWIN_FILE_IO_SUPPORT, 816
 - OSCL_HAS_NATIVE_FILE_CACHE_ENABLE, 816
 - OSCL_HAS_PV_FILE_CACHE, 816
 - OSCL_HAS_SOCKET_SUPPORT, 816
 - OSCL_HAS_SYMBIAN_COMPATIBLE_IO_FUNCTION, 816
 - OSCL_HAS_SYMBIAN_DNS_SERVER, 816
 - OSCL_HAS_SYMBIAN_SOCKET_SERVER, 816
 - OSCL_IPPROTO_IP, 816
 - OSCL_IPPROTO_TCP, 816
 - OSCL_IPPROTO_UDP, 816
 - OSCL_SD_BOTH, 816
 - OSCL_SD_RECEIVE, 816
 - OSCL_SD_SEND, 816
 - OSCL SOCK_DGRAM, 816
 - OSCL SOCK_STREAM, 816
 - OSCL SOCKOPT_IP_ADDMEMBERSHIP, 816
 - OSCL SOCKOPT_IP_MULTICAST_TTL, 816
 - OSCL SOCKOPT_IP_TOS, 816
 - OSCL SOCKOPT_SOL_REUSEADDR, 816
 - OSCL_SOL_IP, 816
 - OSCL_SOL_SOCKET, 816
 - OSCL_SOL_TCP, 816
 - OSCL_SOL_UDP, 816
 - OscIAccept, 816
 - OscIBind, 817
 - OscICloseSocket, 817
 - OscIConnect, 817
 - OscIConnectComplete, 817
 - OscIGetAsyncSockErr, 817
 - OscIGetDottedAddr, 817
 - OscIGetDottedAddrVector, 818
 - OscIGetHostbyname, 818
 - OscIGetPeerName, 818
 - OscIJoin, 818
 - OscIListen, 819

- OscIMakeInAddr, [819](#)
- OscIMakeSockAddr, [819](#)
- OscIPipe, [819](#)
- OscIReadFD, [819](#)
- OscIRecv, [819](#)
- OscIRecvFrom, [819](#)
- OscISend, [820](#)
- OscISendTo, [820](#)
- OscISetNonBlocking, [820](#)
- OscISetRecvBufferSize, [820](#)
- OscISetSockOpt, [820](#)
- OscIShutdown, [820](#)
- OscISocket, [821](#)
- OscISocketCleanup, [821](#)
- OscISocketSelect, [821](#)
- OscISocketStartup, [821](#)
- OscIUnMakeInAddr, [821](#)
- OscIUnMakeSockAddr, [822](#)
- OscIValidInetAddr, [822](#)
- OscIWriteFD, [822](#)
- TIpMReq, [822](#)
- TOscIFileOffset, [822](#)
- TOscIHostent, [822](#)
- TOscISockAddr, [822](#)
- TOscISockAddrLen, [822](#)
- TOscISocket, [822](#)
- osclconfig_io_check.h, [823](#)
 - __verify__TOscIFileOffset__defined__, [823](#)
- osclconfig_ix86.h, [824](#)
- osclconfig_lib.h, [825](#)
 - OSCL_HAS_RUNTIME_LIB_-LOADING_SUPPORT, [825](#)
 - OSCL_LIB_READ_DEBUG_LIBS, [825](#)
 - PV_DYNAMIC_LOADING_CONFIG_-FILE_PATH, [825](#)
 - PV_RUNTIME_LIB_FILENAME_-EXTENSION, [825](#)
- osclconfig_lib_check.h, [826](#)
- osclconfig_limits_typedefs.h, [827](#)
 - OSCL_CHAR_IS_SIGNED, [827](#)
 - OSCL_CHAR_IS_UNSIGNED, [827](#)
- osclconfig_memory.h, [828](#)
 - OSCL_BYPASS_MEMMGT, [828](#)
 - OSCL_HAS_GLOBAL_NEW_DELETE, [828](#)
 - OSCL_HAS_HEAP_BASE_SUPPORT, [828](#)
 - OSCL_HAS_SYMBIAN_MEMORY_-FUNCS, [828](#)
 - PVMEM_INST_LEVEL, [828](#)
- osclconfig_memory_check.h, [829](#)
- osclconfig_no_os.h, [830](#)
- osclconfig_proc.h, [831](#)
 - osclconfig_proc_check.h, [832](#)
 - __verify__TOscIConditionObject__defined__, [832](#)
 - __verify__TOscIMutexObject__defined__, [832](#)
 - __verify__TOscISemaphoreObject__defined__, [832](#)
 - __verify__TOscIThreadFuncArg__defined__, [832](#)
 - __verify__TOscIThreadFuncRet__defined__, [832](#)
 - __verify__TOscIThreadId__defined__, [832](#)
 - __verify__TOscIThreadObject__defined__-, [832](#)
 - osclconfig_proc_unix_android.h, [834](#)
 - OSCL_HAS_NON_PREEMPTIVE_-THREAD_SUPPORT, [835](#)
 - OSCL_HAS_PTHREAD_SUPPORT, [835](#)
 - OSCL_HAS_SEM_TIMEDWAIT_-SUPPORT, [835](#)
 - OSCL_HAS_SYMBIAN_SCHEDULER, [835](#)
 - OSCL_HAS_THREAD_SUPPORT, [835](#)
 - OSCL_THREAD_DECL, [835](#)
 - TOscIConditionObject, [835](#)
 - TOscIMutexObject, [835](#)
 - TOscISemaphoreObject, [835](#)
 - TOscIThreadFuncArg, [835](#)
 - TOscIThreadFuncRet, [835](#)
 - TOscIThreadId, [835](#)
 - TOscIThreadObject, [835](#)
 - osclconfig_proc_unix_common.h, [836](#)
 - OSCL_HAS_NON_PREEMPTIVE_-THREAD_SUPPORT, [837](#)
 - OSCL_HAS_PTHREAD_SUPPORT, [837](#)
 - OSCL_HAS_SEM_TIMEDWAIT_-SUPPORT, [837](#)
 - OSCL_HAS_SYMBIAN_SCHEDULER, [837](#)
 - OSCL_HAS_THREAD_SUPPORT, [837](#)
 - OSCL_THREAD_DECL, [837](#)
 - TOscIConditionObject, [837](#)
 - TOscIMutexObject, [837](#)
 - TOscISemaphoreObject, [837](#)
 - TOscIThreadFuncArg, [837](#)
 - TOscIThreadFuncRet, [837](#)
 - TOscIThreadId, [837](#)
 - TOscIThreadObject, [837](#)
 - osclconfig_time.h, [838](#)
 - OSCL_HAS_UNIX_TIME_FUNCS, [838](#)
 - OscIBasicDateTimeStruct, [838](#)
 - OscIBasicTimeStruct, [838](#)
 - osclconfig_time_check.h, [839](#)
 - __Validate__BasicTimeDateStruct__, [839](#)

- __Validate__BasicTimeStruct__, 839
- osclconfig_unix_android.h, 840
- _STRLIT, 843
- _STRLIT_CHAR, 843
- _STRLIT_WCHAR, 843
- INT64, 843
- INT64_HILO, 843
- OSCL_DISABLE_INLINES, 843
- OSCL_HAS_ANSI_MATH_SUPPORT, 843
- OSCL_HAS_ANSI_STDIO_SUPPORT, 843
- OSCL_HAS_ANSI_STDLIB_SUPPORT, 843
- OSCL_HAS_ANSI_STRING_SUPPORT, 843
- OSCL_HAS_ANSI_WIDE_STRING_SUPPORT, 843
- OSCL_HAS_BASIC_LOCK, 843
- OSCL_HAS_GLOBAL_VARIABLE_SUPPORT, 843
- OSCL_HAS_MSWIN_PARTIAL_SUPPORT, 843
- OSCL_HAS_MSWIN_SUPPORT, 843
- OSCL_HAS_MSWIN_TIME_SUPPORT, 843
- OSCL_HAS_NATIVE_INT64_TYPE, 843
- OSCL_HAS_NATIVE_UINT64_TYPE, 843
- OSCL_HAS_SYMBIAN_SUPPORT, 843
- OSCL_HAS_TLS_SUPPORT, 843
- OSCL_HAS_UNICODE_SUPPORT, 843
- OSCL_HAS_UNIX_SUPPORT, 843
- OSCL_MEMFRAG_PTR_BEFORE_LEN, 843
- OSCL_NATIVE_INT64_TYPE, 843
- OSCL_NATIVE_UINT64_TYPE, 843
- OSCL_NATIVE_WCHAR_TYPE, 843
- OSCL_TLS_GET_FUNC, 843
- OSCL_TLS_IS_KEYED, 843
- OSCL_TLS_KEY_CREATE_FUNC, 843
- OSCL_TLS_KEY_DELETE_FUNC, 843
- OSCL_TLS_STORE_FUNC, 843
- TOsclBasicLockObject, 843
- TOsclTlsKey, 843
- UINT64, 843
- UINT64_HILO, 843
- osclconfig_unix_common.h, 844
- _STRLIT, 847
- _STRLIT_CHAR, 847
- _STRLIT_WCHAR, 847
- INT64, 847
- INT64_HILO, 847
- OSCL_DISABLE_INLINES, 847
- OSCL_HAS_ANSI_MATH_SUPPORT, 847
- OSCL_HAS_ANSI_STDIO_SUPPORT, 847
- OSCL_HAS_ANSI_STDLIB_SUPPORT, 847
- OSCL_HAS_ANSI_STRING_SUPPORT, 847
- OSCL_HAS_ANSI_WIDE_STRING_SUPPORT, 847
- OSCL_HAS_BASIC_LOCK, 847
- OSCL_HAS_GLOBAL_VARIABLE_SUPPORT, 847
- OSCL_HAS_MSWIN_PARTIAL_SUPPORT, 847
- OSCL_HAS_MSWIN_SUPPORT, 847
- OSCL_HAS_MSWIN_TIME_SUPPORT, 847
- OSCL_HAS_NATIVE_INT64_TYPE, 847
- OSCL_HAS_NATIVE_UINT64_TYPE, 847
- OSCL_HAS_SYMBIAN_SUPPORT, 847
- OSCL_HAS_TLS_SUPPORT, 847
- OSCL_HAS_UNICODE_SUPPORT, 847
- OSCL_HAS_UNIX_SUPPORT, 847
- OSCL_MEMFRAG_PTR_BEFORE_LEN, 847
- OSCL_NATIVE_INT64_TYPE, 847
- OSCL_NATIVE_UINT64_TYPE, 847
- OSCL_NATIVE_WCHAR_TYPE, 847
- OSCL_TLS_GET_FUNC, 847
- OSCL_TLS_IS_KEYED, 847
- OSCL_TLS_KEY_CREATE_FUNC, 847
- OSCL_TLS_KEY_DELETE_FUNC, 847
- OSCL_TLS_STORE_FUNC, 847
- TOsclBasicLockObject, 847
- TOsclTlsKey, 847
- UINT64, 847
- UINT64_HILO, 847
- osclconfig_util.h, 848
- OSCL_CLOCK_HAS_DRIFT_CORRECTION, 848
- OSCL_HAS_SYMBIAN_MATH, 848
- OSCL_HAS_SYMBIAN_TIMERS, 848
- OSCL_RAND_MAX, 848
- SLEEP_ONE_SEC, 848
- osclconfig_util_check.h, 849
- OscConnect
 - osclconfig_io.h, 817
- OscConnectComplete
 - osclconfig_io.h, 817
- OscConnectMethod, 346
- OscConnectMethod
 - ~OscConnectMethod, 346

- Connect, [346](#)
- ConnectRequest, [346](#)
- NewL, [346](#)
- OscConnectRequest, [347](#)
 - OscConnectRequest, [347](#)
 - OscSocketI, [538](#)
- OscConnectRequest
 - Connect, [347](#)
 - OscConnectRequest, [347](#)
- OscDestructDealloc, [348](#)
- OscDestructDealloc
 - destruct_and_dealloc, [348](#)
- OscDNS, [349](#)
 - OscSocketServ, [554](#)
- OscDNS
 - ~OscDNS, [349](#)
 - CancelGetHostByName, [349](#)
 - GetHostByName, [350](#)
 - NewL, [350](#)
 - OscDNSRequestAO, [350](#)
- OscDNSI, [351](#)
 - OscDNSRequestAO, [363](#)
 - OscSocketServI, [556](#)
- OscDNSI
 - ~OscDNSI, [351](#)
 - Close, [351](#)
 - DNSRequestParam, [352](#)
 - GetHostByName, [351](#)
 - GetHostByNameResponseContainsAlias-Info, [352](#)
 - GetHostByNameSuccess, [352](#)
 - GetNextHost, [352](#)
 - GetNextHostSuccess, [352](#)
 - NewL, [352](#)
 - Open, [352](#)
 - OscDNSRequest, [352](#)
 - OscGetHostByNameRequest, [352](#)
- OscDNSIBase, [353](#)
 - OscDNSIBase, [354](#)
- OscDNSIBase
 - ~OscDNSIBase, [354](#)
 - CancelFxn, [354](#)
 - CancelGetHostByName, [354](#)
 - Close, [354](#)
 - GetHostByName, [354](#)
 - GetHostByNameResponseContainsAlias-Info, [354](#)
 - GetHostByNameSuccess, [354](#)
 - GetNextHost, [354](#)
 - GetNextHostSuccess, [354](#)
 - iAlloc, [355](#)
 - iSocketServ, [355](#)
 - IsReady, [354](#)
 - Open, [354](#)
 - OscDNSIBase, [354](#)
 - OscDNSRequest, [355](#)
 - OscGetHostByNameRequest, [355](#)
- OscDNSMethod, [356](#)
 - OscDNSMethod, [357](#)
 - OscDNSRequestAO, [363](#)
- OscDNSMethod
 - Abort, [357](#)
 - AbortAll, [357](#)
 - CancelMethod, [357](#)
 - ConstructL, [357](#)
 - iAlloc, [358](#)
 - iDNSFxn, [358](#)
 - iDNSObserver, [358](#)
 - iDNSRequestAO, [358](#)
 - iId, [358](#)
 - iLogger, [358](#)
 - MethodDone, [357](#)
 - OscDNSMethod, [357](#)
 - Run, [357](#)
 - StartMethod, [357](#)
- OscDNSObserver, [359](#)
- OscDNSObserver
 - ~OscDNSObserver, [359](#)
 - HandleDNSEvent, [359](#)
- OscDNSRequest, [360](#)
 - OscDNSI, [352](#)
 - OscDNSIBase, [355](#)
 - OscDNSRequest, [360](#)
 - OscDNSRequestAO, [363](#)
- OscDNSRequest
 - ~OscDNSRequest, [360](#)
 - Activate, [360](#)
 - CancelRequest, [360](#)
 - Complete, [360](#)
 - iActive, [360](#)
 - iDNSRequestAO, [360](#)
 - iDNSRequestParam, [360](#)
 - OscDNSRequest, [360](#)
- OscDNSRequestAO, [361](#)
 - OscDNS, [350](#)
 - OscDNSRequestAO, [362](#)
- OscDNSRequestAO
 - Abort, [362](#)
 - Cancelled, [362](#)
 - ConstructL, [362](#)
 - DoCancel, [362](#)
 - Failure, [362](#)
 - GetHostByNameParam, [363](#)
 - GetSocketError, [362](#)
 - iDNSI, [363](#)
 - iDNSMethod, [363](#)
 - iLogger, [363](#)
 - iSocketError, [363](#)

- NewRequest, [362](#)
- OscIDNSI, [363](#)
- OscIDNSMethod, [363](#)
- OscIDNSRequest, [363](#)
- OscIDNSRequestAO, [362](#)
- RequestDone, [362](#)
- Run, [362](#)
- Serv, [363](#)
- Success, [363](#)
- OscDoubleLink, [364](#)
 - OscDoubleLink, [364](#)
- OscDoubleLink
 - iNext, [364](#)
 - InsertAfter, [364](#)
 - InsertBefore, [364](#)
 - iPrev, [364](#)
 - OscDoubleLink, [364](#)
 - Remove, [364](#)
- OscDoubleList, [365](#)
 - OscDoubleList, [365](#)
- OscDoubleList
 - Head, [365](#)
 - InsertHead, [365](#)
 - InsertTail, [365](#)
 - IsHead, [365](#)
 - IsTail, [365](#)
 - OscDoubleList, [365](#)
 - Tail, [365](#)
- OscDoubleListBase, [366](#)
 - OscDoubleListBase, [367](#)
- OscDoubleListBase
 - getHead, [367](#)
 - getOffset, [367](#)
 - iHead, [367](#)
 - Insert, [367](#)
 - InsertHead, [367](#)
 - InsertTail, [367](#)
 - iOffset, [367](#)
 - IsEmpty, [367](#)
 - OscDoubleListBase, [367](#)
 - Reset, [367](#)
 - SetOffset, [367](#)
- OscDoubleRunner, [368](#)
 - OscDoubleRunner, [368](#)
- OscDoubleRunner
 - iHead, [368](#)
 - iNext, [368](#)
 - iOffset, [368](#)
 - operator T *, [368](#)
 - operator++, [368](#)
 - operator--, [368](#)
 - OscDoubleRunner, [368](#)
 - Set, [368](#)
 - SetToHead, [368](#)
 - SetToTail, [368](#)
- OscErrAlreadyExists
 - osclerror, [91](#)
- OscErrAlreadyInstalled
 - osclerror, [91](#)
- OscErrArgument
 - osclerror, [91](#)
- OscErrBadHandle
 - osclerror, [91](#)
- OscErrBusy
 - osclerror, [91](#)
- OscErrCancelled
 - osclerror, [91](#)
- OscErrCorrupt
 - osclerror, [91](#)
- OscErrGeneral
 - osclerror, [91](#)
- OscErrInvalidState
 - osclerror, [91](#)
- OscErrNoHandler
 - osclerror, [91](#)
- OscErrNoMemory
 - osclerror, [91](#)
- OscErrNone
 - osclerror, [91](#)
- OscErrNoResources
 - osclerror, [91](#)
- OscErrNotInstalled
 - osclerror, [91](#)
- OscErrNotReady
 - osclerror, [91](#)
- OscErrNotSupported
 - osclerror, [91](#)
- OscError, [370](#)
 - OscErrorTrapImp, [376](#)
 - OscExecSchedulerCommonBase, [395](#)
 - OscTrapStack, [598](#)
- OscError
 - Leave, [370](#)
 - LeaveIfError, [370](#)
 - LeaveIfNull, [370](#)
 - Pop, [370](#)
 - PopDealloc, [370](#), [371](#)
 - PushL, [371](#)
- osclerror
 - _PV_TRAP, [87](#)
 - _PV_TRAP_NO_TLS, [87](#)
 - internalLeave, [87](#)
 - OSCL_BAD_ALLOC_EXCEPTION_CODE, [87](#)
 - OSCL_CATCH, [87](#)
 - OSCL_CATCH_ANY, [88](#)
 - OSCL_ERR_NONE, [88](#)
 - OSCL_FIRST_CATCH, [88](#)

- OSCL_FIRST_CATCH_ANY, 88
- OSCL_GetLastError, 92
- OSCL_IsErrnoSupported, 92
- OSCL_JUMP_MAX_JUMP_MARKS, 88
- OSCL_LAST_CATCH, 88
- OSCL_LEAVE, 89
- OSCL_MAX_TRAP_LEVELS, 89
- OSCL_SetLastError, 92
- OSCL_StrError, 92
- OSCL_TRAPSTACK_POP, 89
- OSCL_TRAPSTACK_POPDEALLOC, 89
- OSCL_TRAPSTACK_PUSH, 89
- OSCL_TRY, 89
- OSCL_TRY_NO_TLS, 89
- OscErrAlreadyExists, 91
- OscErrAlreadyInstalled, 91
- OscErrArgument, 91
- OscErrBadHandle, 91
- OscErrBusy, 91
- OscErrCancelled, 91
- OscErrCorrupt, 91
- OscErrGeneral, 91
- OscErrInvalidState, 91
- OscErrNoHandler, 91
- OscErrNoMemory, 91
- OscErrNone, 91
- OscErrNoResources, 91
- OscErrNotInstalled, 91
- OscErrNotReady, 91
- OscErrNotSupported, 91
- OscErrOverflow, 91
- OscErrSystemCallFailed, 91
- OscErrThreadContextIncorrect, 91
- OscErrTimeout, 91
- OscErrUnderflow, 91
- OscFailure, 91
- OscLeaveCode, 92
- OscPending, 91
- OscReturnCode, 92
- OscSuccess, 91
- OscTrapOperation, 92
- PVError_DoLeave, 91
- PVERROR_IMP_JUMPS, 91
- PVERRORTRAP_REGISTRY, 91
- PVERRORTRAP_REGISTRY_ID, 92
- OscErrorAllocator, 372
 - OscErrorAllocator, 372
- OscErrorAllocator
 - allocate, 372
 - deallocate, 372
 - operator delete, 373
 - operator new, 373
 - OscErrorAllocator, 372
- OscErrorTrap, 374
 - OscErrorTrapImp, 376
 - OscTrapStack, 598
- OscErrorTrap
 - Cleanup, 374
 - GetErrorTrapImp, 374
 - Init, 374
- OscErrorTrapImp, 375
 - OscJump, 419
 - OscTrapStack, 598
- OscErrorTrapImp
 - CPVInterfaceProxy, 376
 - iJumpData, 376
 - iLeave, 376
 - iTrapStack, 376
 - OscError, 376
 - OscErrorTrap, 376
 - OscExecScheduler, 376
 - OscExecSchedulerCommonBase, 376
 - OscJump, 376
 - OscJumpMark, 376
 - OscScheduler, 376
 - OscTrapStack, 376
 - Trap, 375
 - TrapNoTls, 375
 - UnTrap, 375
- OscErrOverflow
 - osclerror, 91
- OscErrSystemCallFailed
 - osclerror, 91
- OscErrThreadContextIncorrect
 - osclerror, 91
- OscErrTimeout
 - osclerror, 91
- OscErrUnderflow
 - osclerror, 91
- OscException, 377
 - OscException, 377
- OscException
 - getLeaveCode, 377
 - OscException, 377
- OscExclusiveArrayPtr, 378
 - OscExclusiveArrayPtr, 379
- OscExclusiveArrayPtr
 - ~OscExclusiveArrayPtr, 379
 - _Ptr, 380
 - get, 379
 - operator *, 379
 - operator->, 379
 - operator=, 379
 - OscExclusiveArrayPtr, 379
 - release, 380
 - set, 380
- OscExclusivePtr, 381
 - OscExclusivePtr, 382

- OscExclusivePtr
 - ~OscExclusivePtr, 382
 - _Ptr, 383
 - get, 382
 - operator *, 382
 - operator->, 382
 - operator=, 382
 - OscExclusivePtr, 382
 - release, 383
 - set, 383
- OscExclusivePtrA, 384
 - OscExclusivePtrA, 385
- OscExclusivePtrA
 - ~OscExclusivePtrA, 385
 - _Ptr, 386
 - get, 385
 - operator *, 385
 - operator->, 385
 - operator=, 385
 - OscExclusivePtrA, 385
 - release, 386
 - set, 386
- OscExecScheduler, 387
 - OscErrorTrapImp, 376
 - OscExecSchedulerBase, 389
 - OscExecSchedulerCommonBase, 395
 - PVActiveBase, 614
 - PVActiveStats, 615
 - PVThreadContext, 634
- OscExecScheduler
 - Current, 387
 - OscScheduler, 388
 - RegisterForCallback, 387
 - RunSchedulerNonBlocking, 387
- OscExecSchedulerBase, 389
 - PVThreadContext, 634
- OscExecSchedulerBase
 - OscCoeActiveScheduler, 389
 - OscExecScheduler, 389
 - PVActiveBase, 389
- OscExecSchedulerCommonBase, 390
 - EOtherExecStats_Last, 392
 - EOtherExecStats_NativeOS, 392
 - EOtherExecStats_QueueTime, 392
 - EOtherExecStats_ReleaseTime, 392
 - EOtherExecStats_WaitTime, 392
 - OscErrorTrapImp, 376
 - OscExecSchedulerCommonBase, 393
 - PVActiveStats, 615
 - PVThreadContext, 634
- OscExecSchedulerCommonBase
 - ~OscExecSchedulerCommonBase, 393
 - AddToExecTimerQ, 393
 - BeginScheduling, 393
 - BeginStats, 393
 - BlockingLoopL, 393
 - CallRunExec, 393
 - CleanupExecQ, 393
 - CleanupStatQ, 393
 - ConstructL, 393
 - ConstructStatQ, 393
 - EndScheduling, 393
 - EndStats, 393
 - Error, 393
 - FindPVBase, 393
 - GetId, 393
 - GetName, 393
 - GetScheduler, 393
 - iAlloc, 397
 - iBlockingMode, 397
 - iDebugLogger, 397
 - iDefAlloc, 397
 - iDelta, 397
 - iDoStop, 397
 - iDoSuspend, 397
 - iErrorTrapImp, 397
 - iExecTimerQ, 397
 - iGrandTotalTicks, 397
 - iLogger, 397
 - iLogPerfIndentStr, 397
 - iLogPerfIndentStrLen, 397
 - iLogPerfTotal, 397
 - iName, 397
 - iNativeMode, 397
 - IncLogPerf, 394
 - InitExecQ, 394
 - InstallScheduler, 394
 - iNumAOAdded, 397
 - iOtherExecStats, 397
 - iPVStatQ, 397
 - iPVStats, 397
 - iReadyQ, 397
 - iResumeSem, 397
 - IsInstalled, 394
 - IsStarted, 394
 - iStopper, 397
 - iStopperCrit, 397
 - iSuspended, 397
 - iThreadContext, 397
 - iTime, 397
 - iTimeCompareThreshold, 397
 - iTotalPercent, 397
 - iTotalTicksTemp, 397
 - OscActiveObject, 395
 - OscCoeActiveScheduler, 395
 - OscError, 395
 - OscExecScheduler, 395
 - OscExecSchedulerCommonBase, 393

- OscReadyQ, 395
- OscScheduler, 395
- OscTimerCompare, 395
- OscTimerObject, 397
- PendComplete, 394
- PVActiveBase, 397
- PVActiveStats, 397
- PVSchedulerStopper, 397
- PVThreadContext, 397
- RequestCanceled, 394
- ResetLogPerf, 394
- ResumeScheduler, 394
- SetScheduler, 394
- ShowStats, 394
- ShowSummaryStats, 394
- StartNativeScheduler, 394
- StartScheduler, 394
- StopScheduler, 394
- SuspendScheduler, 395
- TOtherExecStats, 392
- UninstallScheduler, 395
- UpdateTimers, 395
- UpdateTimersMsec, 395
- WaitForReadyAO, 395
- OscExtractFilenameFromFullpath
 - OscFileManager, 405
- OscFailure
 - osclerror, 91
- OscFileCache, 399
 - Osc_File, 185
 - OscFileCache, 400
- OscFileCache
 - ~OscFileCache, 400
 - _fixedCaches, 400
 - _movableCache, 400
 - AddFixedCache, 400
 - Close, 400
 - EndOfFile, 400
 - FileSize, 400
 - Flush, 400
 - Open, 400
 - OscFileCache, 400
 - OscFileCacheBuffer, 400
 - Read, 400
 - Seek, 400
 - Tell, 400
 - Write, 400
- OscFileCacheBuffer, 401
 - Osc_File, 185
 - OscFileCache, 400
 - OscFileCacheBuffer, 402
- OscFileCacheBuffer
 - capacity, 402
 - Contains, 402
 - currentPos, 402
 - endPos, 402
 - filePosition, 402
 - FillFromFile, 402
 - iContainer, 402
 - isFixed, 402
 - IsUpdated, 402
 - OscFileCacheBuffer, 402
 - pBuffer, 402
 - Preceeds, 402
 - PrepRead, 402
 - PrepWrite, 402
 - SetPosition, 402
 - updateEnd, 402
 - updateStart, 402
 - usableSize, 402
 - WriteUpdatesToFile, 402
- OscFileHandle, 403
 - OscFileHandle, 403
- OscFileHandle
 - Handle, 403
 - Osc_File, 403
 - OscFileHandle, 403
- OscFileManager, 404
 - OSCL_FILE_ATTRIBUTE_ARCHIVE, 404
 - OSCL_FILE_ATTRIBUTE_DIRECTORY, 404
 - OSCL_FILE_ATTRIBUTE_HIDDEN, 404
 - OSCL_FILE_ATTRIBUTE_NORMAL, 404
 - OSCL_FILE_ATTRIBUTE_READONLY, 404
 - OSCL_FILE_ATTRIBUTE_SYSTEM, 404
- OscFileManager
 - OSCL_FILE_ATTRIBUTE_TYPE, 404
 - OscExtractFilenameFromFullpath, 405
 - OscGetFileAttributes, 405
 - OscGetFileCreationTime, 405, 406
 - OscGetFileLastAccessTime, 406
 - OscGetFileLastWriteTime, 407
 - OscGetFileSize, 407
- OscFileStats, 409
 - OscFileStats, 409
- OscFileStats
 - End, 409
 - Log, 409
 - LogAll, 409
 - OscFileStats, 409
 - Start, 409
- OscFileStatsItem, 410
- OscFileStatsItem
 - iOpCount, 410

- iParam, [410](#)
- iParam2, [410](#)
- iStartTick, [410](#)
- iTotalTicks, [410](#)
- OscFloat
 - osclbase, [34](#)
- OscGetAsyncSockErr
 - osclconfig_io.h, [817](#)
- OscGetDottedAddr
 - osclconfig_io.h, [817](#)
- OscGetDottedAddrVector
 - osclconfig_io.h, [818](#)
- OscGetFileAttributes
 - OscFileManager, [405](#)
- OscGetFileCreationTime
 - OscFileManager, [405](#), [406](#)
- OscGetFileLastAccessTime
 - OscFileManager, [406](#)
- OscGetFileLastWriteTime
 - OscFileManager, [407](#)
- OscGetFileSize
 - OscFileManager, [407](#)
- OscGethostbyname
 - osclconfig_io.h, [818](#)
- OscGetHostByNameMethod, [411](#)
 - OscGetHostByNameRequest, [412](#)
- OscGetHostByNameMethod
 - ~OscGetHostByNameMethod, [411](#)
 - GetHostByName, [411](#)
 - NewL, [411](#)
- OscGetHostByNameRequest, [412](#)
 - OscDNSI, [352](#)
 - OscDNSIBase, [355](#)
- OscGetHostByNameRequest
 - OscGetHostByNameMethod, [412](#)
- OscGetPeerName
 - osclconfig_io.h, [818](#)
- OscInit, [413](#)
- OscInit
 - Cleanup, [413](#)
 - Init, [413](#)
- OscInteger64Transport, [414](#)
- OscInteger64Transport
 - iHigh, [414](#)
 - iLow, [414](#)
- osclio
 - EOscFileOp_Close, [97](#)
 - EOscFileOp_EndOfFile, [97](#)
 - EOscFileOp_Flush, [97](#)
 - EOscFileOp_Last, [98](#)
 - EOscFileOp_NativeClose, [97](#)
 - EOscFileOp_NativeEndOfFile, [98](#)
 - EOscFileOp_NativeFlush, [98](#)
 - EOscFileOp_NativeOpen, [97](#)
 - EOscFileOp_NativeRead, [97](#)
 - EOscFileOp_NativeSeek, [98](#)
 - EOscFileOp_NativeSetSize, [98](#)
 - EOscFileOp_NativeSize, [98](#)
 - EOscFileOp_NativeTell, [98](#)
 - EOscFileOp_NativeWrite, [98](#)
 - EOscFileOp_Open, [97](#)
 - EOscFileOp_Read, [97](#)
 - EOscFileOp_Seek, [97](#)
 - EOscFileOp_SetSize, [97](#)
 - EOscFileOp_Size, [97](#)
 - EOscFileOp_Tell, [97](#)
 - EOscFileOp_Write, [97](#)
 - EPVDNSCancel, [98](#)
 - EPVDNSFailure, [98](#)
 - EPVDNSGetHostByName, [98](#)
 - EPVDNSPending, [98](#)
 - EPVDNSSuccess, [98](#)
 - EPVDNSTimeout, [98](#)
 - oscl_chdir, [98](#)
 - OSCL_FILE_CHAR_PATH_-
DELIMITER, [96](#)
 - OSCL_FILE_STATS_LOGGER_NODE,
[96](#)
 - OSCL_FILE_WCHAR_PATH_-
DELIMITER, [96](#)
 - OSCL_FILEMGMT_E_ALREADY_-
EXISTS, [97](#)
 - OSCL_FILEMGMT_E_NO_MATCH, [97](#)
 - OSCL_FILEMGMT_E_NOT_EMPTY, [97](#)
 - OSCL_FILEMGMT_E_NOT_-
IMPLEMENTED, [97](#)
 - OSCL_FILEMGMT_E_OK, [97](#)
 - OSCL_FILEMGMT_E_PATH_NOT_-
FOUND, [97](#)
 - OSCL_FILEMGMT_E_PATH_TOO_-
LONG, [97](#)
 - OSCL_FILEMGMT_E_PERMISSION_-
DENIED, [97](#)
 - OSCL_FILEMGMT_E_SYS_SPECIFIC,
[97](#)
 - OSCL_FILEMGMT_E_UNKNOWN, [97](#)
 - OSCL_FILEMGMT_ERR_TYPE, [97](#)
 - OSCL_FILEMGMT_MODE_DIR, [97](#)
 - OSCL_FILEMGMT_MODES, [97](#)
 - OSCL_FILEMGMT_PERMS, [97](#)
 - OSCL_FILEMGMT_PERMS_EXECUTE,
[97](#)
 - OSCL_FILEMGMT_PERMS_READ, [97](#)
 - OSCL_FILEMGMT_PERMS_WRITE, [97](#)
 - OSCL_FSSTAT, [96](#)
 - oscl_getcwd, [98](#), [99](#)
 - OSCL_IO_EXTENSION_MAXLEN, [96](#)
 - OSCL_IO_FILENAME_MAXLEN, [96](#)

- oscl_mkdir, [99](#)
- oscl_rename, [99](#), [100](#)
- oscl_rmdir, [100](#)
- oscl_stat, [100](#), [101](#)
- OSCL_STAT_BUF, [96](#)
- oscl_stats, [101](#)
- TOscIFileHandle, [96](#)
- TOscIFileOffsetInt32, [96](#)
- TOscIFileOp, [97](#)
- TPVDNSEvent, [98](#)
- TPVDNSFxn, [98](#)
- OscIpmReq, [415](#)
 - OscIpmReq, [415](#)
- OscIpmReq
 - interfaceAddr, [415](#)
 - multicastAddr, [415](#)
 - OscIpmReq, [415](#)
- OscIPSocketI, [416](#)
 - OscIPSocketI, [417](#)
- OscIPSocketI
 - ~OscIPSocketI, [417](#)
 - Alloc, [417](#)
 - Bind, [417](#)
 - Close, [417](#)
 - ConstructL, [417](#)
 - GetPeerName, [417](#)
 - GetRecvData, [417](#)
 - GetSendData, [417](#)
 - iAddress, [418](#)
 - iAlloc, [418](#)
 - iId, [418](#)
 - iLogger, [418](#)
 - iObserver, [418](#)
 - iSocket, [418](#)
 - iSocketServ, [418](#)
 - Join, [417](#)
 - OscIPSocketI, [417](#)
 - OscIsocketMethod, [418](#)
 - OscIsocketRequestAO, [418](#)
 - SetOptionToReuseAddress, [417](#)
 - SetRecvBufferSize, [417](#)
 - SetTOS, [417](#)
 - SocketServ, [417](#)
 - ThreadLogoff, [417](#)
 - ThreadLogon, [417](#)
- OscIJoin
 - osclconfig_io.h, [818](#)
- OscIJump, [419](#)
 - OscIErrorTrapImp, [376](#)
- OscIJump
 - ~OscIJump, [419](#)
 - Jump, [419](#)
 - OscIErrorTrapImp, [419](#)
 - StaticJump, [419](#)
 - Top, [419](#)
- OscIJumpMark
 - OscIErrorTrapImp, [376](#)
- OscILeaveCode
 - osclerror, [92](#)
- OscIListen
 - osclconfig_io.h, [819](#)
- OscIListenMethod, [420](#)
- OscIListenMethod
 - ~OscIListenMethod, [420](#)
 - Listen, [420](#)
 - ListenRequest, [420](#)
 - NewL, [420](#)
- OscIListenRequest, [421](#)
 - OscIListenRequest, [421](#)
- OscIListenRequest
 - Listen, [421](#)
 - OscIListenRequest, [421](#)
- OscILockBase, [422](#)
- OscILockBase
 - ~OscILockBase, [422](#)
 - Lock, [422](#)
 - Unlock, [422](#)
- OscIMakeInAddr
 - osclconfig_io.h, [819](#)
- OscIMakeSockAddr
 - osclconfig_io.h, [819](#)
- OscIMem, [423](#)
 - OscIMemGlobalAuditObject, [439](#)
- OscIMem
 - Cleanup, [423](#)
 - Init, [423](#)
- OscIMemAllocator, [424](#)
- OscIMemAllocator
 - allocate, [424](#)
 - allocate_fl, [424](#)
 - deallocate, [424](#)
- OscIMemAllocDestructDealloc, [425](#)
- OscIMemAllocDestructDealloc
 - allocate, [425](#)
 - allocate_fl, [425](#)
 - deallocate, [425](#)
 - destruct_and_dealloc, [425](#)
- OscIMemAudit, [427](#)
 - OscIMemAudit, [427](#)
- OscIMemAudit
 - ~OscIMemAudit, [427](#)
 - GetLock, [428](#)
 - MM_AddTag, [428](#)
 - MM_allocate, [428](#)
 - MM_CreateAllocNodeInfo, [428](#)
 - MM_deallocate, [428](#)
 - MM_GetAllocNo, [428](#)
 - MM_GetAllocNodeInfo, [428](#)

- MM_GetExistingTag, 429
- MM_GetMode, 429
- MM_GetNumAllocNodes, 429
- MM_GetOverheadStats, 429
- MM_GetPostfillPattern, 429
- MM_GetPrefillPattern, 429
- MM_GetRefCount, 429
- MM_GetRootNode, 430
- MM_GetStats, 430
- MM_GetStatsInDepth, 430
- MM_GetTagNode, 430
- MM_GetTreeNodees, 430
- MM_ReleaseAllocNodeInfo, 430
- MM_SetFailurePoint, 430
- MM_SetMode, 431
- MM_SetPostfillPattern, 431
- MM_SetPrefillPattern, 431
- MM_SetTagLevel, 431
- MM_UnsetFailurePoint, 431
- MM_Validate, 431
- OscMemAudit, 427
- OscMemGlobalAuditObject, 432
- OSCLMemAutoPtr, 433
 - OSCLMemAutoPtr, 434
- OSCLMemAutoPtr
 - ~OSCLMemAutoPtr, 434
 - _Ownership, 436
 - allocate, 435
 - deallocate, 435
 - get, 435
 - operator *, 435
 - operator->, 435
 - operator=, 435
 - OSCLMemAutoPtr, 434
 - release, 435
 - setWithoutOwnership, 435
 - takeOwnership, 436
- OscMemBasicAllocator, 437
- OscMemBasicAllocator
 - allocate, 437
 - deallocate, 437
- OscMemBasicAllocDestructDealloc, 438
- OscMemBasicAllocDestructDealloc
 - allocate, 438
 - deallocate, 438
 - destruct_and_dealloc, 438
- OscMemGlobalAuditObject, 439
 - OscMemAudit, 432
- OscMemGlobalAuditObject
 - audit_type, 439
 - getGlobalMemAuditObject, 439
 - OscMem, 439
- OscMemInit
 - osclmemory, 61
- osclmemory
 - _OSCL_CLEANUP_BASE_CLASS, 49
 - _OSCL_TRAP_NEW, 49
 - _oscl_audit_calloc, 58
 - _oscl_audit_free, 58
 - _oscl_audit_malloc, 58
 - _oscl_audit_new, 58
 - _oscl_audit_realloc, 59
 - _oscl_calloc, 59
 - _oscl_default_audit_calloc, 59
 - _oscl_default_audit_malloc, 59
 - _oscl_default_audit_new, 59
 - _oscl_default_audit_realloc, 59
 - _oscl_free, 59
 - _oscl_malloc, 59
 - _oscl_realloc, 59
 - ALLOC_NODE_FLAG, 61
 - COMPUTE_MEM_ALIGN_SIZE, 50
 - DEFAULT_MM_AUDIT_MODE, 51
 - DEFAULT_POSTFILL_PATTERN, 51
 - DEFAULT_PREFILL_PATTERN, 51
 - FENCE_PATTERN, 51
 - MEM_ALIGN_SIZE, 51
 - MIN_FENCE_SIZE, 51
 - MM_ALLOC_MAX_QUERY_-
FILENAME_LEN, 51
 - MM_ALLOC_MAX_QUERY_TAG_LEN,
51
 - MM_AllocNodeAutoPtr, 58
 - MM_AUDIT_ALLOC_NODE_-
ENABLE_FLAG, 51
 - MM_AUDIT_ALLOC_NODE_-
SUPPORT, 51
 - MM_AUDIT_FAILURE_SIMULATION_-
SUPPORT, 51
 - MM_AUDIT_FENCE_SUPPORT, 51
 - MM_AUDIT_FILL_SUPPORT, 51
 - MM_AUDIT_INCLUDE_ALL_HEAP_-
VALIDATION, 51
 - MM_AUDIT_POSTFILL_FLAG, 51
 - MM_AUDIT_PREFILL_FLAG, 51
 - MM_AUDIT_SUPPRESS_FILENAME_-
FLAG, 51
 - MM_AUDIT_VALIDATE_ALL_HEAP_-
FLAG, 51
 - MM_AUDIT_VALIDATE_BLOCK, 51
 - MM_AUDIT_VALIDATE_ON_FREE_-
FLAG, 51
 - MM_StatsNodeTagTreeType, 58
 - MMAuditCharAutoPtr, 58
 - MMAuditUint8AutoPtr, 58
 - operator delete, 59
 - operator delete[], 59
 - operator new, 59

- operator new[], 59
- OSCL_ALLOC_DELETE, 51
- OSCL_ALLOC_NEW, 52
- OSCL_ARRAY_DELETE, 52
- OSCL_ARRAY_NEW, 52
- OSCL_AUDIT_ARRAY_NEW, 52
- OSCL_AUDIT_CALLOC, 53
- OSCL_AUDIT_MALLOC, 53
- OSCL_AUDIT_NEW, 53
- OSCL_AUDIT_REALLOC, 54
- OSCL_CALLOC, 54
- oscl_calloc, 54
- OSCL_CLEANUP_BASE_CLASS, 54
- OSCL_DEFAULT_FREE, 55
- OSCL_DEFAULT_MALLOC, 55
- OSCL_DELETE, 55
- OSCL_DISABLE_WARNING_-
RETURN_TYPE_NOT_UDT, 55
- OSCL_DISABLE_WARNING_-
TRUNCATE_DEBUG_MESSAGE,
55
- OSCL_FREE, 55
- oscl_free, 55
- OSCL_HAS_GLOBAL_NEW_DELETE,
55
- OSCL_MALLOC, 56
- oscl_malloc, 56
- oscl_mem_aligned_size, 59
- oscl_memcmp, 60
- oscl_memcpy, 60
- oscl_membmove, 60
- oscl_membmove32, 60
- oscl_memset, 61
- OSCL_NEW, 56
- OSCL_PLACEMENT_NEW, 56
- OSCL_REALLOC, 56
- oscl_realloc, 56
- OSCL_TRAP_ALLOC_NEW, 56
- OSCL_TRAP_AUDIT_NEW, 57
- OSCL_TRAP_NEW, 57
- OscMemInit, 61
- OscMemStatsNodeAutoPtr, 58
- OscTagTreeType, 58
- TagTree_Allocator, 58
- OscMemoryFragment, 440
- OscMemoryFragment
 - len, 440
 - ptr, 440
- OscMemPoolAllocator, 441
 - OscMemPoolAllocator, 441
- OscMemPoolAllocator
 - ~OscMemPoolAllocator, 441
 - CreateMemPool, 441
 - DestroyMemPool, 441
 - oscl_mem_aligned_size, 441
 - OscMemPoolAllocator, 441
 - OscMemPoolFixedChunkAllocator, 442
 - OscMemPoolFixedChunkAllocator, 443
 - OscMemPoolFixedChunkAllocator
 - ~OscMemPoolFixedChunkAllocator, 443
 - addRef, 443
 - allocate, 443
 - CancelFreeChunkAvailableCallback, 443
 - createmempool, 443
 - deallocate, 444
 - destroymempool, 444
 - enablenullpointerreturn, 444
 - iCheckNextAvailableFreeChunk, 445
 - iChunkAlignment, 445
 - iChunkSize, 445
 - iChunkSizeMemAligned, 445
 - iEnableNullPtrReturn, 445
 - iFreeMemChunkList, 445
 - iMemPool, 445
 - iMemPoolAligned, 445
 - iMemPoolAllocator, 445
 - iNextAvailableContextData, 445
 - iNumChunk, 445
 - iObserver, 445
 - iRefCount, 445
 - notifyfreechunkavailable, 444
 - OscMemPoolFixedChunkAllocator, 443
 - removeRef, 444
 - OscMemPoolFixedChunkAllocatorObserver, 446
 - OscMemPoolFixedChunkAllocatorObserver
 - ~OscMemPoolFixedChunkAllocatorObserver, 446
 - freechunkavailable, 446
 - OscMemPoolResizableAllocator, 447
 - OscMemPoolResizableAllocator, 448
 - OscMemPoolResizableAllocator
 - ~OscMemPoolResizableAllocator, 448
 - addnewmempoolbuffer, 448
 - addRef, 448
 - allocate, 449
 - allocateblock, 449
 - CancelFreeChunkAvailableCallback, 449
 - CancelFreeMemoryAvailableCallback, 449
 - deallocate, 449
 - deallocateblock, 449
 - destroyallmempoolbuffers, 449
 - enablenullpointerreturn, 449
 - findfreeblock, 450
 - getAllocatedSize, 450
 - getAvailableSize, 450
 - getBufferSize, 450
 - getLargestContiguousFreeBlockSize, 450

- getMemPoolBufferAllocatedSize, [450](#)
- getMemPoolBufferSize, [450](#)
- iBlockInfoAlignedSize, [452](#)
- iBufferInfoAlignedSize, [452](#)
- iCheckFreeMemoryAvailable, [452](#)
- iCheckNextAvailable, [452](#)
- iEnableNullPtrReturn, [452](#)
- iExpectedNumBlocksPerBuffer, [452](#)
- iFreeMemContextData, [452](#)
- iFreeMemPoolObserver, [452](#)
- iMaxNewMemPoolBufferSz, [452](#)
- iMemPoolBufferAllocator, [452](#)
- iMemPoolBufferList, [452](#)
- iMemPoolBufferNumLimit, [452](#)
- iMemPoolBufferSize, [452](#)
- iNextAvailableContextData, [452](#)
- iObserver, [452](#)
- iRefCount, [452](#)
- iRequestedAvailableFreeMemSize, [452](#)
- iRequestedNextAvailableSize, [452](#)
- memoryPoolBufferMgmtOverhead, [450](#)
- notifyfreeblockavailable, [450](#)
- notifyfreememoryavailable, [450](#)
- OscMemPoolResizableAllocator, [448](#)
- removeRef, [451](#)
- setMaxSzForNewMemPoolBuffer, [451](#)
- trim, [451](#)
- validateblock, [451](#)
- OscMemPoolResizableAllocator::MemPoolBlockInfo, [453](#)
- OscMemPoolResizableAllocator::MemPool-BlockInfo
 - iBlockBuffer, [453](#)
 - iBlockPostFence, [453](#)
 - iBlockPreFence, [453](#)
 - iBlockSize, [453](#)
 - iNextFreeBlock, [453](#)
 - iParentBuffer, [453](#)
 - iPrevFreeBlock, [453](#)
- OscMemPoolResizableAllocator::MemPoolBufferInfo, [454](#)
- OscMemPoolResizableAllocator::MemPool-BufferInfo
 - iAllocatedSz, [454](#)
 - iBufferPostFence, [454](#)
 - iBufferPreFence, [454](#)
 - iBufferSize, [454](#)
 - iEndAddr, [454](#)
 - iNextFreeBlock, [454](#)
 - iNumOutstanding, [454](#)
 - iStartAddr, [454](#)
- OscMemPoolResizableAllocatorMemoryObserver, [455](#)
- OscMemPoolResizableAllocatorMemory-Observer
 - ~OscMemPoolResizableAllocatorMemoryObserver, [455](#)
 - freememoryavailable, [455](#)
- OscMemPoolResizableAllocatorObserver, [456](#)
- OscMemPoolResizableAllocatorObserver
 - ~OscMemPoolResizableAllocatorObserver, [456](#)
 - freeblockavailable, [456](#)
- OscMemStatsNode, [457](#)
- OscMemStatsNode
 - OscMemStatsNode, [457](#)
- OscMemStatsNode
 - ~OscMemStatsNode, [457](#)
 - operator delete, [457](#)
 - operator new, [457](#)
 - OscMemStatsNode, [457](#)
 - pMMFIPParam, [457](#)
 - pMMStats, [457](#)
 - reset, [457](#)
 - tag, [457](#)
- OscMemStatsNodeAutoPtr
 - osclmemory, [58](#)
- OscMutex, [458](#)
- OscMutex
 - OscMutex, [458](#)
- OscMutex
 - ~OscMutex, [458](#)
 - Close, [458](#)
 - Create, [458](#)
 - Lock, [459](#)
 - OscMutex, [458](#)
 - TryLock, [459](#)
 - Unlock, [459](#)
- OscNameString, [460](#)
- OscNameString
 - OscNameString, [460](#)
- OscNameString
 - MaxLen, [460](#)
 - OscNameString, [460](#)
 - Set, [460](#)
 - Str, [460](#)
- OscNativeFile, [461](#)
- OscNativeFile
 - Osc_FileServer, [193](#)
 - OscNativeFile, [462](#)
- OscNativeFile
 - ~OscNativeFile, [462](#)
 - Close, [462](#)
 - EndOfFile, [462](#)
 - Flush, [462](#)
 - GetError, [462](#)
 - GetReadAsyncNumElements, [462](#)
 - HasAsyncRead, [462](#)
 - Mode, [462](#)
 - Open, [462](#)
 - OscNativeFile, [462](#)

- Read, 462
- ReadAsync, 462
- ReadAsyncCancel, 462
- Seek, 463
- SetSize, 463
- Size, 463
- Tell, 463
- Write, 463
- OscNativeFileParams, 464
 - OscNativeFileParams, 464
- OscNativeFileParams
 - iAsyncReadBufferSize, 464
 - iNativeAccessMode, 464
 - iNativeBufferSize, 464
 - OscNativeFileParams, 464
- OscNetworkAddress, 465
 - OscNetworkAddress, 465
- OscNetworkAddress
 - ipAddr, 465
 - operator==, 465
 - OscNetworkAddress, 465
 - port, 465
- OscNoYieldMutex
 - oscl_mutex.h, 724
- OscNullLock, 466
- OscNullLock
 - ~OscNullLock, 466
 - Lock, 466
 - Unlock, 466
- OscPending
 - osclerror, 91
- OscPipe
 - osclconfig_io.h, 819
- OscPriorityLink, 467
- OscPriorityLink
 - iPriority, 467
- OscPriorityList, 468
 - OscPriorityList, 468
- OscPriorityList
 - Head, 468
 - Insert, 468
 - IsHead, 468
 - IsTail, 468
 - OscPriorityList, 468
 - Tail, 468
- OscPriorityQueue, 469
 - OscPriorityQueue, 470
- OscPriorityQueue
 - ~OscPriorityQueue, 470
 - c, 472
 - comp, 472
 - compare_EQ, 470
 - compare_LT, 470
 - const_reference, 470
 - container_type, 470
 - empty, 471
 - find_heap, 471
 - iterator, 470
 - oscl_priqueue_test, 472
 - OscPriorityQueue, 470
 - pop, 471
 - pop_heap, 471
 - push, 471
 - push_heap, 471
 - remove, 471
 - reserve, 471
 - size, 471
 - swap, 471
 - top, 471
 - validate, 472
 - value_type, 470
 - vec, 472
- OscPriorityQueueBase, 473
 - Osc_Vector_Base, 291
- OscPriorityQueueBase
 - ~OscPriorityQueueBase, 473
 - construct, 473
 - find_heap, 473
 - pop_heap, 473
 - push_heap, 473
 - remove, 473
- osclproc
 - EPVThreadContext_InThread, 105
 - EPVThreadContext_NonOscThread, 105
 - EPVThreadContext_OscThread, 105
 - EPVThreadContext_Undetermined, 105
 - OSCL_PERF_SUMMARY_LOGGING, 104
 - OSCL_REQUEST_ERR_CANCEL, 105
 - OSCL_REQUEST_ERR_GENERAL, 105
 - OSCL_REQUEST_ERR_NONE, 105
 - OSCL_REQUEST_PENDING, 105
 - OSCL_ZEROIZE, 104
 - OscPtrAdd, 105
 - OscPtrSub, 105
 - PV_SCHED_CHECK_Q, 104
 - PV_SCHED_ENABLE_AO_STATS, 104
 - PV_SCHED_ENABLE_LOOP_STATS, 104
 - PV_SCHED_ENABLE_PERF_LOGGING, 104
 - PV_SCHED_ENABLE_THREAD_CONTEXT_CHECKS, 104
 - PV_SCHED_FAIR_SCHEDULING, 104
 - PV_SCHED_LOG_Q, 104
 - PVEXECNAMELEN, 104
 - PVSCHEDNAMELEN, 104
 - QUE_ITER_BEGIN, 104

- QUE_ITER_END, 104
- TOscIReady, 105
- TPVThreadContext, 105
- OscIProcStatus, 474
 - ALREADY_SUSPENDED_ERROR, 474
 - BAD_THREADID_ADDR_ERROR, 474
 - EXCEED_MAX_COUNT_VARIABLE_ERROR, 475
 - EXCEED_MAX_SEM_COUNT_ERROR, 475
 - INVALID_ACCESS_ERROR, 475
 - INVALID_ARGUMENT_ERROR, 475
 - INVALID_FUNCTION_ERROR, 475
 - INVALID_HANDLE_ERROR, 475
 - INVALID_OPERATION_ERROR, 475
 - INVALID_PARAM_ERROR, 474
 - INVALID_POINTER_ERROR, 475
 - INVALID_PRIORITY_ERROR, 474
 - INVALID_THREAD_ERROR, 474
 - INVALID_THREAD_ID_ERROR, 474
 - MAX_THRDS_REACHED_ERROR, 474
 - MUTEX_LOCKED_ERROR, 475
 - NO_PERMISSION_ERROR, 474
 - NOT_ENOUGH_MEMORY_ERROR, 474
 - NOT_ENOUGH_RESOURCES_ERROR, 474
 - NOT_IMPLEMENTED, 475
 - NOT_SUSPENDED_ERROR, 474
 - OTHER_ERROR, 474
 - OUTOFMEMORY_ERROR, 474
 - PSHARED_ATTRIBUTE_SETTING_ERROR, 475
 - PSHARED_NOT_ZERO_ERROR, 475
 - RELOCK_MUTEX_ERROR, 475
 - SEM_NOT_SIGNALED_ERROR, 475
 - SUCCESS_ERROR, 474
 - SYSTEM_RESOURCES_UNAVAILABLE_ERROR, 475
 - THREAD_1_INACTIVE_ERROR, 474
 - THREAD_BLOCK_ERROR, 475
 - THREAD_NOT_OWN_MUTEX_ERROR, 475
 - TOO_MANY_THREADS_ERROR, 474
 - WAIT_ABANDONED_ERROR, 475
 - WAIT_TIMEOUT_ERROR, 475
- OscIProcStatus
 - eOscIProcError, 474
- OscIPtr, 476
 - OscIPtr, 476
- OscIPtr
 - Append, 476
 - Length, 476
 - OscIPtr, 476
 - Ptr, 476
 - Set, 476
 - SetLength, 476
 - Zero, 476
- OscIPtrAdd
 - osclproc, 105
- OscIPtrC, 478
 - OscIPtrC, 479
- OscIPtrC
 - Left, 479
 - Length, 479
 - OscIPtrC, 479
 - Ptr, 479
 - Right, 479
 - Set, 479
 - SetLength, 479
 - Zero, 479
- OscIPtrSub
 - osclproc, 105
- OscIRand, 480
- OscIRand
 - Rand, 480
 - Seed, 480
- OscIReadFD
 - osclconfig_io.h, 819
- OscIReadyAlloc, 481
- OscIReadyAlloc
 - allocate, 481
 - allocate_fl, 481
 - deallocate, 481
- OscIReadyCompare, 482
 - PVActiveBase, 614
- OscIReadyCompare
 - compare, 482
- OscIReadyQ, 483
 - OscIExecSchedulerCommonBase, 395
 - PVActiveBase, 614
 - PVActiveStats, 615
- OscIReadyQ
 - Callback, 484
 - Construct, 484
 - Depth, 484
 - IsIn, 484
 - PendComplete, 484
 - PopTop, 484
 - RegisterForCallback, 484
 - Remove, 484
 - ThreadLogoff, 484
 - ThreadLogon, 484
 - TimerCallback, 484
 - Top, 484
 - WaitAndPopTop, 484
 - WaitForRequestComplete, 484
- OscIReadySetPosition
 - PVActiveBase, 614

- OscRecv
 - osclconfig_io.h, 819
- OscRecvFrom
 - osclconfig_io.h, 819
- OscRecvFromMethod, 485
- OscRecvFromMethod
 - ~OscRecvFromMethod, 485
 - GetRecvData, 485
 - NewL, 485
 - RecvFrom, 485
 - RecvFromRequest, 485
- OscRecvFromRequest, 487
 - OscRecvFromRequest, 487
 - OscSocketI, 538
- OscRecvFromRequest
 - GetRecvData, 487
 - OscRecvFromRequest, 487
 - RecvFrom, 487
 - Success, 487
- OscRecvMethod, 489
- OscRecvMethod
 - ~OscRecvMethod, 489
 - GetRecvData, 489
 - NewL, 489
 - Recv, 489
 - RecvRequest, 489
- OscRecvRequest, 490
 - OscRecvRequest, 490
 - OscSocketI, 538
- OscRecvRequest
 - GetRecvData, 490
 - OscRecvRequest, 490
 - Recv, 490
 - Success, 490
- OscRefCounter, 491
- OscRefCounter
 - ~OscRefCounter, 491
 - addRef, 491
 - getCount, 491
 - removeRef, 491
- OscRefCounterDA, 493
 - OscRefCounterDA, 493
- OscRefCounterDA
 - ~OscRefCounterDA, 493
 - addRef, 494
 - getCount, 494
 - OscRefCounterDA, 493
 - removeRef, 494
- OscRefCounterMemFrag, 495
 - OscRefCounterMemFrag, 495
- OscRefCounterMemFrag
 - ~OscRefCounterMemFrag, 495
 - getCapacity, 496
 - getCount, 496
 - getMemFrag, 496
 - getMemFragPtr, 496
 - getMemFragSize, 496
 - getRefCounter, 496
 - operator=, 496
 - OscRefCounterMemFrag, 495
- OscRefCounterMTDA, 497
 - OscRefCounterMTDA, 497
- OscRefCounterMTDA
 - ~OscRefCounterMTDA, 497
 - addRef, 498
 - getCount, 498
 - OscRefCounterMTDA, 497
 - removeRef, 498
- OscRefCounterMTSA, 499
 - OscRefCounterMTSA, 499
- OscRefCounterMTSA
 - ~OscRefCounterMTSA, 499
 - addRef, 500
 - getCount, 500
 - OscRefCounterMTSA, 499
 - removeRef, 500
- OscRefCounterSA, 501
 - OscRefCounterSA, 501
- OscRefCounterSA
 - ~OscRefCounterSA, 501
 - addRef, 502
 - getCount, 502
 - OscRefCounterSA, 501
 - removeRef, 502
- OscRegistryAccessClient, 503
 - OscRegistryAccessClient, 503
 - OscRegistryClientImpl, 511
 - OscRegistryServTlsImpl, 514
- OscRegistryAccessClient
 - ~OscRegistryAccessClient, 503
 - Close, 503
 - Connect, 503
 - GetFactories, 503
 - GetFactory, 503
 - OscRegistryAccessClient, 503
- OscRegistryAccessClientImpl, 505
- OscRegistryAccessClientTlsImpl, 506
- OscRegistryAccessElement, 507
- OscRegistryAccessElement
 - iFactory, 507
 - iMimeType, 507
- OscRegistryClient, 508
 - OscRegistryClient, 508
 - OscRegistryClientImpl, 511
 - OscRegistryServTlsImpl, 514
- OscRegistryClient
 - ~OscRegistryClient, 508
 - Close, 508

- Connect, [508](#)
- OscRegistryClient, [508](#)
- Register, [508](#)
- UnRegister, [509](#)
- OscRegistryClientImpl, [510](#)
- OscRegistryClientImpl
 - Close, [511](#)
 - Connect, [511](#)
 - GetFactories, [511](#)
 - GetFactory, [511](#)
 - OscRegistryAccessClient, [511](#)
 - OscRegistryClient, [511](#)
 - Register, [511](#)
 - UnRegister, [511](#)
- OscRegistryClientTlsImpl, [512](#)
- OscRegistryServTlsImpl, [513](#)
 - OscRegistryServTlsImpl, [514](#)
- OscRegistryServTlsImpl
 - ~OscRegistryServTlsImpl, [514](#)
 - Close, [514](#)
 - Connect, [514](#)
 - GetFactories, [514](#)
 - GetFactory, [514](#)
 - OscRegistryAccessClient, [514](#)
 - OscRegistryClient, [514](#)
 - OscRegistryServTlsImpl, [514](#)
 - Register, [514](#)
 - UnRegister, [514](#)
- OscReturnCode
 - osclerror, [92](#)
- OscScheduler, [515](#)
 - OscErrorTrapImp, [376](#)
 - OscExecScheduler, [388](#)
 - OscExecSchedulerCommonBase, [395](#)
- OscScheduler
 - Cleanup, [515](#)
 - Init, [515](#)
- OscSchedulerCommonBase
 - PVActiveBase, [614](#)
- OscSchedulerObserver, [516](#)
- OscSchedulerObserver
 - ~OscSchedulerObserver, [516](#)
 - OscSchedulerReadyCallback, [516](#)
 - OscSchedulerTimerCallback, [516](#)
- OscSchedulerReadyCallback
 - OscSchedulerObserver, [516](#)
- OscSchedulerTimerCallback
 - OscSchedulerObserver, [516](#)
- OscScopedLock, [517](#)
 - OscScopedLock, [517](#)
- OscScopedLock
 - ~OscScopedLock, [517](#)
 - OscScopedLock, [517](#)
- OscSelect, [518](#)
- OscSelect, [519](#)
- OscSelect
 - iErrAlloc, [519](#)
 - iHeapCheck, [519](#)
 - iOscBase, [519](#)
 - iOscErrorTrap, [519](#)
 - iOscLogger, [519](#)
 - iOscMemory, [519](#)
 - iOscScheduler, [519](#)
 - iOutputFile, [519](#)
 - iSchedulerAlloc, [519](#)
 - iSchedulerName, [519](#)
 - iSchedulerReserve, [519](#)
 - OscSelect, [519](#)
- OscSemaphore, [520](#)
 - OscSemaphore, [520](#)
- OscSemaphore
 - ~OscSemaphore, [520](#)
 - Close, [520](#)
 - Create, [520](#)
 - OscSemaphore, [520](#)
 - Signal, [521](#)
 - TryWait, [521](#)
 - Wait, [521](#)
- OscSend
 - osclconfig_io.h, [820](#)
- OscSendMethod, [522](#)
- OscSendMethod
 - ~OscSendMethod, [522](#)
 - GetSendData, [522](#)
 - NewL, [522](#)
 - Send, [522](#)
 - SendRequest, [522](#)
- OscSendRequest, [523](#)
 - OscSendRequest, [523](#)
 - OscSocketI, [538](#)
- OscSendRequest
 - GetSendData, [523](#)
 - OscSendRequest, [523](#)
 - Send, [523](#)
 - Success, [523](#)
- OscSendTo
 - osclconfig_io.h, [820](#)
- OscSendToMethod, [524](#)
- OscSendToMethod
 - ~OscSendToMethod, [524](#)
 - GetSendData, [524](#)
 - NewL, [524](#)
 - SendTo, [524](#)
 - SendToRequest, [524](#)
- OscSendToRequest, [525](#)
 - OscSendToRequest, [525](#)
 - OscSocketI, [538](#)
- OscSendToRequest

- GetSendData, [525](#)
- OscSendToRequest, [525](#)
- SendTo, [525](#)
- Success, [525](#)
- OscSetNonBlocking
 - osclconfig_io.h, [820](#)
- OscSetRecvBufferSize
 - osclconfig_io.h, [820](#)
- OscSetSockOpt
 - osclconfig_io.h, [820](#)
- OscSharedPtr, [526](#)
 - OscSharedPtr, [527](#)
- OscSharedPtr
 - ~OscSharedPtr, [527](#)
 - get_count, [527](#)
 - GetRefCount, [527](#)
 - GetRep, [527](#)
 - operator *, [527](#)
 - operator TheClass *, [528](#)
 - operator->, [528](#)
 - operator=, [528](#)
 - OscSharedPtr, [527](#)
 - Unbind, [528](#)
- OscShutdown
 - osclconfig_io.h, [820](#)
- OscShutdownMethod, [529](#)
- OscShutdownMethod
 - ~OscShutdownMethod, [529](#)
 - NewL, [529](#)
 - Shutdown, [529](#)
 - ShutdownRequest, [529](#)
- OscShutdownRequest, [530](#)
 - OscShutdownRequest, [530](#)
 - OscSocketI, [538](#)
- OscShutdownRequest
 - OscShutdownRequest, [530](#)
 - Shutdown, [530](#)
- OscSingleton, [531](#)
 - OscSingleton, [531](#)
- OscSingleton
 - ~OscSingleton, [531](#)
 - _Ptr, [532](#)
 - operator *, [531](#)
 - operator->, [531](#)
 - OscSingleton, [531](#)
 - set, [531](#)
- OscSingletonRegistry, [533](#)
- OscSingletonRegistry
 - getInstance, [533](#)
 - lockAndGetInstance, [533](#)
 - OscBase, [533](#)
 - registerInstance, [533](#)
 - registerInstanceAndUnlock, [533](#)
- OscSocket
 - osclconfig_io.h, [821](#)
- OscSocketCleanup
 - osclconfig_io.h, [821](#)
- OscSocketI, [534](#)
 - OscSocketRequestAO, [552](#)
 - OscSocketServI, [556](#)
- OscSocketI
 - ~OscSocketI, [535](#)
 - Accept, [535](#)
 - Bind, [535](#)
 - Close, [535](#)
 - Connect, [535](#)
 - GetPeerName, [535](#)
 - Join, [536](#)
 - Listen, [536](#)
 - Logger, [536](#)
 - MakeAddr, [536](#)
 - MakeMulticastGroupInformation, [536](#)
 - NewL, [536](#)
 - Open, [536](#)
 - OscAcceptRequest, [538](#)
 - OscConnectRequest, [538](#)
 - OscRecvFromRequest, [538](#)
 - OscRecvRequest, [538](#)
 - OscSendRequest, [538](#)
 - OscSendToRequest, [538](#)
 - OscShutdownRequest, [538](#)
 - OscTCPSocket, [538](#)
 - OscUDPSocket, [538](#)
 - ProcessAccept, [536](#)
 - ProcessConnect, [537](#)
 - ProcessRecv, [537](#)
 - ProcessRecvFrom, [537](#)
 - ProcessSend, [537](#)
 - ProcessSendTo, [537](#)
 - ProcessShutdown, [537](#)
 - Recv, [537](#)
 - RecvFrom, [537](#)
 - RecvFromSuccess, [537](#)
 - RecvSuccess, [537](#)
 - Send, [537](#)
 - SendSuccess, [537](#)
 - SendTo, [537](#)
 - SendToSuccess, [537](#)
 - SetRecvBufferSize, [537](#)
 - SetSockOpt, [538](#)
 - Shutdown, [538](#)
 - Socket, [538](#)
 - ThreadLogoff, [538](#)
 - ThreadLogon, [538](#)
- OscSocketIBase, [539](#)
 - OscSocketIBase, [540](#)
- OscSocketIBase
 - ~OscSocketIBase, [540](#)

- Accept, 540
- Bind, 540
- BindAsync, 540
- CancelAccept, 541
- CancelBind, 541
- CancelConnect, 541
- CancelFxn, 541
- CancelListen, 541
- CancelRecv, 541
- CancelRecvFrom, 541
- CancelSend, 541
- CancelSendTo, 541
- CancelShutdown, 541
- Close, 541
- Connect, 541
- GetShutdown, 541
- HasAsyncBind, 541
- HasAsyncListen, 541
- iAlloc, 543
- iSocketServ, 543
- IsOpen, 541
- Join, 541
- Listen, 541
- ListenAsync, 541
- Open, 542
- OscSocketIBase, 540
- OscSocketMethod, 543
- OscSocketRequest, 543
- OscSocketRequestAO, 543
- OscTCPSocket, 543
- OscUDPSocket, 543
- Recv, 542
- RecvFrom, 542
- RecvFromSuccess, 542
- RecvSuccess, 542
- Send, 542
- SendSuccess, 542
- SendTo, 542
- SendToSuccess, 542
- Shutdown, 543
- OscSocketMethod, 544
 - OscIPSocketI, 418
 - OscSocketIBase, 543
 - OscSocketMethod, 545
 - OscSocketRequestAO, 552
- OscSocketMethod
 - ~OscSocketMethod, 545
 - Abort, 545
 - AbortAll, 545
 - Alloc, 545
 - CancelMethod, 545
 - ConstructL, 545
 - iContainer, 546
 - iSocketFxn, 546
 - iSocketRequestAO, 546
 - MethodDone, 545
 - OscSocketMethod, 545
 - Run, 545
 - StartMethod, 546
 - ThreadLogoff, 546
 - ThreadLogon, 546
- OscSocketObserver, 547
- OscSocketObserver
 - ~OscSocketObserver, 547
 - HandleSocketEvent, 547
- OscSocketRequest, 548
 - OscSocketIBase, 543
 - OscSocketRequest, 548
 - OscSocketRequestAO, 552
 - OscSocketServI, 556
- OscSocketRequest
 - Activate, 548
 - CancelRequest, 548
 - Complete, 548
 - Fxn, 548
 - iParam, 548
 - iSocketI, 548
 - iSocketRequestAO, 548
 - OscSocketRequest, 548
- OscSocketRequestAO, 549
 - OscIPSocketI, 418
 - OscSocketIBase, 543
 - OscSocketRequestAO, 550
- OscSocketRequestAO
 - ~OscSocketRequestAO, 550
 - Abort, 550
 - Alloc, 550
 - CleanupParam, 550
 - ConstructL, 550
 - DoCancel, 550
 - GetSocketError, 550
 - iContainer, 552
 - Id, 551
 - iParam, 552
 - iParamSize, 552
 - iSocketError, 552
 - NewRequest, 551
 - OscSocketI, 552
 - OscSocketMethod, 552
 - OscSocketRequest, 552
 - OscSocketRequestAO, 550
 - RequestDone, 551
 - Run, 551
 - SocketI, 551
 - SocketObserver, 551
 - Success, 551
- OscSocketSelect
 - osclconfig_io.h, 821

- OscSocketServ, [553](#)
 - OscSocketServI, [556](#)
- OscSocketServ
 - ~OscSocketServ, [553](#)
 - Close, [553](#)
 - Connect, [553](#)
 - NewL, [554](#)
 - OscIDNS, [554](#)
 - OscTCPSocket, [554](#)
 - OscUDPSocket, [554](#)
- OscSocketServI, [555](#)
 - OscSocketServRequestList, [559](#)
- OscSocketServI
 - Close, [555](#)
 - Connect, [555](#)
 - IsServerThread, [556](#)
 - LoopbackSocket, [556](#)
 - NewL, [556](#)
 - OscIDNSI, [556](#)
 - OscSocketI, [556](#)
 - OscSocketRequest, [556](#)
 - OscSocketServ, [556](#)
 - OscSocketServRequestList, [556](#)
 - OscTCPSocketI, [556](#)
 - OscUDPSocketI, [556](#)
- OscSocketServIBase, [557](#)
 - ESocketServ_Connected, [557](#)
 - ESocketServ_Error, [558](#)
 - ESocketServ_Idle, [557](#)
 - OscSocketServIBase, [558](#)
- OscSocketServIBase
 - ~OscSocketServIBase, [558](#)
 - Close, [558](#)
 - Connect, [558](#)
 - iAlloc, [558](#)
 - iLogger, [558](#)
 - iServerError, [558](#)
 - iServState, [558](#)
 - IsServConnected, [558](#)
 - OscSocketServIBase, [558](#)
 - State, [558](#)
 - TSocketServState, [557](#)
- OscSocketServRequestList, [559](#)
 - OscSocketServI, [556](#)
 - OscSocketServRequestList, [559](#)
- OscSocketServRequestList
 - Add, [559](#)
 - Close, [559](#)
 - Open, [559](#)
 - OscSocketServI, [559](#)
 - OscSocketServRequestList, [559](#)
 - Remove, [559](#)
 - StartCancel, [559](#)
 - WaitOnRequests, [559](#)
- Wakeup, [559](#)
- OscSocketServRequestQElem, [561](#)
 - OscSocketServRequestQElem, [561](#)
- OscSocketServRequestQElem
 - iCancel, [561](#)
 - iSelect, [561](#)
 - iSocketRequest, [561](#)
 - OscSocketServRequestQElem, [561](#)
- OscSocketStartup
 - osclconfig_io.h, [821](#)
- OscSocketTOS, [562](#)
 - EPVCritic_Ecp, [562](#)
 - EPVFlash, [562](#)
 - EPVHiRel, [562](#)
 - EPVHiThrpt, [562](#)
 - EPVImmediate, [562](#)
 - EPVInetControl, [562](#)
 - EPVLDelay, [562](#)
 - EPVNetControl, [562](#)
 - EPVNoTOS, [562](#)
 - EPVOverrideFlash, [562](#)
 - EPVPriority, [562](#)
 - EPVRoutine, [562](#)
 - OscSocketTOS, [563](#)
- OscSocketTOS
 - ClearTOS, [563](#)
 - GetTOS, [563](#)
 - OscSocketTOS, [563](#)
 - SetPrecedence, [563](#)
 - SetPriority, [563](#)
 - TPVServicePrecedence, [562](#)
 - TPVServicePriority, [562](#)
- OscSuccess
 - osclerror, [91](#)
- OscTagTreeType
 - osclmemory, [58](#)
- OscTCPSocket, [564](#)
 - OscSocketI, [538](#)
 - OscSocketIBase, [543](#)
 - OscSocketServ, [554](#)
- OscTCPSocket
 - ~OscTCPSocket, [565](#)
 - Accept, [565](#)
 - Bind, [565](#)
 - BindAsync, [565](#)
 - CancelAccept, [566](#)
 - CancelBind, [566](#)
 - CancelConnect, [566](#)
 - CancelListen, [566](#)
 - CancelRecv, [566](#)
 - CancelSend, [566](#)
 - CancelShutdown, [566](#)
 - Close, [567](#)
 - Connect, [567](#)

- GetAcceptedSocketL, [567](#)
- GetPeerName, [567](#)
- GetRecvData, [568](#)
- GetSendData, [568](#)
- Listen, [568](#)
- ListenAsync, [568](#)
- NewL, [568](#)
- Recv, [569](#)
- Send, [569](#)
- SetOptionToReuseAddress, [569](#)
- SetTOS, [570](#)
- Shutdown, [570](#)
- ThreadLogoff, [570](#)
- ThreadLogon, [570](#)
- OscITCPSocketI, [571](#)
 - OscISocketServI, [556](#)
- OscITCPSocketI
 - ~OscITCPSocketI, [572](#)
 - Accept, [572](#)
 - BindAsync, [572](#)
 - CancelAccept, [572](#)
 - CancelBind, [572](#)
 - CancelConnect, [572](#)
 - CancelListen, [572](#)
 - CancelRecv, [572](#)
 - CancelSend, [572](#)
 - CancelShutdown, [572](#)
 - Close, [572](#)
 - Connect, [572](#)
 - GetAcceptedSocketL, [572](#)
 - GetRecvData, [572](#)
 - GetSendData, [572](#)
 - Listen, [572](#)
 - ListenAsync, [573](#)
 - NewL, [573](#)
 - Recv, [573](#)
 - Send, [573](#)
 - Shutdown, [573](#)
 - ThreadLogoff, [573](#)
 - ThreadLogon, [573](#)
- OscIThread, [574](#)
 - OscIThread, [574](#)
- OscIThread
 - ~OscIThread, [574](#)
 - CompareId, [574](#)
 - Create, [575](#)
 - EnableKill, [575](#)
 - Exit, [575](#)
 - GetId, [575](#)
 - GetPriority, [576](#)
 - OscIThread, [574](#)
 - Resume, [576](#)
 - SetPriority, [576](#)
 - SleepMillisec, [576](#)
 - Suspend, [577](#)
 - Terminate, [577](#)
- OscIThread_State
 - oscl_thread.h, [789](#)
- OscIThreadLock, [578](#)
 - OscIThreadLock, [578](#)
- OscIThreadLock
 - ~OscIThreadLock, [578](#)
 - Lock, [578](#)
 - OscIThreadLock, [578](#)
 - Unlock, [578](#)
- OscIThreadPriority
 - oscl_thread.h, [789](#)
- OscITickCount, [579](#)
- OscITickCount
 - MsecToTicks, [579](#)
 - TickCount, [579](#)
 - TickCountFrequency, [579](#)
 - TickCountPeriod, [579](#)
 - TicksToMsec, [579](#)
- OSCLTICKCOUNT_MAX_TICKS
 - osclutil, [68](#)
- OscITimer, [581](#)
 - OscITimer, [582](#)
- OscITimer
 - ~OscITimer, [582](#)
 - callback_timer_type, [582](#)
 - CallbackTimer< Alloc >, [583](#)
 - Cancel, [582](#)
 - Clear, [582](#)
 - OscITimer, [582](#)
 - Request, [582](#)
 - SetExactFrequency, [582](#)
 - SetFrequency, [583](#)
 - SetObserver, [583](#)
 - TimerBaseElapsed, [583](#)
- OscITimerCompare, [584](#)
 - OscIExecSchedulerCommonBase, [395](#)
- OscITimerCompare
 - compare, [584](#)
- OscITimerObject, [585](#)
 - OscIExecSchedulerCommonBase, [397](#)
 - OscITimerObject, [586](#)
 - PVActiveBase, [614](#)
 - PVActiveStats, [615](#)
 - PVThreadContext, [634](#)
- OscITimerObject
 - ~OscITimerObject, [586](#)
 - AddToScheduler, [586](#)
 - After, [586](#)
 - Cancel, [586](#)
 - DoCancel, [586](#)
 - IsBusy, [587](#)
 - OscITimerObject, [586](#)

- Priority, [587](#)
- RemoveFromScheduler, [587](#)
- RunError, [587](#)
- RunIfNotReady, [587](#)
- SetBusy, [587](#)
- SetStatus, [587](#)
- Status, [588](#)
- StatusRef, [588](#)
- OscTimerObserver, [589](#)
- OscTimerObserver
 - ~OscTimerObserver, [589](#)
 - TimeoutOccurred, [589](#)
- OscTimerQ, [590](#)
- OscTimerQ
 - Add, [590](#)
 - Construct, [590](#)
 - IsIn, [590](#)
 - Pop, [590](#)
 - PopTop, [590](#)
 - Remove, [590](#)
 - Top, [590](#)
- OscTLS, [591](#)
 - OscTLS, [591](#)
- OscTLS
 - ~OscTLS, [591](#)
 - _Ptr, [592](#)
 - operator *, [591](#)
 - operator->, [591](#)
 - OscTLS, [591](#)
 - set, [591](#)
- OscTLSEx, [593](#)
 - OscTLSEx, [593](#)
- OscTLSEx
 - ~OscTLSEx, [593](#)
 - _Ptr, [594](#)
 - operator *, [593](#)
 - operator->, [593](#)
 - OscTLSEx, [593](#)
 - set, [593](#)
- OscTLSRegistry, [595](#)
- OscTLSRegistry
 - getInstance, [595](#)
 - OscBase, [595](#)
 - registerInstance, [595](#)
- OscTLSRegistryEx, [596](#)
- OscTLSRegistryEx
 - getInstance, [596](#)
 - registerInstance, [596](#)
- OscTrapItem, [597](#)
 - OscTrapItem, [597](#)
- OscTrapItem
 - OscTrapItem, [597](#)
 - OscTrapStack, [597](#)
 - OscTrapStackItem, [597](#)
- OscTrapOperation
 - osclerror, [92](#)
- OscTrapStack, [598](#)
 - OscErrorTrapImp, [376](#)
 - OscTrapItem, [597](#)
- OscTrapStack
 - OscError, [598](#)
 - OscErrorTrap, [598](#)
 - OscErrorTrapImp, [598](#)
- OscTrapStackItem, [599](#)
 - OscTrapItem, [597](#)
 - OscTrapStackItem, [599](#)
- OscTrapStackItem
 - iCBase, [599](#)
 - iNext, [599](#)
 - iTAny, [599](#)
 - iTrapOperation, [599](#)
 - OscTrapStackItem, [599](#)
- OscUDPSocket, [600](#)
 - OscSocketI, [538](#)
 - OscSocketIBase, [543](#)
 - OscSocketServ, [554](#)
- OscUDPSocket
 - ~OscUDPSocket, [601](#)
 - Bind, [601](#)
 - BindAsync, [601](#)
 - CancelBind, [601](#)
 - CancelRecvFrom, [601](#)
 - CancelSendTo, [601](#)
 - Close, [602](#)
 - GetPeerName, [602](#)
 - GetRecvData, [602](#)
 - GetSendData, [602](#)
 - Join, [602](#)
 - JoinMulticastGroup, [603](#)
 - NewL, [603](#)
 - RecvFrom, [603](#)
 - SendTo, [604](#)
 - SetMulticastTTL, [604](#)
 - SetOptionToReuseAddress, [604](#)
 - SetRecvBufferSize, [605](#)
 - SetTOS, [605](#)
 - ThreadLogoff, [605](#)
 - ThreadLogon, [605](#)
- OscUDPSocketI, [606](#)
 - OscSocketServI, [556](#)
- OscUDPSocketI
 - ~OscUDPSocketI, [607](#)
 - BindAsync, [607](#)
 - CancelBind, [607](#)
 - CancelRecvFrom, [607](#)
 - CancelSendTo, [607](#)
 - Close, [607](#)
 - GetRecvData, [607](#)

- GetSendData, [607](#)
- JoinMulticastGroup, [607](#)
- NewL, [607](#)
- RecvFrom, [607](#)
- SendTo, [607](#)
- SetMulticastTTL, [607](#)
- ThreadLogoff, [607](#)
- ThreadLogon, [607](#)
- OscUid32
 - oscl_uuid.h, [801](#)
- OscUnMakeInAddr
 - osclconfig_io.h, [821](#)
- OscUnMakeSockAddr
 - osclconfig_io.h, [822](#)
- osclutil
 - ~OSCL_HeapString, [83](#)
 - ~OSCL_StackString, [83](#)
 - ~OSCL_wHeapString, [83](#)
 - ~OSCL_wStackString, [83](#)
 - APPEND_MEDIA_AT_END, [83](#)
 - BufferFreeFuncPtr, [68](#)
 - EOSCL_StringOp_CompressASCII, [69](#)
 - EOSCL_StringOp_UTF16ToUTF8, [69](#)
 - EOSCL_wStringOp_ExpandASCII, [69](#)
 - EOSCL_wStringOp_UTF8ToUTF16, [69](#)
 - extract_string, [69](#)
 - get_cstr, [69](#)
 - get_maxsize, [70](#)
 - get_size, [70](#)
 - get_str, [71](#)
 - GetBufferState, [71](#)
 - GetFragment, [71](#)
 - MAX_NUMBER_OF_BYTE_PER_UTF8, [68](#)
 - MediaTimestamp, [68](#)
 - operator=, [71–73](#)
 - oscl_abs, [73](#)
 - OSCL_ASCII_CASE_MAGIC_BIT, [83](#)
 - oscl_asin, [73](#)
 - oscl_atan, [73](#)
 - oscl_cos, [73](#)
 - oscl_exp, [73](#)
 - oscl_floor, [73](#)
 - OSCL_HeapString, [73, 74](#)
 - oscl_isdigit, [68](#)
 - oscl_log, [74](#)
 - oscl_log10, [74](#)
 - oscl_pow, [74](#)
 - oscl_sin, [75](#)
 - oscl_sprintf, [75](#)
 - oscl_sqrt, [75](#)
 - OSCL_StackString, [75, 76](#)
 - oscl_str_escape_xml, [76](#)
 - oscl_str_is_valid_utf8, [76](#)
 - oscl_str_need_escape_xml, [77](#)
 - oscl_str_truncate_utf8, [77](#)
 - oscl_str_unescape_uri, [77, 78](#)
 - oscl_tan, [78](#)
 - OSCL_TStrPtrLen, [68](#)
 - oscl_UnicodeToUTF8, [78](#)
 - oscl_UTF8ToUnicode, [79](#)
 - oscl_vsnprintf, [79, 81](#)
 - OSCL_wHeapString, [81](#)
 - OSCL_wStackString, [81](#)
 - OscComponentFactory, [68](#)
 - OSCLTICKCOUNT_MAX_TICKS, [68](#)
 - PV_atof, [81](#)
 - PV_atoi, [81](#)
 - set, [81–83](#)
 - skip_to_line_term, [83](#)
 - skip_to_whitespace, [83](#)
 - skip_whitespace, [83](#)
 - skip_whitespace_and_line_term, [83](#)
 - StrCSumPtrLen, [68](#)
 - StrPtrLen, [68](#)
 - TOSCL_StringOp, [69](#)
 - TOSCL_wStringOp, [69](#)
 - WStrPtrLen, [68](#)
 - OscUuid, [609](#)
 - OscUuid, [610](#)
 - OscUuid
 - data1, [610](#)
 - data2, [610](#)
 - data3, [610](#)
 - data4, [610](#)
 - operator!=, [610](#)
 - operator=, [610](#)
 - operator==, [610](#)
 - OscUuid, [610](#)
 - OscValidInetAddr
 - osclconfig_io.h, [822](#)
 - OscWriteFD
 - osclconfig_io.h, [822](#)
 - other
 - Osc_TAlloc::rebind, [282](#)
 - other_chartype
 - OSCL_FastString, [175](#)
 - OSCL_HeapString, [196](#)
 - OSCL_HeapStringA, [198](#)
 - OSCL_StackString, [256](#)
 - OSCL_wFastString, [293](#)
 - OSCL_wHeapString, [296](#)
 - OSCL_wHeapStringA, [298](#)
 - OSCL_wStackString, [301](#)
 - OTHER_ERROR
 - OscProcStatus, [474](#)
 - OUTOFMEMORY_ERROR
 - OscProcStatus, [474](#)

- overwrite
 - CFastRep, 127
- pad
 - MM_AllocBlockFence, 146
 - MM_AllocBlockHdr, 147
- pair_citerator_citerator
 - OscI_Map, 217
- pair_iterator_bool
 - OscI_Map, 217
 - OscI_TagTree, 268
- pair_iterator_iterator
 - OscI_Map, 217
- pAllocInfo
 - MM_AllocNode, 150
- parent
 - OscI_Rb_Tree_Node_Base, 253
 - OscI_TagTree::Node, 278
- pAudit
 - OscIAuditCB, 319
- pBasePosition
 - OscIBinStream, 337
- pBuffer
 - OscIFileCacheBuffer, 402
- peakNumAllocs
 - MM_Stats_t, 164
- peakNumBytes
 - MM_Stats_t, 164
- PendComplete
 - OscIActiveObject, 310
 - OscIExecSchedulerCommonBase, 394
 - OscIReadyQ, 484
- PendForExec
 - OscIActiveObject, 310
- per_allocation_overhead
 - MM_AuditOverheadStats, 160
- perms
 - oscl_stat_buf, 257
- PersistHostAddress
 - GetHostByNameParam, 135
- pFileName
 - MM_AllocInfo, 149
- pMemBlock
 - MM_AllocInfo, 149
 - MM_AllocQueryInfo, 151
- pMMFIParam
 - OscIMemStatsNode, 457
- pMMStats
 - OscIMemStatsNode, 457
- pNext
 - MM_AllocNode, 150
- pNode
 - MM_AllocBlockHdr, 147
- pointer
 - MemAllocator, 145
 - OscI_Map, 217
 - OscI_Queue, 235
 - OscI_Rb_Tree, 242
 - OscI_Rb_Tree_Const_Iterator, 246
 - OscI_Rb_Tree_Iterator, 249
 - OscI_TagTree::const_iterator, 272
 - OscI_TagTree::iterator, 275
 - OscI_TAlloc, 280
 - OscI_Vector, 284
- Pop
 - OscIError, 370
 - OscITimerQ, 590
- pop
 - OscI_Queue, 236
 - OscI_Queue_Base, 238
 - OscIPriorityQueue, 471
- pop_back
 - OscI_Vector, 286
 - OscI_Vector_Base, 290
- pop_heap
 - OscIPriorityQueue, 471
 - OscIPriorityQueueBase, 473
- PopDealloc
 - OscIError, 370, 371
- PopTop
 - OscIReadyQ, 484
 - OscITimerQ, 590
- port
 - OscINetworkAddress, 465
- PositionInBlock
 - OscIBinStream, 336
- pPosition
 - OscIBinStream, 337
- pPrev
 - MM_AllocNode, 150
- Preceeds
 - OscIFileCacheBuffer, 402
- PrepRead
 - OscIFileCacheBuffer, 402
- PrepWrite
 - OscIFileCacheBuffer, 402
- Priority
 - OscIActiveObject, 310
 - OscITimerObject, 587
- ProcessAccept
 - OscISocketI, 536
- ProcessConnect
 - OscISocketI, 537
- ProcessRecv
 - OscISocketI, 537
- ProcessRecvFrom
 - OscISocketI, 537
- ProcessSend

- OscSocketI, [537](#)
- ProcessSendTo
 - OscSocketI, [537](#)
- ProcessShutdown
 - OscSocketI, [537](#)
- pRootNode
 - MM_AllocBlockHdr, [147](#)
- pruneSubtree
 - MM_Audit_Imp, [158](#)
- PSHARED_ATTRIBUTE_SETTING_ERROR
 - OscProcStatus, [475](#)
- PSHARED_NOT_ZERO_ERROR
 - OscProcStatus, [475](#)
- pStats
 - MM_Stats_CB, [162](#)
- pStatsNode
 - MM_AllocInfo, [149](#)
 - OscAuditCB, [319](#)
- Ptr
 - OscIPtr, [476](#)
 - OscIPtrC, [479](#)
- ptr
 - OscMemoryFragment, [440](#)
 - StrPtrLen, [647](#)
 - WStrPtrLen, [658](#)
- push
 - Osc_Queue, [236](#)
 - Osc_Queue_Base, [238](#)
 - OscPriorityQueue, [471](#)
- push_back
 - Osc_Vector, [287](#)
 - Osc_Vector_Base, [290](#)
- push_front
 - Osc_Vector, [287](#)
 - Osc_Vector_Base, [291](#)
- push_heap
 - OscPriorityQueue, [471](#)
 - OscPriorityQueueBase, [473](#)
- PushL
 - OscError, [371](#)
- PV8601TIME_BUFFER_SIZE
 - osclbase, [45](#)
- PV8601timeStrBuf
 - osclbase, [34](#)
- PV8601ToRFC822
 - osclbase, [44](#)
- PV_atof
 - osclutil, [81](#)
- PV_atoi
 - osclutil, [81](#)
- PV_CHAR_CLOSE_BRACKET
 - oscl_uuid.h, [801](#)
- PV_CHAR_COMMA
 - oscl_uuid.h, [801](#)
- PV_DNS_IS_THREAD
 - oscl_dns_tuneables.h, [676](#)
- PV_DNS_SERVER
 - oscl_dns_tuneables.h, [676](#)
- PV_DYNAMIC_LOADING_CONFIG_FILE_PATH
 - osclconfig_lib.h, [825](#)
- PV_OSCL_SOCKET_1MB_RECV_BUF
 - oscl_socket_tuneables.h, [773](#)
- PV_OSCL_SOCKET_SERVER_LOGGER_OUTPUT
 - oscl_socket_tuneables.h, [773](#)
- PV_OSCL_SOCKET_STATS_LOGGING
 - oscl_socket_tuneables.h, [773](#)
- PV_RUNTIME_LIB_FILENAME_EXTENSION
 - osclconfig_lib.h, [825](#)
- PV_SCHED_CHECK_Q
 - osclproc, [104](#)
- PV_SCHED_ENABLE_AO_STATS
 - osclproc, [104](#)
- PV_SCHED_ENABLE_LOOP_STATS
 - osclproc, [104](#)
- PV_SCHED_ENABLE_PERF_LOGGING
 - osclproc, [104](#)
- PV_SCHED_ENABLE_THREAD_CONTEXT_CHECKS
 - osclproc, [104](#)
- PV_SCHED_FAIR_SCHEDULING
 - osclproc, [104](#)
- PV_SCHED_LOG_Q
 - osclproc, [104](#)
- PV_SOCKET_REQUEST_AO_PRIORITY
 - oscl_socket_tuneables.h, [773](#)
- PV_SOCKET_SERVER
 - oscl_socket_tuneables.h, [773](#)
- PV_SOCKET_SERVER_AO_INTERVAL_MSEC
 - oscl_socket_tuneables.h, [774](#)
- PV_SOCKET_SERVER_AO_PRIORITY
 - oscl_socket_tuneables.h, [774](#)
- PV_SOCKET_SERVER_IS_THREAD
 - oscl_socket_tuneables.h, [774](#)
- PV_SOCKET_SERVER_SELECT
 - oscl_socket_tuneables.h, [774](#)
- PV_SOCKET_SERVER_SELECT_LOOPBACK_SOCKET
 - oscl_socket_tuneables.h, [774](#)
- PV_SOCKET_SERVER_SELECT_TIMEOUT_MSEC
 - oscl_socket_tuneables.h, [774](#)
- PV_SOCKET_SERVER_THREAD_PRIORITY
 - oscl_socket_tuneables.h, [774](#)

- PV_SOCKET_SERVI_STATS
 - oscl_socket_tuneables.h, 774
- PVActiveBase, 611
 - OscExecSchedulerBase, 389
 - OscExecSchedulerCommonBase, 397
 - PVActiveBase, 612
 - PVActiveStats, 615
 - PVThreadContext, 634
- PVActiveBase
 - ~PVActiveBase, 612
 - Activate, 612
 - AddToScheduler, 612
 - Cancel, 612
 - Destroy, 612
 - DoCancel, 612
 - iAddedNum, 614
 - iBusy, 614
 - iName, 614
 - iPVActiveStats, 614
 - iPVReadyQLink, 614
 - IsAdded, 612
 - IsInAnyQ, 613
 - iStatus, 614
 - iThreadContext, 614
 - OscActiveObject, 614
 - OscExecScheduler, 614
 - OscReadyCompare, 614
 - OscReadyQ, 614
 - OscReadySetPosition, 614
 - OscSchedulerCommonBase, 614
 - OscTimerObject, 614
 - PVActiveBase, 612
 - PVActiveStats, 614
 - RemoveFromScheduler, 613
 - Run, 613
 - RunError, 613
- PVActiveStats, 615
 - OscExecSchedulerCommonBase, 397
 - PVActiveBase, 614
- PVActiveStats
 - OscActiveObject, 615
 - OscExecScheduler, 615
 - OscExecSchedulerCommonBase, 615
 - OscReadyQ, 615
 - OscTimerObject, 615
 - PVActiveBase, 615
- PVCleanupStack
 - _OscHeapBase, 110
- PVError_DoLeave
 - oscl_error_imp_fatalerror.h, 684
 - oscl_error_imp_jumps.h, 686
 - osclerror, 91
- PVERROR_IMP_JUMPS
 - osclerror, 91
- PVERRORTRAP_REGISTRY
 - osclerror, 91
- PVERRORTRAP_REGISTRY_ID
 - osclerror, 92
- PVEXECNAMELEN
 - osclproc, 104
- PVLogger, 616
 - ~PVLogger, 617
 - AddAppender, 617
 - AddFilter, 617
 - alloc_type, 617
 - Cleanup, 618
 - DisableAppenderInheritance, 618
 - filter_status_type, 617
 - GetLoggerObject, 618
 - GetLogLevel, 618
 - GetNumAppenders, 618
 - GetParent, 619
 - Init, 619
 - IsActive, 619
 - log_level_type, 617
 - LogMsgBuffers, 619
 - LogMsgBuffersV, 619
 - LogMsgString, 620
 - LogMsgStringV, 620
 - message_id_type, 617
 - PVLogger, 617
 - PVLoggerRegistry, 621
 - RemoveAppender, 620
 - SetLogLevel, 620
 - SetLogLevelAndPropagate, 621
 - SetParent, 621
- pvlogger.h, 850
 - _PVLOGGER_LOGBIN, 852
 - _PVLOGGER_LOGBIN_V, 852
 - _PVLOGGER_LOGMSG, 852
 - _PVLOGGER_LOGMSG_V, 852
 - PVLOGGER_ENABLE, 852
 - PVLOGGER_INST_LEVEL, 853
 - PVLOGGER_INST_LEVEL_SUPPORT, 853
 - PVLOGGER_LEVEL_UNINITIALIZED, 856
 - PVLOGGER_LOG_USE_ONLY, 853
 - PVLOGGER_LOGBIN, 853
 - PVLOGGER_LOGBIN_PVLOGMSG_INST_HLDBG, 853
 - PVLOGGER_LOGBIN_PVLOGMSG_INST_LLDBG, 854
 - PVLOGGER_LOGBIN_PVLOGMSG_INST_MLDBG, 854
 - PVLOGGER_LOGBIN_PVLOGMSG_INST_PROF, 854

PVLOGGER_LOGBIN_PVLOGMSG_-
 INST_REL, 854
 PVLOGGER_LOGBIN_V, 854
 PVLOGGER_LOGBIN_V_-
 PVLOGMSG_INST_HLDBG, 854
 PVLOGGER_LOGBIN_V_-
 PVLOGMSG_INST_LLDBG, 854
 PVLOGGER_LOGBIN_V_-
 PVLOGMSG_INST_PROF, 854
 PVLOGGER_LOGBIN_V_-
 PVLOGMSG_INST_REL, 854
 PVLOGGER_LOGBIN_V_-
 PVLOGMSG_V_INST_MLDBG,
 854
 PVLOGGER_LOGMSG, 854
 PVLOGGER_LOGMSG_PVLOGMSG_-
 INST_HLDBG, 854
 PVLOGGER_LOGMSG_PVLOGMSG_-
 INST_LLDBG, 855
 PVLOGGER_LOGMSG_PVLOGMSG_-
 INST_MLDBG, 855
 PVLOGGER_LOGMSG_PVLOGMSG_-
 INST_PROF, 855
 PVLOGGER_LOGMSG_PVLOGMSG_-
 INST_REL, 855
 PVLOGGER_LOGMSG_V, 855
 PVLOGGER_LOGMSG_V_-
 PVLOGMSG_INST_HLDBG, 855
 PVLOGGER_LOGMSG_V_-
 PVLOGMSG_INST_LLDBG, 855
 PVLOGGER_LOGMSG_V_-
 PVLOGMSG_INST_MLDBG,
 855
 PVLOGGER_LOGMSG_V_-
 PVLOGMSG_INST_PROF, 855
 PVLOGGER_LOGMSG_V_-
 PVLOGMSG_INST_REL, 855
 PVLOGMSG_ALERT, 856
 PVLOGMSG_CRIT, 856
 PVLOGMSG_DEBUG, 856
 PVLOGMSG_EMERG, 856
 PVLOGMSG_ERR, 856
 PVLOGMSG_FATAL_ERROR, 856
 PVLOGMSG_INFO, 857
 PVLOGMSG_INST_HLDBG, 855
 PVLOGMSG_INST_LLDBG, 855
 PVLOGMSG_INST_MLDBG, 855
 PVLOGMSG_INST_PROF, 856
 PVLOGMSG_INST_REL, 856
 PVLOGMSG_NONFATAL_ERROR, 857
 PVLOGMSG_NOTICE, 857
 PVLOGMSG_STACK_TRACE, 857
 PVLOGMSG_STATISTIC, 857
 PVLOGMSG_VERBOSE, 857
 PVLOGMSG_WARNING, 857
 pvlogger_accessories.h, 858
 PVLOGGER_FILTER_ACCEPT, 858
 PVLOGGER_FILTER_NEUTRAL, 858
 PVLOGGER_FILTER_REJECT, 858
 pvlogger_c.h, 859
 PVLOGGER_C_INST_LEVEL, 860
 pvLogger_GetLoggerObject, 860
 pvLogger_IsActive, 860
 pvLogger_LogMsgString, 860
 PVLOGMSG_C_ALERT, 860
 PVLOGMSG_C_CRIT, 860
 PVLOGMSG_C_EMERG, 860
 PVLOGMSG_C_ERR, 860
 PVLOGMSG_C_INFO, 860
 PVLOGMSG_C_INST_HLDBG, 860
 PVLOGMSG_C_INST_LLDBG, 860
 PVLOGMSG_C_INST_MLDBG, 860
 PVLOGMSG_C_INST_PROF, 860
 PVLOGMSG_C_INST_REL, 860
 PVLOGMSG_C_NOTICE, 860
 PVLOGMSG_C_STACK_DEBUG, 860
 PVLOGMSG_C_STACK_TRACE, 860
 PVLOGMSG_C_WARNING, 860
 PVLOGGER_C_INST_LEVEL
 pvlogger_c.h, 860
 PVLOGGER_ENABLE
 pvlogger.h, 852
 PVLOGGER_FILTER_ACCEPT
 pvlogger_accessories.h, 858
 PVLOGGER_FILTER_NEUTRAL
 pvlogger_accessories.h, 858
 PVLOGGER_FILTER_REJECT
 pvlogger_accessories.h, 858
 pvLogger_GetLoggerObject
 pvlogger_c.h, 860
 PVLOGGER_INST_LEVEL
 osclconfig.h, 804
 pvlogger.h, 853
 PVLOGGER_INST_LEVEL_SUPPORT
 pvlogger.h, 853
 pvLogger_IsActive
 pvlogger_c.h, 860
 PVLOGGER_LEVEL_UNINITIALIZED
 pvlogger.h, 856
 PVLOGGER_LOG_USE_ONLY
 pvlogger.h, 853
 PVLOGGER_LOGBIN
 pvlogger.h, 853
 PVLOGGER_LOGBIN_PVLOGMSG_INST_-
 HLDBG
 pvlogger.h, 853
 PVLOGGER_LOGBIN_PVLOGMSG_INST_-
 LLDBG

- pvlogger.h, [854](#)
- PVLOGGER_LOGBIN_PVLOGMSG_INST_-MLDBG
 - pvlogger.h, [854](#)
- PVLOGGER_LOGBIN_PVLOGMSG_INST_-PROF
 - pvlogger.h, [854](#)
- PVLOGGER_LOGBIN_PVLOGMSG_INST_-REL
 - pvlogger.h, [854](#)
- PVLOGGER_LOGBIN_V
 - pvlogger.h, [854](#)
- PVLOGGER_LOGBIN_V_PVLOGMSG_-INST_HLDBG
 - pvlogger.h, [854](#)
- PVLOGGER_LOGBIN_V_PVLOGMSG_-INST_LLDBG
 - pvlogger.h, [854](#)
- PVLOGGER_LOGBIN_V_PVLOGMSG_-INST_PROF
 - pvlogger.h, [854](#)
- PVLOGGER_LOGBIN_V_PVLOGMSG_-INST_REL
 - pvlogger.h, [854](#)
- PVLOGGER_LOGBIN_V_PVLOGMSG_V_-INST_MLDBG
 - pvlogger.h, [854](#)
- PVLOGGER_LOGMSG
 - pvlogger.h, [854](#)
- PVLOGGER_LOGMSG_PVLOGMSG_-INST_HLDBG
 - pvlogger.h, [854](#)
- PVLOGGER_LOGMSG_PVLOGMSG_-INST_LLDBG
 - pvlogger.h, [855](#)
- PVLOGGER_LOGMSG_PVLOGMSG_-INST_MLDBG
 - pvlogger.h, [855](#)
- PVLOGGER_LOGMSG_PVLOGMSG_-INST_PROF
 - pvlogger.h, [855](#)
- PVLOGGER_LOGMSG_PVLOGMSG_-INST_REL
 - pvlogger.h, [855](#)
- PVLOGGER_LOGMSG_V
 - pvlogger.h, [855](#)
- PVLOGGER_LOGMSG_V_PVLOGMSG_-INST_HLDBG
 - pvlogger.h, [855](#)
- PVLOGGER_LOGMSG_V_PVLOGMSG_-INST_LLDBG
 - pvlogger.h, [855](#)
- PVLOGGER_LOGMSG_V_PVLOGMSG_-INST_MLDBG
 - pvlogger.h, [855](#)
- PVLOGGER_LOGMSG_V_PVLOGMSG_-INST_PROF
 - pvlogger.h, [855](#)
- PVLOGGER_LOGMSG_V_PVLOGMSG_-INST_REL
 - pvlogger.h, [855](#)
- PVLOGGER_LOGMSG_V_PVLOGMSG_-INST_MLDBG
 - pvlogger.h, [855](#)
- PVLogger_LogMsgString
 - pvlogger_c.h, [860](#)
- pvlogger_registry.h, [861](#)
- PVLoggerAppender, [622](#)
- PVLoggerAppender
 - ~PVLoggerAppender, [622](#)
 - AppendBuffers, [622](#)
 - AppendString, [622](#)
 - message_id_type, [622](#)
- PVLoggerFilter, [623](#)
- PVLoggerFilter
 - ~PVLoggerFilter, [624](#)
 - filter_status_type, [623](#)
 - FilterOpaqueMessge, [624](#)
 - FilterString, [624](#)
 - log_level_type, [623](#)
 - message_id_type, [623](#)
- PVLoggerLayout, [625](#)
- PVLoggerLayout
 - ~PVLoggerLayout, [625](#)
 - FormatOpaqueMessage, [625](#)
 - FormatString, [625](#)
 - message_id_type, [625](#)
- PVLoggerRegistry, [627](#)
- PVLogger, [621](#)
- PVLoggerRegistry, [627](#)
- PVLoggerRegistry
 - ~PVLoggerRegistry, [627](#)
 - alloc_type, [627](#)
 - CreatePVLogger, [628](#)
 - GetPVLoggerObject, [628](#)
 - GetPVLoggerRegistry, [628](#)
 - log_level_type, [627](#)
 - PVLoggerRegistry, [627](#)
 - SetNodeLogLevelExplicit, [628](#)
- PVLOGMSG_ALERT
 - pvlogger.h, [856](#)
- PVLOGMSG_C_ALERT
 - pvlogger_c.h, [860](#)
- PVLOGMSG_C_CRIT
 - pvlogger_c.h, [860](#)
- PVLOGMSG_C_EMERG
 - pvlogger_c.h, [860](#)
- PVLOGMSG_C_ERR
 - pvlogger_c.h, [860](#)
- PVLOGMSG_C_INFO
 - pvlogger_c.h, [860](#)

PVLOGMSG_C_INST_HLDBG
 pvlogger.c.h, 860
 PVLOGMSG_C_INST_LLDBG
 pvlogger.c.h, 860
 PVLOGMSG_C_INST_MLDBG
 pvlogger.c.h, 860
 PVLOGMSG_C_INST_PROF
 pvlogger.c.h, 860
 PVLOGMSG_C_INST_REL
 pvlogger.c.h, 860
 PVLOGMSG_C_NOTICE
 pvlogger.c.h, 860
 PVLOGMSG_C_STACK_DEBUG
 pvlogger.c.h, 860
 PVLOGMSG_C_STACK_TRACE
 pvlogger.c.h, 860
 PVLOGMSG_C_WARNING
 pvlogger.c.h, 860
 PVLOGMSG_CRIT
 pvlogger.h, 856
 PVLOGMSG_DEBUG
 pvlogger.h, 856
 PVLOGMSG_EMERG
 pvlogger.h, 856
 PVLOGMSG_ERR
 pvlogger.h, 856
 PVLOGMSG_FATAL_ERROR
 pvlogger.h, 856
 PVLOGMSG_INFO
 pvlogger.h, 857
 PVLOGMSG_INST_HLDBG
 pvlogger.h, 855
 PVLOGMSG_INST_LLDBG
 pvlogger.h, 855
 PVLOGMSG_INST_MLDBG
 pvlogger.h, 855
 PVLOGMSG_INST_PROF
 pvlogger.h, 856
 PVLOGMSG_INST_REL
 pvlogger.h, 856
 PVLOGMSG_NONFATAL_ERROR
 pvlogger.h, 857
 PVLOGMSG_NOTICE
 pvlogger.h, 857
 PVLOGMSG_STACK_TRACE
 pvlogger.h, 857
 PVLOGMSG_STATISTIC
 pvlogger.h, 857
 PVLOGMSG_VERBOSE
 pvlogger.h, 857
 PVLOGMSG_WARNING
 pvlogger.h, 857
 PVMEM_INST_LEVEL
 osclbase, 34
 osclconfig_memory.h, 828
 PVNETWORKADDRESS_LEN
 oscl_socket_types.h, 775
 PVOsclBase_Cleanup
 osclbase, 44
 PVOsclBase_Init
 osclbase, 44
 PVSCHEDNAMELEN
 osclproc, 104
 PVSchedulerStopper, 630
 OscExecSchedulerCommonBase, 397
 PVSchedulerStopper, 630
 PVSchedulerStopper
 ~PVSchedulerStopper, 630
 PVSchedulerStopper, 630
 PVSOCK_ERR_BAD_PARAM
 oscl_socket_imp_pv.h, 758
 PVSOCK_ERR_NOT_IMPLEMENTED
 oscl_socket_imp_pv.h, 758
 PVSOCK_ERR_NOT_SUPPORTED
 oscl_socket_imp_pv.h, 758
 PVSOCK_ERR_SERV_NOT_CONNECTED
 oscl_socket_imp_pv.h, 758
 PVSOCK_ERR_SOCK_NO_SERV
 oscl_socket_imp_pv.h, 758
 PVSOCK_ERR_SOCK_NOT_CONNECTED
 oscl_socket_imp_pv.h, 758
 PVSOCK_ERR_SOCK_NOT_OPEN
 oscl_socket_imp_pv.h, 758
 PVSockBufRecv, 631
 PVSockBufRecv, 631
 PVSockBufRecv
 iLen, 631
 iMaxLen, 631
 iPtr, 631
 PVSockBufRecv, 631
 PVSockBufSend, 632
 PVSockBufSend, 632
 PVSockBufSend
 iLen, 632
 iPtr, 632
 PVSockBufSend, 632
 PVThreadContext, 633
 OscExecSchedulerCommonBase, 397
 PVThreadContext, 633
 PVThreadContext
 ~PVThreadContext, 633
 EnterThreadContext, 633
 ExitThreadContext, 633
 Id, 633
 IsSameThreadContext, 633
 OscActiveObject, 634
 OscCoeActiveScheduler, 634
 OscCoeActiveSchedulerBase, 634

- OsciExecScheduler, [634](#)
- OsciExecSchedulerBase, [634](#)
- OsciExecSchedulerCommonBase, [634](#)
- OsciTimerObject, [634](#)
- PVActiveBase, [634](#)
- PVThreadContext, [633](#)
- ThreadHasScheduler, [634](#)

- QUE_ITER_BEGIN
 - osclproc, [104](#)
- QUE_ITER_END
 - osclproc, [104](#)

- Rand
 - OsciRand, [480](#)
- Read
 - Osci_File, [182](#)
 - OsciAsyncFile, [316](#)
 - OsciBinIStreamBigEndian, [325](#)
 - OsciFileCache, [400](#)
 - OsciNativeFile, [462](#)
- read
 - OSCL_String, [261](#)
 - OSCL_wString, [304](#)
- Read_uint16
 - OsciBinIStreamBigEndian, [325](#)
 - OsciBinIStreamLittleEndian, [328](#)
- Read_uint32
 - OsciBinIStreamBigEndian, [325](#)
 - OsciBinIStreamLittleEndian, [328](#)
- Read_uint8
 - OsciBinIStream, [322](#)
- ReadAsync
 - OsciNativeFile, [462](#)
- ReadAsyncCancel
 - OsciNativeFile, [462](#)
- rebalance
 - Osci_Rb_Tree_Base, [244](#)
- rebalance_for_erase
 - Osci_Rb_Tree_Base, [244](#)
- Recv
 - OsciRecvMethod, [489](#)
 - OsciRecvRequest, [490](#)
 - OsciSocketI, [537](#)
 - OsciSocketIBase, [542](#)
 - OsciTCPSocket, [569](#)
 - OsciTCPSocketI, [573](#)
- RecvFrom
 - OsciRecvFromMethod, [485](#)
 - OsciRecvFromRequest, [487](#)
 - OsciSocketI, [537](#)
 - OsciSocketIBase, [542](#)
 - OsciUDPSocket, [603](#)
 - OsciUDPSocketI, [607](#)
- RecvFromParam, [635](#)
 - RecvFromParam, [635](#)
- RecvFromParam
 - iAddr, [635](#)
 - iBufRecv, [635](#)
 - iFlags, [635](#)
 - iMultiMaxLen, [635](#)
 - iPacketLen, [635](#)
 - iPacketSource, [635](#)
 - RecvFromParam, [635](#)
- RecvFromRequest
 - OsciRecvFromMethod, [485](#)
- RecvFromSuccess
 - OsciSocketI, [537](#)
 - OsciSocketIBase, [542](#)
- RecvParam, [637](#)
 - RecvParam, [637](#)
- RecvParam
 - iBufRecv, [637](#)
 - iFlags, [637](#)
 - RecvParam, [637](#)
- RecvRequest
 - OsciRecvMethod, [489](#)
- RecvSuccess
 - OsciSocketI, [537](#)
 - OsciSocketIBase, [542](#)
- red
 - Osci_Rb_Tree_Node_Base, [252](#)
- RedBl
 - Osci_Rb_Tree_Node_Base, [252](#)
- refcount
 - CHeapRep, [129](#)
- reference
 - Osci_Map, [217](#)
 - Osci_Queue, [235](#)
 - Osci_Rb_Tree, [242](#)
 - Osci_Rb_Tree_Const_Iterator, [246](#)
 - Osci_Rb_Tree_Iterator, [249](#)
 - Osci_TagTree::const_iterator, [272](#)
 - Osci_TagTree::iterator, [275](#)
 - Osci_TAlloc, [280](#)
 - Osci_Vector, [284](#)
- Register
 - OsciComponentRegistry, [342](#)
 - OsciRegistryClient, [508](#)
 - OsciRegistryClientImpl, [511](#)
 - OsciRegistryServTlsImpl, [514](#)
- RegisterForCallback
 - OsciExecScheduler, [387](#)
 - OsciReadyQ, [484](#)
- registerInstance
 - OsciSingletonRegistry, [533](#)
 - OsciTLSRegistry, [595](#)
 - OsciTLSRegistryEx, [596](#)

- registerInstanceAndUnlock
 - OscSingletonRegistry, 533
- release
 - OscExclusiveArrayPtr, 380
 - OscExclusivePtr, 383
 - OscExclusivePtrA, 386
 - OSCLMemAutoPtr, 435
- RELOCK_MUTEX_ERROR
 - OscProcStatus, 475
- Remove
 - OscDoubleLink, 364
 - OscReadyQ, 484
 - OscSocketServRequestList, 559
 - OscTimerQ, 590
- remove
 - OscPriorityQueue, 471
 - OscPriorityQueueBase, 473
- remove_element
 - Osc_Linked_List, 208
 - Osc_Linked_List_Base, 213
 - Osc_MTLlinked_List, 226
- remove_ref
 - CHeapRep, 129
- removeALLAllocNodes
 - MM_Audit_Imp, 158
- removeAllocNode
 - MM_Audit_Imp, 158
- RemoveAppender
 - PVLogger, 620
- RemoveFixedCache
 - Osc_File, 182
- RemoveFromScheduler
 - OscActiveObject, 310
 - OscTimerObject, 587
 - PVActiveBase, 613
- RemoveRef
 - DNSRequestParam, 133
- removeRef
 - Osc_DefAllocWithRefCount, 173
 - OscMemPoolFixedChunkAllocator, 444
 - OscMemPoolResizableAllocator, 451
 - OscRefCount, 491
 - OscRefCountDA, 494
 - OscRefCountMTDA, 498
 - OscRefCountMTSA, 500
 - OscRefCountSA, 502
- Request
 - OscTimer, 582
- RequestCanceled
 - OscExecSchedulerCommonBase, 394
- RequestDone
 - OscDNSRequestAO, 362
 - OscSocketRequestAO, 551
- reserve
 - Osc_Queue_Base, 238
 - Osc_Vector_Base, 291
 - OscPriorityQueue, 471
- ReserveSpace
 - OscBinStream, 336
- Reset
 - OscDoubleListBase, 367
- reset
 - BufferState, 118
 - MM_FailInsertParam, 161
 - MM_Stats_t, 164
 - OscMemStatsNode, 457
- ResetLogPerf
 - OscExecSchedulerCommonBase, 394
- Resume
 - OscThread, 576
- ResumeScheduler
 - OscExecSchedulerCommonBase, 394
- retrieveParentTag
 - MM_Audit_Imp, 158
- retrieveParentTagLength
 - MM_Audit_Imp, 158
- RFC822ToPV8601
 - osclbase, 45
- Right
 - OscPtrC, 479
- right
 - Osc_Rb_Tree_Node_Base, 253
- rotate_left
 - Osc_Rb_Tree_Base, 244
- rotate_right
 - Osc_Rb_Tree_Base, 244
- Run
 - CallbackTimer, 123
 - OscDNSMethod, 357
 - OscDNSRequestAO, 362
 - OscSocketMethod, 545
 - OscSocketRequestAO, 551
 - PVActiveBase, 613
- RunError
 - OscActiveObject, 310
 - OscTimerObject, 587
 - PVActiveBase, 613
- RunIfNotReady
 - OscActiveObject, 311
 - OscTimerObject, 587
- RunSchedulerNonBlocking
 - OscExecScheduler, 387
- save_registry
 - TLSSStorageOps, 655
- second
 - Osc_Pair, 233
- SECONDS

- osclbase, [35](#)
- Seed
 - OscIRand, [480](#)
- Seek
 - OscI_File, [182](#)
 - OscIAsyncFile, [316](#)
 - OscIBinStream, [336](#)
 - OscIFileCache, [400](#)
 - OscINativeFile, [463](#)
- seek_type
 - OscI_File, [179](#)
- SEEKCUR
 - OscI_File, [179](#)
- SEEKEND
 - OscI_File, [179](#)
- seekFromCurrentPosition
 - OscIBinStream, [336](#)
- SEEKSET
 - OscI_File, [179](#)
- self
 - OscI_Map, [217](#)
 - OscI_Rb_Tree_Const_Iterator, [246](#)
 - OscI_Rb_Tree_Iterator, [249](#)
 - OscI_TagTree::const_iterator, [272](#)
 - OscI_TagTree::iterator, [275](#)
- SEM_NOT_SIGNALED_ERROR
 - OscIProcStatus, [475](#)
- Send
 - OscISendMethod, [522](#)
 - OscISendRequest, [523](#)
 - OscISocketI, [537](#)
 - OscISocketIBase, [542](#)
 - OscITCPsocket, [569](#)
 - OscITCPsocketI, [573](#)
- SendParam, [638](#)
 - SendParam, [638](#)
- SendParam
 - iBufSend, [638](#)
 - iFlags, [638](#)
 - iXferLen, [638](#)
 - SendParam, [638](#)
- SendRequest
 - OscISendMethod, [522](#)
- SendSuccess
 - OscISocketI, [537](#)
 - OscISocketIBase, [542](#)
- SendTo
 - OscISendToMethod, [524](#)
 - OscISendToRequest, [525](#)
 - OscISocketI, [537](#)
 - OscISocketIBase, [542](#)
 - OscIUDPSocket, [604](#)
 - OscIUDPSocketI, [607](#)
- SendToParam, [639](#)
 - SendToParam, [639](#)
- SendToParam
 - ~SendToParam, [639](#)
 - iAddr, [639](#)
 - iBufSend, [639](#)
 - iFlags, [639](#)
 - iXferLen, [639](#)
 - SendToParam, [639](#)
- SendToRequest
 - OscISendToMethod, [524](#)
- SendToSuccess
 - OscISocketI, [537](#)
 - OscISocketIBase, [542](#)
- Serv
 - OscIDNSRequestAO, [363](#)
- Set
 - OscIDoubleRunner, [368](#)
 - OscINameString, [460](#)
 - OscIPtr, [476](#)
 - OscIPtrC, [479](#)
- set
 - CHeapRep, [129](#)
 - CStackRep, [131](#)
 - OSCL_FastString, [176](#), [177](#)
 - OSCL_HeapStringA, [200](#), [201](#)
 - OSCL_wFastString, [294](#)
 - OSCL_wHeapStringA, [299](#)
 - OscIExclusiveArrayPtr, [380](#)
 - OscIExclusivePtr, [383](#)
 - OscIExclusivePtrA, [386](#)
 - OscISingleton, [531](#)
 - OscITLS, [591](#)
 - OscITLSEx, [593](#)
 - osclutil, [81–83](#)
- set_from_ntp_time
 - TimeValue, [653](#)
- set_from_system_time
 - NTPTIME, [168](#)
- set_int64
 - OscI_Int64_Utils, [203](#)
- set_len
 - OSCL_String, [261](#)
 - OSCL_wString, [305](#)
- set_length
 - OSCL_FastString, [177](#)
 - OSCL_wFastString, [294](#)
- set_next
 - OscI_Opaque_Type_Alloc_LL, [230](#)
- set_r
 - CHeapRep, [127](#)
- set_rep
 - CHeapRep, [129](#)
 - OSCL_String, [261](#), [262](#)
 - OSCL_wString, [305](#)

- set_to_current_time
 - NTPTIME, 168
 - TimeValue, 653
- set_to_zero
 - TimeValue, 654
- set_uint64
 - Osci_Int64_Utils, 203
- set_w
 - CFastRep, 127
- set_zulu
 - TimeValue, 654
- setAllocNodeFlag
 - MM_AllocBlockHdr, 147
- SetAsyncReadBufferSize
 - Osci_File, 182
- SetBusy
 - OsciActiveObject, 311
 - OsciTimerObject, 587
- SetCacheObserver
 - Osci_File, 183
- setChecksum
 - StrCSumPtrLen, 644
- SetExactFrequency
 - OsciTimer, 582
- SetFileHandle
 - Osci_File, 183
- SetFrequency
 - OsciTimer, 583
- SetInUse
 - OsciAsyncFileBuffer, 318
- SetLength
 - OsciPtr, 476
 - OsciPtrC, 479
- SetLoggingEnable
 - Osci_File, 183
- SetLogLevel
 - PVLogger, 620
- SetLogLevelAndPropagate
 - PVLogger, 621
- setMaxSzForNewMemPoolBuffer
 - OsciMemPoolResizableAllocator, 451
- SetMulticastTTL
 - OsciUDPSocket, 604
 - OsciUDPSocketI, 607
- SetNativeAccessMode
 - Osci_File, 183
- SetNativeBufferSize
 - Osci_File, 184
- SetNodeLogLevelExplicit
 - PVLoggerRegistry, 628
- SetObserver
 - OsciTimer, 583
- SetOffset
 - OsciAsyncFileBuffer, 318
- OsciDoubleListBase, 367
- SetOptionToReuseAddress
 - OsciIPSocketI, 417
 - OsciTCPSocket, 569
 - OsciUDPSocket, 604
- SetParent
 - PVLogger, 621
- SetPosition
 - OsciFileCacheBuffer, 402
- SetPrecedence
 - OsciSocketTOS, 563
- SetPriority
 - OsciSocketTOS, 563
 - OsciThread, 576
- setPtrLen
 - StrCSumPtrLen, 644
 - StrPtrLen, 647
 - WStrPtrLen, 658
- SetPVCacheSize
 - Osci_File, 184
- SetRecvBufferSize
 - OsciIPSocketI, 417
 - OsciSocketI, 537
 - OsciUDPSocket, 605
- setrep_to_char
 - OSCL_String, 262
- setrep_to_wide_char
 - OSCL_wString, 305
- SetScheduler
 - OsciExecSchedulerCommonBase, 394
- SetSize
 - Osci_File, 184
 - OsciNativeFile, 463
- SetSockOpt
 - OsciSocketI, 538
- SetStatus
 - OsciActiveObject, 311
 - OsciTimerObject, 587
- SetSummaryStatsLoggingEnable
 - Osci_File, 184
- SetTimestamp
 - MediaData, 143
- SetToHead
 - OsciDoubleRunner, 368
- SetTOS
 - OsciIPSocketI, 417
 - OsciTCPSocket, 570
 - OsciUDPSocket, 605
- SetToTail
 - OsciDoubleRunner, 368
- setWithoutOwnership
 - OSCLMemAutoPtr, 435
- ShowStats
 - OsciExecSchedulerCommonBase, 394

- ShowSummaryStats
 - OscExecSchedulerCommonBase, [394](#)
- Shutdown
 - OscShutdownMethod, [529](#)
 - OscShutdownRequest, [530](#)
 - OscSocketI, [538](#)
 - OscSocketIBase, [543](#)
 - OscTCPSocket, [570](#)
 - OscTCPSocketI, [573](#)
- ShutdownParam, [640](#)
 - ShutdownParam, [640](#)
- ShutdownParam
 - iHow, [640](#)
 - ShutdownParam, [640](#)
- ShutdownRequest
 - OscShutdownMethod, [529](#)
- Signal
 - OscSemaphore, [521](#)
- Size
 - Osc_File, [184](#)
 - OscAsyncFile, [316](#)
 - OscNativeFile, [463](#)
- size
 - CFastRep, [127](#)
 - CHeapRep, [129](#)
 - CStackRep, [131](#)
 - MM_AllocBlockHdr, [147](#)
 - MM_AllocInfo, [149](#)
 - MM_AllocQueryInfo, [151](#)
 - Osc_Map, [220](#)
 - Osc_Queue_Base, [238](#)
 - Osc_Rb_Tree, [242](#)
 - Osc_TagTree, [270](#)
 - Osc_Vector_Base, [291](#)
 - OscPriorityQueue, [471](#)
 - StrPtrLen, [647](#)
 - WStrPtrLen, [658](#)
- size_type
 - Osc_Map, [217](#)
 - Osc_Queue, [235](#)
 - Osc_Rb_Tree, [242](#)
 - Osc_Tag_Base, [266](#)
 - Osc_TagTree, [268](#)
 - Osc_TAlloc, [280](#)
- sizeof_T
 - Osc_Linked_List_Base, [214](#)
 - Osc_Queue_Base, [239](#)
 - Osc_Vector_Base, [291](#)
- skip_to_line_term
 - osclutil, [83](#)
- skip_to_whitespace
 - osclutil, [83](#)
- skip_whitespace
 - osclutil, [83](#)
- skip_whitespace_and_line_term
 - osclutil, [83](#)
- SLEEP_ONE_SEC
 - osclconfig_util.h, [848](#)
- SleepMillisec
 - OscThread, [576](#)
- Socket
 - OscSocketI, [538](#)
- SocketI
 - OscSocketRequestAO, [551](#)
- SocketObserver
 - OscSocketRequestAO, [551](#)
- SocketRequestParam, [641](#)
 - SocketRequestParam, [642](#)
- SocketRequestParam
 - iFxn, [642](#)
 - SocketRequestParam, [642](#)
- SocketServ
 - OscIPSocketI, [417](#)
- sort_children
 - Osc_TagTree::Node, [278](#)
- specialFragBuffer
 - OscBinStream, [337](#)
- Start
 - OscFileStats, [409](#)
- Start_on_creation
 - oscl_thread.h, [789](#)
- StartAsyncRead
 - OscAsyncFileBuffer, [318](#)
- StartCancel
 - OscSocketServRequestList, [559](#)
- StartMethod
 - OscDNSMethod, [357](#)
 - OscSocketMethod, [546](#)
- StartNativeScheduler
 - OscExecSchedulerCommonBase, [394](#)
- StartScheduler
 - OscExecSchedulerCommonBase, [394](#)
- State
 - OscSocketServIBase, [558](#)
- state
 - OscBinStream, [337](#)
- state_t
 - OscBinStream, [335](#)
- StaticJump
 - OscJump, [419](#)
- stats_overhead
 - MM_AuditOverheadStats, [160](#)
- Status
 - OscActiveObject, [311](#)
 - OscTimerObject, [588](#)
- status_t
 - BufFragStatusClass, [122](#)
- StatusRef

- OscActiveObject, [311](#)
- OscTimerObject, [588](#)
- StopScheduler
 - OscExecSchedulerCommonBase, [394](#)
- Str
 - OscNameString, [460](#)
- StrCSumPtrLen, [643](#)
 - osclutil, [68](#)
 - StrCSumPtrLen, [644](#)
- StrCSumPtrLen
 - checkSum, [644](#)
 - ChecksumType, [644](#)
 - getChecksum, [644](#)
 - isCIEquivalentTo, [644](#)
 - operator!=, [644](#)
 - operator=, [644](#)
 - operator==, [644](#)
 - setChecksum, [644](#)
 - setPtrLen, [644](#)
 - StrCSumPtrLen, [644](#)
- StrPtrLen, [646](#)
 - osclutil, [68](#)
 - StrPtrLen, [647](#)
- StrPtrLen
 - c_str, [647](#)
 - isCIEquivalentTo, [647](#)
 - isCIPrefixOf, [647](#)
 - isLetter, [647](#)
 - len, [647](#)
 - length, [647](#)
 - operator!=, [647](#)
 - operator=, [647](#)
 - operator==, [647](#)
 - ptr, [647](#)
 - setPtrLen, [647](#)
 - size, [647](#)
 - StrPtrLen, [647](#)
- Success
 - OscIDNSRequestAO, [363](#)
 - OscRecvFromRequest, [487](#)
 - OscRecvRequest, [490](#)
 - OscSendRequest, [523](#)
 - OscSendToRequest, [525](#)
 - OscSocketRequestAO, [551](#)
- SUCCESS_ERROR
 - OscProcStatus, [474](#)
- Suspend
 - OscThread, [577](#)
- Suspend_on_creation
 - oscl_thread.h, [789](#)
- SuspendScheduler
 - OscExecSchedulerCommonBase, [395](#)
- swap
 - Osc_Opaque_Type_Compare, [231](#)
- OscPriorityQueue, [471](#)
- SYSTEM_RESOURCES_UNAVAILABLE_ERROR
 - OscProcStatus, [475](#)
- tag
 - MM_AllocQueryInfo, [151](#)
 - MM_Stats_CB, [162](#)
 - Osc_Tag, [263](#)
 - Osc_TagTree::Node, [278](#)
 - OscMemStatsNode, [457](#)
- tag_ancestor
 - Osc_Tag_Base, [266](#)
- tag_base_type
 - Osc_Tag_Base, [266](#)
 - Osc_TagTree, [268](#)
- tag_base_unit
 - Osc_Tag_Base, [266](#)
- tag_cmp
 - Osc_Tag_Base, [266](#)
- tag_copy
 - Osc_Tag_Base, [266](#)
- tag_depth
 - Osc_Tag_Base, [266](#)
- tag_len
 - Osc_Tag_Base, [266](#)
- tag_type
 - Osc_TagTree, [268](#)
- tagAllocator
 - Osc_Tag, [263](#)
- TagTree_Allocator
 - osclmemory, [58](#)
- Tail
 - OscDoubleList, [365](#)
 - OscPriorityList, [468](#)
- tail
 - Osc_Linked_List_Base, [214](#)
- takeOwnership
 - OSCLMemAutoPtr, [436](#)
- TDNSRequestParamAllocator
 - oscl_dns_param.h, [674](#)
- Tell
 - Osc_File, [184](#)
 - OscAsyncFile, [316](#)
 - OscFileCache, [400](#)
 - OscNativeFile, [463](#)
- tellg
 - OscBinStream, [336](#)
- Terminate
 - OscThread, [577](#)
- the_list
 - Osc_MTLlinked_List, [226](#)
- THREAD_1_INACTIVE_ERROR
 - OscProcStatus, [474](#)

- THREAD_BLOCK_ERROR
 - OscIProcStatus, 475
- THREAD_NOT_OWN_MUTEX_ERROR
 - OscIProcStatus, 475
- ThreadHasScheduler
 - PVThreadContext, 634
- ThreadLogoff
 - OscIIPSocketI, 417
 - OscIReadyQ, 484
 - OscISocketI, 538
 - OscISocketMethod, 546
 - OscITCPSocket, 570
 - OscITCPSocketI, 573
 - OscIUDPSocket, 605
 - OscIUDPSocketI, 607
- ThreadLogon
 - OscIIPSocketI, 417
 - OscIReadyQ, 484
 - OscISocketI, 538
 - OscISocketMethod, 546
 - OscITCPSocket, 570
 - OscITCPSocketI, 573
 - OscIUDPSocket, 605
 - OscIUDPSocketI, 607
- ThreadPriorityAboveNormal
 - oscl_thread.h, 790
- ThreadPriorityBelowNormal
 - oscl_thread.h, 789
- ThreadPriorityHighest
 - oscl_thread.h, 790
- ThreadPriorityLow
 - oscl_thread.h, 789
- ThreadPriorityLowest
 - oscl_thread.h, 789
- ThreadPriorityNormal
 - oscl_thread.h, 789
- ThreadPriorityTimeCritical
 - oscl_thread.h, 790
- TickCount
 - OscITickCount, 579
- TickCountFrequency
 - OscITickCount, 579
- TickCountPeriod
 - OscITickCount, 579
- TicksToMsec
 - OscITickCount, 579
- TimeoutOccurred
 - OscITimerObserver, 589
- TimerBaseElapsed
 - CallbackTimerObserver, 125
 - OscITimer, 583
- TimerCallback
 - OscIReadyQ, 484
- timestamp
 - MediaData, 143
- TimeUnits
 - osclbase, 35
- TimeValue, 648
 - TimeValue, 650, 651
- TimeValue
 - get_ISO8601_str_time, 651
 - get_local_time, 651
 - get_pv8601_str_time, 651
 - get_rfc822_gmtime_str, 651
 - get_sec, 652
 - get_str_ctime, 652
 - get_timeval_ptr, 652
 - get_timevalue_in_usec, 652
 - get_usec, 652
 - is_zero, 653
 - is_zulu, 653
 - NTPTIME, 654
 - operator *=, 653
 - operator !=, 654
 - operator +=, 653
 - operator -=, 653
 - operator <, 654
 - operator <=, 654
 - operator =, 653
 - operator ==, 654
 - operator >, 654
 - operator >=, 654
 - set_from_ntp_time, 653
 - set_to_current_time, 653
 - set_to_zero, 654
 - set_zulu, 654
 - TimeValue, 650, 651
 - to_msec, 654
- TIpMReq
 - osclconfig_io.h, 822
- TLSStorageOps, 655
 - get_registry, 655
 - save_registry, 655
- to_msec
 - TimeValue, 654
- to_system_time
 - NTPTIME, 168
- TOO_MANY_FRAGS
 - BufFragStatusClass, 122
- TOO_MANY_THREADS_ERROR
 - OscIProcStatus, 474
- Top
 - OscIJump, 419
 - OscIReadyQ, 484
 - OscITimerQ, 590
- top
 - OscIPriorityQueue, 471

- TOSCL_StringOp
 - osclutil, 69
- TOSCL_wStringOp
 - osclutil, 69
- TOsclBasicLockObject
 - osclconfig_unix_android.h, 843
 - osclconfig_unix_common.h, 847
- TOsclConditionObject
 - osclconfig_proc_unix_android.h, 835
 - osclconfig_proc_unix_common.h, 837
- TOsclFileHandle
 - osclio, 96
- TOsclFileOffset
 - osclconfig_io.h, 822
- TOsclFileOffsetInt32
 - osclio, 96
- TOsclFileOp
 - osclio, 97
- TOsclHostent
 - osclconfig_io.h, 822
- TOsclMutexObject
 - osclconfig_proc_unix_android.h, 835
 - osclconfig_proc_unix_common.h, 837
- TOsclReady
 - osclproc, 105
- TOsclSemaphoreObject
 - osclconfig_proc_unix_android.h, 835
 - osclconfig_proc_unix_common.h, 837
- TOsclSockAddr
 - osclconfig_io.h, 822
- TOsclSockAddrLen
 - osclconfig_io.h, 822
- TOsclSocket
 - osclconfig_io.h, 822
- TOsclSocketServStatEvent
 - oscl_socket_stats.h, 771
- TOsclSocketStatEvent
 - oscl_socket_stats.h, 771
- TOsclThreadFuncArg
 - osclconfig_proc_unix_android.h, 835
 - osclconfig_proc_unix_common.h, 837
- TOsclThreadFuncPtr
 - oscl_thread.h, 789
- TOsclThreadFuncRet
 - osclconfig_proc_unix_android.h, 835
 - osclconfig_proc_unix_common.h, 837
- TOsclThreadId
 - osclconfig_proc_unix_android.h, 835
 - osclconfig_proc_unix_common.h, 837
- TOsclThreadObject
 - osclconfig_proc_unix_android.h, 835
 - osclconfig_proc_unix_common.h, 837
- TOsclTlsKey
 - osclbase, 35
- osclconfig_unix_android.h, 843
- osclconfig_unix_common.h, 847
- totalbytes
 - oscl_fsstat, 194
- totalNumAllocs
 - MM_Stats_t, 164
- totalNumBytes
 - MM_Stats_t, 164
- TOtherExecStats
 - OscExecSchedulerCommonBase, 392
- TPVDNSEvent
 - osclio, 98
- TPVDNSFxn
 - osclio, 98
- TPVServicePrecedence
 - OscSocketTOS, 562
- TPVServicePriority
 - OscSocketTOS, 562
- TPVSocketEvent
 - oscl_socket_types.h, 775
- TPVSocketFxn
 - oscl_socket_types.h, 776
- TPVSocketOptionLevel
 - oscl_socket_types.h, 776
- TPVSocketOptionName
 - oscl_socket_types.h, 776
- TPVSocketShutdown
 - oscl_socket_types.h, 776
- TPVThreadContext
 - osclproc, 105
- Trap
 - OscErrorTrapImp, 375
- TrapNoTls
 - OscErrorTrapImp, 375
- TReadyQueLink, 656
 - TReadyQueLink, 656
- TReadyQueLink
 - iAOPriority, 656
 - iIsIn, 656
 - iSeqNum, 656
 - iTimeQueuedTicks, 656
 - iTimeToRunTicks, 656
 - TReadyQueLink, 656
- trim
 - OscMemPoolResizableAllocator, 451
- TryLock
 - OscMutex, 459
- TryWait
 - OscSemaphore, 521
- TSocketServState
 - OscSocketServIBase, 557
- TSymbianAccessMode
 - Osc_File, 179

- uint
 - osclbase, [35](#)
- UINT64
 - osclconfig_unix_android.h, [843](#)
 - osclconfig_unix_common.h, [847](#)
- uint64
 - osclbase, [35](#)
- UINT64_HILO
 - osclconfig_unix_android.h, [843](#)
 - osclconfig_unix_common.h, [847](#)
- Unbind
 - OscISharedPtr, [528](#)
- UninstallScheduler
 - OscIExecSchedulerCommonBase, [395](#)
- unix_ntp_offset
 - osclbase, [45](#)
- Unlock
 - OscILockBase, [422](#)
 - OscIMutex, [459](#)
 - OscINullOrLock, [466](#)
 - OscIThreadLock, [578](#)
- UnRegister
 - OscIRegistryClient, [509](#)
 - OscIRegistryClientImpl, [511](#)
 - OscIRegistryServTlsImpl, [514](#)
- Unregister
 - OscIComponentRegistry, [342](#)
- UnTrap
 - OscIErrorTrapImp, [375](#)
- update
 - MM_Stats_t, [164](#)
- UpdateData
 - OscIAsyncFileBuffer, [318](#)
- updateEnd
 - OscIFileCacheBuffer, [402](#)
- updateStart
 - OscIFileCacheBuffer, [402](#)
- updateStatsNode
 - MM_Audit_Imp, [158](#)
- updateStatsNodeInFailure
 - MM_Audit_Imp, [158](#)
- UpdateTimers
 - OscIExecSchedulerCommonBase, [395](#)
- UpdateTimersMsec
 - OscIExecSchedulerCommonBase, [395](#)
- upper_bound
 - OscI_Map, [220](#), [221](#)
 - OscI_Rb_Tree, [242](#)
- usableSize
 - OscIFileCacheBuffer, [402](#)
- USEC_PER_SEC
 - osclbase, [45](#)
- validate
 - MM_Audit_Imp, [158](#)
 - OscIPriorityQueue, [472](#)
- validate_all_heap
 - MM_Audit_Imp, [158](#)
- validateblock
 - OscIMemPoolResizableAllocator, [451](#)
- Value
 - OscIAOStatus, [313](#)
- value
 - OscI_Rb_Tree_Node, [251](#)
 - OscI_TagTree::Node, [278](#)
- value_comp
 - OscI_Map, [221](#)
- value_compare
 - OscI_Map::value_compare, [222](#)
- value_type
 - OscI_Map, [217](#)
 - OscI_Queue, [235](#)
 - OscI_Rb_Tree, [242](#)
 - OscI_Rb_Tree_Const_Iterator, [246](#)
 - OscI_Rb_Tree_Iterator, [249](#)
 - OscI_Rb_Tree_Node, [251](#)
 - OscI_TagTree, [268](#)
 - OscI_TAlloc, [280](#)
 - OscI_Vector, [284](#)
 - OscIPriorityQueue, [470](#)
- vec
 - OscIPriorityQueue, [472](#)
- Wait
 - OscISemaphore, [521](#)
- WAIT_ABANDONED_ERROR
 - OscIProcStatus, [475](#)
- WAIT_TIMEOUT_ERROR
 - OscIProcStatus, [475](#)
- WaitAndPopTop
 - OscIReadyQ, [484](#)
- WaitForReadyAO
 - OscIExecSchedulerCommonBase, [395](#)
- WaitForRequestComplete
 - OscIReadyQ, [484](#)
- WaitOnRequests
 - OscISocketServRequestList, [559](#)
- Wakeup
 - OscISocketServRequestList, [559](#)
- writable
 - CFastRep, [127](#)
- Write
 - OscI_File, [185](#)
 - OscIAsyncFile, [316](#)
 - OscIFileCache, [400](#)
 - OscINativeFile, [463](#)
- write
 - OSCL_String, [262](#)

OSCL_wString, [305](#)
OscBinOStream, [329](#)
WriteUnsignedLong
 OscBinOStreamBigEndian, [331](#)
 OscBinOStreamLittleEndian, [333](#)
WriteUnsignedShort
 OscBinOStreamBigEndian, [331](#)
 OscBinOStreamLittleEndian, [333](#)
WriteUpdatesToFile
 OscFileCacheBuffer, [402](#)
WStrPtrLen, [657](#)
 osclutil, [68](#)
 WStrPtrLen, [658](#)
WStrPtrLen
 c_str, [658](#)
 isCIEquivalentTo, [658](#)
 len, [658](#)
 length, [658](#)
 operator!=, [658](#)
 operator=, [658](#)
 operator==, [658](#)
 ptr, [658](#)
 setPtrLen, [658](#)
 size, [658](#)
 WStrPtrLen, [658](#)

xsubi
 MM_FailInsertParam, [161](#)

Zero
 OscIPtr, [476](#)
 OscIPtrC, [479](#)